

## SAS® Dating Tips - A Beginner's Guide to SAS Dates

Erik S. Larsen, Independent Consultant

### Abstract

Do you want to be the life of the party at SESUG 2003? Being proficient at using SAS dates and times may not make you instantly popular, but it will make your programs work better and allow you to use the power of SAS dates to manipulate data and make your reports and graphs look more professional. We will explore the basics of how SAS stores dates and times internally and how you can use them to calculate values in terms of days, weeks and years. Finally we will explore some date functions and formats to make your reports look so good you will never want to look at a Microsoft Excel® spreadsheet again.

First off, let's get down to basics. How are dates stored in SAS? Generally speaking, SAS has two types of variables it stores in its data sets: numeric and character. SAS dates, times and datetimes fall into the numeric category, and more specifically, integer numbers (except for fractions of seconds.) When SAS moved from Version 5 to 6, it was decided to store dates and times in SAS as the number of days and/or seconds from January 1, 1960.

A SAS date value of zero is equivalent to January 1, 1960. Negative numbers refer to dates prior to January 1, 1960 and positive numbers refer to dates after January 1, 1960. It is a useful to know this feature because it allows the user to manipulate dates in mathematical terms. The example below shows how you can calculate the number of days between treatments:

```
DATA DATES;
    PREV_DT = "14FEB2003"d;
    NEXT_DT = "28MAR2003"d;

    DAYS = NEXT_DT-PREV_DT + 1;
RUN;
```

You will notice the letter "d" after the quotes. This tells SAS that it is a SAS date value, and not just a character constant. This allows for the following:

```
DAYS = 15,792 - 15,750 + 1 = 43.
```

Without the "d", SAS would attempt to add two character constants to a numeric constant and in the SAS log, resulting in a message of "Invalid numeric data."

SAS also provides numerous (too numerous to mention in this paper!) ways to input and display SAS dates. The SAS language can read in just about any date format you can imagine such as DDMONYYYY in the previous example and other common types such as MM-DD-YYYY and YY-MM-DD. Once the date is in a SAS data set, you are able to manipulate it regardless of format. An example is below:

```
DATA DATES;
    INPUT  JOBDATE      MMDDYY8.
          BEGDATE      YYMMDD6.
          ENDDATE      DATE9.;
RUN;
```

All of the above variables, JOBDATE, BEGDATE and ENDDATE are SAS date values and can be manipulated in many ways. The only assumption is that the input file has dates in the input format shown in the program. The DATE9. format is the DDMONYYYY display type.

SAS dates also have several basic functions available to the programmer. These allow you to perform subsetting or recasting dates by years, months or even days. These examples are below:

```
MON = MONTH(JOBDATE);  
DAY = DAY(JOBDATE);  
YR = YEAR(JOBDATE);  
RE_DATE = MDY(MON, DAY, YR);
```

The value of MON will be an integer from 1 to 12 and DAY will have a value of 1 to 31. The value of YR is the four digit year integer value. These functions allow you to group, subset, and analyze data by these categories. The MDY function will create a SAS date from variables and constants representing the month, day and year respectively.

A SAS datetime value is similar to a date value in that it is a number of intervals from January 1, 1960. The difference is that it is the number of **seconds** from January 1, 1960. This allows the programmer to perform similar operations that are performed on SAS dates in finding the difference in datetime values and two extremely useful functions, DATEPART and TIMEPART. These functions will take a SAS datetime value and extract only the SAS date or time, respectively. An example is below:

```
DT = "13DEC1968:12:32:12"dt;  
DATE = DATEPART(DT);  
TIME = TIMEPART(DT);
```

Please note that the “**dt**” after the quotes tells SAS that the DT variable is a SAS datetime value. This is analogous to the “**d**” for SAS date constants and similarly, a “**t**” for a SAS time constant.

One of the classic mistakes that beginning programmers make with SAS datetime values is that they read a SAS datetime value out of a database such as Sybase or DB2 and then either manipulate or print out a datetime value with a date format. SAS will not tell you that you are making this mistake in the SAS log but you will notice that your date values and calculations will not look correct. Be careful!!!!

A SAS time value is similar to a SAS datetime except that it is the number of seconds from midnight shown in hours, minutes and seconds.

This small introduction to SAS date, time and datetime values just scratches the surface of what you can do with these types of variables. In the *SAS Language Guide, V8* there are numerous other formats, informats and functions to apply to these types of variables. One of my favorite “interview questions” is to ask an applicant how a SAS date is stored internally. With the above information, you now have more than enough ammunition to answer the question appropriately.

### Author Contact Information

Erik S. Larsen  
Independent Consultant  
14 Gadsden Street  
Charleston, South Carolina 29401

A SAS Alliance Affiliate Member  
A SAS Certified Professional V8

(703) 568-3083 (work)

[Larsen770@aol.com](mailto:Larsen770@aol.com)

SAS® is a registered trademark of SAS Institute, Inc., in the USA and other countries. ® indicates USA registration.