Chapter 18

Reading Date and Time Values

Overview

In this chapter, you learn more about a special category of SAS informats/formats related to date and time values. These enable you to read and display a variety of date and time expressions.

Topics:

- Review how SAS stores date and time values.
- Use SAS informats/formats to read/display date/time values or expressions.
- Handle 2-digit year values (a review).
- Use date/time values in calculation.

How SAS Stores and Convert Date Values

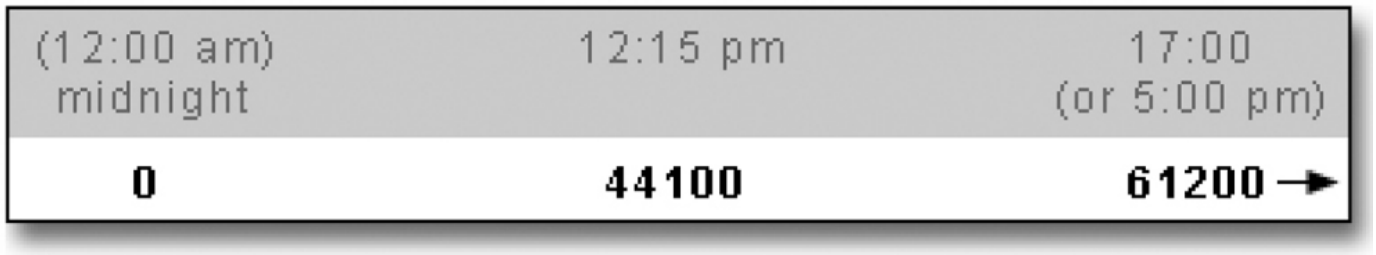
A SAS date value is the number of days from January 1, 1960, to the given date.

Some examples of how an appropriate SAS informat can convert different expressions of a date (January 2, 2000) to a single SAS date value.

Date Expression	SAS Date Informat	SAS Date Value
02Jan00	**DATEw.**	14611
01-02-2000	**MMDDYYw.**	14611
02/01/00	**DDMMYYw.**	14611
2000/01/02	**YYMMDDw.**	14611

How SAS Stores Time Values (a Review)

SAS **time** is stored as the number of **seconds** since the midnight of the day.



SAS **datetime** combines date and time in one value.
Datetime value is the number of **seconds** since the midnight of **January 1, 1960.**

More about SAS Date and Time Values

- SAS date values are based on the Gregorian calendar (365.2425 days). Julian calendar=365.25 days.
- Leap years are adjusted for, but leap seconds are not.
- No adjustments for daylight savings time are made in SAS.

Date and Time Informats

To read date and time values into a variable, SAS date and time informats are used in the INPUT statements. The general form is:

INPUT *<pointer-control> variable* ***informat.*** **;**

where

- *pointer-control* specifies the absolute or relative position to move the pointer.
- *variable* is the name of the variable being read.
- *informat.* is any valid SAS informat. Note that the informat includes a final period.

Some examples of common SAS data and time informats:

DATEw.
DATETIMEw.
MMDDYYw.
TIMEw.

The MMDDYYw. Informat

General form, values read with MMDDYYw. informat:

mmddyy or *mmddyyyy*

Where:

- *mm* is an integer between 01 and 12, representing the month.
- *dd* is an integer between 01 and 31, representing the day.
- *yy* or *yyyy* is an integer that represents the year.

In the MMDDYY*w*. informat, the month, day, and year fields can be separated by blanks or other delimiters (such as **-** or /) or nothing. If delimiters are present, they must occur between all fields in the values and must be the same delimiter. You specify a field width that includes not only the month, day, and year values, but also any delimiters as well. Here are some date expressions that you can read using the MMDDYY*w*. informat:

Date Expression	SAS Date Informat
101599	MMDDYY6.
10/15/99	MMDDYY8.
10 15 99	MMDDYY8.
10-15-1999	MMDDYY10.

The DATEw. Informat

General form, values read with DATEw. informat:

ddmmmyy or *ddmmmyyyy*

Where:

- *dd* is an integer between 01 and 31, representing the day
- *mmm* is the three-letter month's name
- *yy* or *yyyy* is an integer that represents the year.

Blanks or (other special characters, e.g., *) can appear between the day, month, and year, as long as you increase the width to include these delimiters. Here are some date expressions that you can read using the DATEw. informat:

Date Expression	SAS Date Informat
30May00	DATE7.
30May2000	DATE9.
30-May-2000	DATE11.

The TIMEw. Informat

General form, values read with TIMEw. informat:

hh:mm:ss.ss

Where:

- *hh* is an integer from 00 to 23, representing the hour.
- *mm* is an integer from 00 to 59, representing the minute.
- *ss.ss* is an optional field that represents seconds and hundredths of seconds. If you do not enter a value for ss.ss, a value of zero is assumed.

Some examples:

Time Expression	SAS Time Informat
17:00:01.34	TIME11.
17:00	TIME5.
2:34	TIME5.

Note: In the last example, the field is only 4 columns wide, but a *w* value of **5** is specified. **Five** is the minimum acceptable field width for the TIME*w*. informat. If you specify a *w* value less than 5, you get an error.

The DATETIMEw. Informat

General form, values read with DATETIMEw. informat:

ddmmmyy hh:mm:ss.ss

Where:

- *ddmmmyy* is the date value, the same form as the DATEw. Informat.
- *hh:mm:ss.ss* is the time value, the same form as the TIMEw. Informat.

Some examples:

Date and Time Expression	SAS Datetime Informat
30May2000:10:03:17.2	DATETIME20.
30May00 10:03:17.2	DATETIME18.
30May2000/10:03	DATETIME15.

(You must use delimiters to separate the values between year, hour, minutes, and seconds, but the delimiters may be different.)

Something Related to YEARCUTOFF= SAS System Option

- The value of the YEARCUTOFF= system option affects only two-digit year values. The default YEARCUTOFF= value is **1920** or **1926.**

Date Expression	SAS Date Informat	Interpreted As
06Oct59	date7.	06Oct1959
17Mar1783	date9.	17Mar1783

- However, if you specify an inappropriate field width, you will receive incorrect results.

Date Expression	SAS Date Informat	Interpreted As
17Mar1783	date7.	17Mar2017

(The informat specifies a *w* value that is too small to read the entire value, so the last two digits of the year are truncated.)

- Another problem arises if you use the wrong informat to read a date or time expression. The SAS log displays an invalid data message, and the variable's values are set to missing.

```
1---+----10
03/23/98
```

```
3  input birthday date8.;
4  run;
NOTE: Invalid data for BIRTHDAY in line 3 1-8.
RULE: ----+----1----+----3----+----4----+----5
3     03/23/98
BIRTHDAY=. _ERROR_=1 _N_=1
```

Read Date and Time Values

Example: data from a billing department of a hospital.
Variables: lastName, DateIn, DateOut, Roomrate, EquipCost.

Raw Data File Aprdata

```
1---+----10---+----20---+----30---+----40
Akron    04/05/99 04/09/99 175.00 298.45
Brown    04/12/99 05/01/99 125.00 326.78
Carnes   04/27/99 04/29/99 125.00 174.24
Denison  04/11/99 04/12/99 175.00  87.41
Fields   04/15/99 04/22/99 175.00 378.96
Jamison  04/16/99 04/23/99 125.00 346.28
```

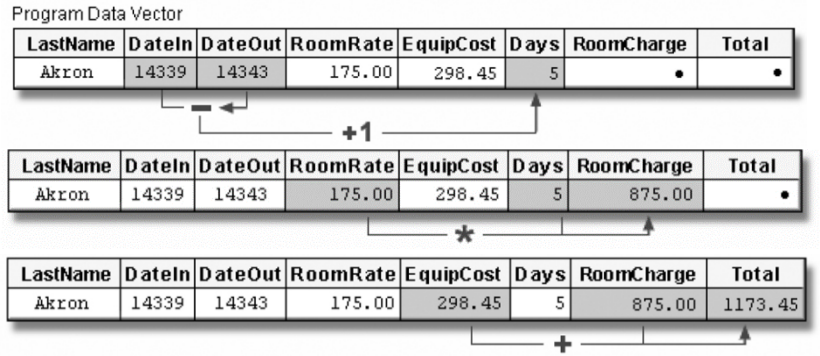
To read in this raw data, use the following input statement:

```
input LastName $8. @10 DateIn mmddyy8. +1 DateOut
mmddyy8. +1 RoomRate 6. @35 EquipCost 6.;
```

Use Date and Time Values in Calculations

Calculate: 1) duration of stay in the hospital
2) total room charge
3) total cost (room charge + equipment cost)

```
options yearcutoff=1920;
data perm.aprbills;
      infile aprdata;
      input LastName $8. @10 DateIn mmddyy8. +1 DateOut
            mmddyy8. +1 RoomRate 6. @35 EquipCost 6.;
      Days=DateOut-DateIn+1;          /* 1) */
      RoomCharge=days*roomrate;       /* 2) */
      Total=RoomCharge+EquipCost;     /* 3) */
run;
proc print data=perm.aprbills;
run;
```



Obs	LastName	DateIn	DateOut	RoomRate	EquipCost	Days	RoomCharge	Total
1	Akron	14339	14343	175	298.45	5	875	1173.45
2	Brown	14346	14365	125	326.78	20	2500	2826.78
3	Carnes	14361	14363	125	174.24	3	375	549.24
4	Denison	14345	14346	175	87.41	2	350	437.41
5	Fields	14349	14356	175	378.96	8	1400	1778.96
6	Jamison	14350	14357	125	346.28	8	1000	1346.28

Use Date and Time Formats

SAS provides many specialized date and time **formats** that enable you to specify how date and time values are displayed.

The WEEKDATEw. Format

The WEEKDATE*w*. format writes date values in the form *day-of-week, month-name dd, yy* (or *yyyy)*.

```
proc print data=perm.aprbills;
  format datein dateout weekdate17.;
run;
```

Obs	LastName	DateIn	DateOut	RoomRate	EquipCost	Days	RoomCharge	Total
1	Akron	Mon, Apr 5, 1999	Fri, Apr 9, 1999	175	298.45	5	875	1173.45
2	Brown	Mon, Apr 12, 1999	Sat, May 1, 1999	125	326.78	20	2500	2826.78
3	Carnes	Tue, Apr 27,1999	Thu, Apr 29, 1999	125	174.24	3	375	549.24
4	Denison	Sun, Apr 11, 1999	Mon, Apr 12, 1999	175	87.41	2	350	437.41
5	Fields	Thu, Apr 15, 1999	Thu, Apr 22, 1999	175	378.96	8	1400	1778.96
6	Jamison	Fri, Apr 16, 1999	Fri, Apr 23, 1999	125	346 28	8	1000	1346.28

Use Date and Time Formats

You can vary the results by changing the ***w*** value in the weekdatew. format.

FORMAT Statement	Result
format datein weekdate3.;	Mon
format datein weekdate9.;	Monday
format datein weekdate17.;	Mon, Apr 5, 1999
format datein weekdate**29**.;	Monday, April 5, 1999

Use Date and Time Formats

The WORDDATEw. Format

The WORDDATEw. format is similar to the WEEKDATEw. format, but it does not display the day of the week and the two-digit year values. It writes date values in the form of month-name dd, yyyy.

```
proc print data=perm.aprbills;
  format datein dateout worddate12.;
run;
```

Obs	LastName	DateIn	DateOut	RoomRate	EquipCost	Days	RoomCharge	Total
1	Akron	Apr 5, 1999	Apr 9, 1999	175	298.45	5	875	1173.45
2	Brown	Apr 12, 1999	May 1, 1999	125	326.78	20	2500	2826.78
3	Carnes	Apr 27,1999	Apr 29, 1999	125	174.24	3	375	549.24
4	Denison	Apr 11, 1999	Apr 12, 1999	175	87.41	2	350	437.41
5	Fields	Apr 15, 1999	Apr 22, 1999	175	378.96	8	1400	1778.96
6	Jamison	Apr 16, 1999	Apr 23, 1999	125	346 28	8	1000	1346.28

Use Date and Time Formats

You can also vary the results by changing the ***w*** value in the worddatew. format.

FORMAT Statement	Result
format datein worddate3.;	Apr
format datein worddate9.;	April
format datein worddate18.;	April 1, 1999

What are the results?

- format datein worddate5.;
- format datein worddate14.;

Questions: Use Date and Time Formats

What are the results?

- format datein worddate5.;
- format datein worddate14.;

- format datein worddate5.; → **Apr** (w: from 3 to 8 output the same value, Apr since it has not reached the max month length of September, which is 9 bytes).

- format datein worddate14.; → **Apr 1, 1999** (since it has not reached the max month length plus the rest of bytes: 9+1+2+1+1+4=18)