Paper 4742-2020

Any Time Zone to Any Time Zone: A Macro to Convert Anything

Joe DeShon

ABSTRACT

This paper describes a flexible and fully configurable macro that converts a datetime stamp from any time zone to any other time zone. The macro can be configured to operate with any and all of the more than 24 time zones in the world.

INTRODUCTION

Our company is a large multi-national manufacturing business with operations in almost every time zone in the world. In addition, we often need to query databases that are maintained in Germany, England, Belgium, Spain, Argentina, Columbia, Brazil, Connecticut, Kansas City, Mexico, Philippines, Singapore, Korea, and China. Many of these databases write records with datetime stamps in their local time. SAS time zone conversion functions generally convert only local time zones. We needed an all-purpose macro capable of converting a datetime stamp from any time zone to any time zone with the need for the user to know what the conversion factors are.

This paper will demonstrate our solution to the problem: A simple, flexible, and customizable macro that will allow the conversion of a datetime from any time zone to any time zone.

EXAMPLE PROBLEMS

What is the time in Kansas City (Daylight Time) when the time is noon in Germany (Summer Time)?

What is the time in China when the time is 1:00 AM in Buenos Aires?

SAS TIME ZONE FUNCTIONS

Base SAS™ provides several functions to convert one time zones, but they are concerned with converting the local time to GMT or the other way around.

For example, the CURRENT_TIME_GMT function and the CURRENT_TIMESTAMP_GMT function return time in Greenwich Mean Time (GMT), but they only return the current time, taken from the system clock; they don't do any other conversions.

The several variations of the TZONE function converts a time to or from UTC time.

And the INTNX function (in several variations) can convert datetime values based on some offset, but it is up to the user to know what those offsets need to be for the required time zones.

MANUAL TIME ZONE CONVERSION IN BASE SAS

Since all datetime stamps in SAS are internally the number of seconds since January 1, 1960, it's relatively straightforward time simply add or subtract to convert from one time zone to another. Datetime is kept in seconds and most conversions need to be even

multiples of one hour. For clarity, I like to add or subtract the datetimes in the format of x*60*60, like this:

```
time_est = time_pst + (3*60*60).
```

In other words, Eastern Standard Time is Pacific Standard Time plus three hours (3*60*60 seconds).

Some time zones are half- or quarter hour differences, which can be represented by .25, .50, or .75 as the multiplier.

But to make this work, you would need to know the difference between every pair of time zones in the world, which would get rather tedious.

As previously noted, several variations of the INTNX functions can accomplish the same thing. But the strength of the INTNX function is that it can increment datetime values by any interval, even when such a conversion would be difficult by other means.

For example, the INTNX function can increment a datetime value of February 27 by seven days, even while accurately accounting for the number of days in February whether it involves a leap year or not.

In time zone conversion, it is necessary to increment only by a matter of hours and/or fractions thereof, which is pretty straightforward. In such cases, the power of the INTNX function, although available, is not needed.

For that reason, the paper's author's preference is to use simple arithmetic rather than the INTNX function while acknowledging that the solution could be done using INTNX as well.

UTC OFFSETS TO THE RESCUE

Fortunately, there is a (somewhat) standard method of identifying time zones with their conversion to UTC (Coordinated Universal Time -- the standard acronym is a compromise between the English acronym and the French acronym, TUC, or "Temps Universel Coordonné"). For most practical purposes, UTC is equivalent to the GMT time zone.

There are generally accepted three- or four-character representations for every time zone in the world, including their relative offset to UTC. Adding the offset to UTC converts from UTC to the time zone, and subtracting that offset converts the time in the other direction.

The abbreviations and their offsets are readily available on the Internet; a representative sample is printed below:

```
========================================
Abbr   Offset      Description
====   ======      ======================
IST     +5.5       India Standard
ART     -3.0       Argentina
HOA     -3.0       Hora Oficial Argentina
EET     +2.0       Eastern European
EEST    +3.0       Eastern European Summer
CET     +1.0       Central European
CEST    +2.0       Central European Summer
WET     +0.0       Central European
WEST    +1.0       Central European Summer
GMT     +0.0       Greenwich Mean
UTC     +0.0       Coordinated Universal
EST     -5.0       Eastern Standard
EDT     -4.0       Eastern Daylight
```

```
CST     -6.0       Central Standard
CDT     -5.0       Central Daylight
MST     -7.0       Mountain Standard
MDT     -6.0       Mountain Daylight
PST     -8.0       Pacific Standard
PDT     -7.0       Pacific Daylight
ZULU     0.0       Zulu (UTC)
========================================
```

Notice that the chart isn't perfect, sometimes there are multiple name and abbreviations for the same time zone. That can be dealt with in the configuration of the macro to suit the individual needs of your situation.

This macro splits the problem into two parts, first a conversion from the original time zone to UTC, then a conversion from UTC to the target time zone.

THE PROBLEM WITH DAYLIGHT SAVING TIME

The concept of Daylight Saving Time (known as "Summer Time" in many locations) is particularly complex. Not all locations observe DST. Locations are likely to convert from standard time and back to DST at different times (opposite in the southern hemisphere from the northern hemisphere). Some locations change their observance of DST from year to year. And some locations change times on something other than even hours.

This macro deals with DST by defining DST as separate time zones with separate UTC offsets. For example Eastern Standard Time is EST (-5), and Eastern Daylight Time is EDT (-4). It is the responsibility of the user to determine the appropriate time zone (standard or daylight) that is required for each particular application.

THE MACRO CODE

```
%macro convert_timezone(in_datetime,from_timezone,to_timezone);

%global readonly

  ist_offset              /** India Standard Time          **/
  art_offset              /** Argentina Time               **/  /** Argentina **/
  hoa_offset              /** Hora Oficial Argentina       **/  /** Argentina **/

  eet_offset              /** Eastern European Time        **/
  eest_offset             /** Eastern European Summer Time **/
  cet_offset              /** Central European Time        **/  /** Germany **/
  cest_offset             /** Central European Summer Time **/  /** Germany **/
  wet_offset              /** Western European Time        **/
  west_offset             /** Western European Summer Time **/

  gmt_offset              /** Greenwich Mean Time          **/
  utc_offset              /** Coordinated Universal Time   **/

  est_offset              /** Eastern Standard Time        **/
  edt_offset              /** Eastern Daylight Time        **/
  cst_offset              /** Central Standard Time        **/
  cdt_offset              /** Central Daylight Time        **/
  mst_offset              /** Mountain Standard Time       **/
```

```
  mdt_offset              /** Mountain Daylight Time       **/
  pst_offset              /** Pacific Standard Time        **/
  pdt_offset              /** Pacific Daylight Time        **/
  zulu_offset             /** Zulu Time (UTC)              **/

  ;

%let ist_offset  = +5.5;  /** India Standard Time          **/
%let art_offset  = -3.0;  /** Argentina Time               **/  /** Argentina **/
%let hoa_offset  = -3.0;  /** Hora Oficial Argentina       **/  /** Argentina **/

%let eet_offset  = +2.0;  /** Eastern European Time        **/
%let eest_offset = +3.0;  /** Eastern European Summer Time **/
%let cet_offset  = +1.0;  /** Central European Time        **/  /** Germany **/
%let cest_offset = +2.0;  /** Central European Summer Time **/  /** Germany **/
%let wet_offset  = +0.0;  /** Central European Time        **/
%let west_offset = +1.0;  /** Central European Summer Time **/

%let gmt_offset  = +0.0;  /** Greenwich Mean Time          **/
%let utc_offset  = +0.0;  /** Coordinated Universal Time   **/

%let est_offset  = -5.0;  /** Eastern Standard Time        **/
%let edt_offset  = -4.0;  /** Eastern Daylight Time        **/
%let cst_offset  = -6.0;  /** Central Standard Time        **/
%let cdt_offset  = -5.0;  /** Central Daylight Time        **/
%let mst_offset  = -7.0;  /** Mountain Standard Time       **/
%let mdt_offset  = -6.0;  /** Mountain Daylight Time       **/
%let pst_offset  = -8.0;  /** Pacific Standard Time        **/
%let pdt_offset  = -7.0;  /** Pacific Daylight Time        **/
%let zulu_offset =  0.0;  /** Zulu Time (UTC)              **/

%if
  %symexist(&&from_timezone._offset)            AND
  %symexist(&&to_timezone._offset)              AND
  %sysfunc(nmiss(&&&from_timezone._offset)) eq 0 AND
  %sysfunc(nmiss(&&&to_timezone._offset)) eq 0
    %then %do;

  (ifn(missing(&in_datetime),.,             /** If input time is null,     **/
                                            /** return a null. Otherwise,  **/
    (&in_datetime                           /** Input datetime             **/
           -                                /** MINUS offset               **/
     ((&&&from_timezone._offset) * 60 * 60) /** Convert to UTC             **/
           +                                /** PLUS offset                **/
     ((&&&to_timezone._offset) * 60 * 60))  /** Convert to output datetime **/
   ,.))                                     /** If result time is null,    **/
                                            /** return a null.             **/

%end;

%else %do;
  %put ERROR: Time zone abbreviation &from_timezone or &to_timezone not defined
in convert_timezone macro!;
%end;
```

```
%mend;
```

USING THE MACRO

The macro is a function-type macro with three parameters: The input datetime, the original time zone, and the desired time zone. The result of the function is the input datetime converted from the original time zone to the desired time zone.

Example:

```
data _null_;
  atlanta_datetime = datetime();
  germany_datetime = %convert_timezone(atlanta_datetime,est,eet);
  putlog atlanta_datetime= datetime20.;
  putlog  germany_datetime= datetime20.;
run;

atlanta_datetime=26AUG2019:15:53:17
germany_datetime=26AUG2019:22:53:17
```

The first parameter can be any SAS expression, variable, function, or constant. The second and third parameters must be unquoted constants, which will be appended within the macro to other values to form valid macro variables.

POSSIBLE IMPROVEMENT

In its current state, the macro is configured at compile time and it's not possible to change the time zones once that compilation has taken place. With a little bit of work with quoting and de-quoting, it would be possible to enhance this macro so that the second and third parameters could be sensitive to variable values, thus creating a more dynamic process.

CONCLUSION

This macro can relieve programmers of the tedious job of keeping track of multiple time zones throughout the world and can help standardize processing throughout the installation. It can be configured to suit the needs of individual circumstances.

REFERENCES

DATABASE OF TIME ZONES OFFSETS

https://www.iana.org/time-zones

LIST OF ABBREVIATIONS AND OFFSETS

https://en.wikipedia.org/wiki/List_of_time_zone_abbreviations

OTHER WIKIPEDIA ARTICLES

https://en.wikipedia.org/wiki/Category:Time_by_country
https://en.wikipedia.org/wiki/Lists_of_time_zones
https://en.wikipedia.org/wiki/List_of_time_zones_by_country
https://en.wikipedia.org/wiki/List_of_time_zones_by_UTC_offset
https://en.wikipedia.org/wiki/List_of_tz_database_time_zones
https://en.wikipedia.org/wiki/List_of_military_time_zones

SAS DOCUMENTATION FOR DATETIME FUNCTIONS

http://documentation.sas.com/?docsetId=allprodslang&docsetTarget=syntaxByCategory-function.htm&docsetVersion=9.4&locale=en#p1b7ohfmgjqjexn1tdm9w63qi31e

CONTACT INFORMATION

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

Joe DeShon
816-210-0950
joedeshon@yahoo.com