ABSTRACT
Databases and spreadsheets commonly store both the date and the time in a single field. Often, the time field is initialized to midnight. Here is a simple macro that can change a datetime to a date.

INTRODUCTION
This paper is meant as a brief introduction to the differences between a SAS® datetime and a SAS date. A beginning knowledge of SAS as well as a basic understanding of macro is assumed. Some of the SAS functions used are somewhat obscure, so it may help to have a manual (finite?) nearby.

THE PROBLEM
Dates stored on databases and even EXCEL spreadsheets are typically stored as datetime values. It is quite often the case that minutes and seconds are not relevant or useful as they are often stored as a default value of midnight. (Something to do with glass slippers and carriages, no doubt.) Like SAS date values, EXCEL date values are stored as numbers. However, SAS dates are the number of days elapsed since Jan 1, 1960, and Excel dates are the number of days starting at Jan 1, 1900. For example, Day 1 in SAS is Jan 2, 1960, Day 0 is Jan1, 1960. Day 1 in Excel is Jan 1, 1900.

THREE DIFFERENT ANIMALS

- A SAS date is the number of days since Jan 1, 1960. (Jan 1, 1960 is the official date SAS institute was incorporated).
- A SAS time is the number of seconds elapsed since midnight.
- A SAS datetime is the number of seconds elapsed since Jan 1, 1960.

There are separate classes of formats, and in some cases, functions, for each.

WHY NOT USE SAS FORMATS, INSTEAD?

For instance, you could change a SAS datetime 17OCT91:14:25:32 TO 17OCT91 by using DATETIME7. instead of DATETIME20.
But then, you
1) still need to keep this in mind when you do date arithmetic.
2) Assuming you haven’t permanently changed the format, you still have to remember to change all of your PRINT, FREQ AND REPORT procedures to reflect the correct format.

WHAT DOES THE MACRO DO?

1) Asks for three parameters: libname, memname and dtvar, all of them “keyword parameters” (can occur in any order, default to NULL).
2) Check for the name of the specified variable (returns a blank if variable doesn’t exist)
3) Ensures variable type is numeric.
4) Ensures that the value passed to the macro does not contain a missing value.
5) If it passes all of the tests above, then it executes the “datepart” function to return the date portion of the argument.
6) If the format being passed is DATETIME20, then change the format to DATE9, via PROC FORMAT called by a “CALL EXECUTE.”
WHY SO MUCH CODE?

The original version of the macro only required one parameter, the variable name, and no matter what, called the datepart function and applied the date format to the value. This was bad for several reasons. The user might have passed the wrong variable name, and would be left puzzling over the error messages that resulted. The format statement by itself is a compiler directive, so in the original version, the format would be immediately changed, whether it was appropriate for the variable given or not!

ROOM FOR IMPROVEMENT

1) Add a parameter to allow for multiple calls to PROC DATASETS
2) Default library to WORK
3) Allow width of datetime to be any user specified length.

CONCLUSION

This macro offers a relatively painless way to convert SAS datetimes into SAS dates.

REFERENCES


ACKNOWLEDGMENTS

Brendan Gilbane, (1930-2001) former dean of College of General Studies, Boston University, for allowing a "C plus" student into college.

CONTACT INFORMATION

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

Stanley Fogleman
Harvard Clinical Research Institute
930 Commonwealth Ave West
Boston MA 02215
Work Phone: (617) 632-1550
Email: vogelmann74@hotmail.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.

APPENDIX

MACRO DROPTIME

%macro droptime(libname=,memname=,dtvar=) / store des='changes datetime to date';
%* reformats a sas datetime variable to a sas date variable;
%* three required parameters:;
%* libname representing sas library to be changed;
%* memname representing sas dataset to be changed;
%* dtvar representing var to be changed;
%* must be run inside a data step;
if vname(&dtvar.) ne ''
and vtype(&dtvar.) eq 'N'
and not missing(&dtvar.)
then &dtvar. = datepart(&dtvar.);
if _n_ = 1 then
   if trim(left(vformat(&dtvar.))) eq 'DATETIME20.'
   then do;
      call execute("proc datasets nolist lib = &libname memtype=data;");
modify &memname;
   format &dtvar date9.;
   quit;
end;
else do;
   file log;
   put 'NOTE: No action taken by macro droptime';
   end;
%mend droptime;