February, 1995

ADVISOR Answers

Q: Is there a simple, but accurate, formula to calculate ages from two date fields? I use INT(date1 - date2)/365.2425, but am unsatisfied with the results. For example, if a date of hire falls on the person's day of birth, the answer is not a whole number, but a fraction, such as 21.0017.

–Paul D'Ambrosio (via CompuServe)

A: This is yet another case where FoxPro's date functions do the job. We can break the dates into their component parts and use those parts to do the calculation. Rather than dealing specifically with age at hire, we'll view the problem more generally as finding the number of years between any two dates and write a function that does the trick.

There are three cases: when the month of the end date is after the month of the start date (so this year's anniversary has passed and the year should be counted); when the month of the end date is before the month of the start date (so this year's anniversary hasn't passed and the year should not be counted); and when the two dates are in the same month. When the months are the same, we have two cases, based on the day of the month - the day of the end date is the same or later than the day of the start date (so the anniversary has passed); the day of the end date is before the day of the start date (so the anniversary hasn't passed).

Here's a function that takes the two dates as parameters and returns the number of years between the two. If either parameter isn't a date or either one is the empty date, the function returns -1.

```
* CompYear.prg
* Compute years between any two events.
* Returns number of years.
* If input is invalid, returns -1

PARAMETERS dEnd,dStart

* Check for valid parameters
IF TYPE("dEnd")<>"D" OR TYPE("dStart")<>"D"
   RETURN -1
ENDIF

IF dEnd={} OR dStart={}
   RETURN -1
ENDIF

DO CASE
CASE MONTH(dEnd)>MONTH(dStart)
   * Subtract to get years
   RETURN YEAR(dEnd)-YEAR(dStart)
CASE MONTH(dEnd)<MONTH(dStart)
   * Subtract and then adjust
   RETURN YEAR(dEnd)-YEAR(dStart)-1
OTHERWISE
   * Same month, so check days
   IF DAY(dEnd)>=DAY(dStart)
```

```
      * End is same or later, just subtract
      RETURN YEAR(dEnd)-YEAR(dStart)
   ELSE
      * Subtract and adjust
      RETURN YEAR(dEnd)-YEAR(dStart)-1
   ENDIF
ENDCASE

RETURN
```

To get more precise ages requires more work, but it's similar. In some situations, you need to know the years and months. The function below returns a string formatted like "nnn Years nn Months". It counts only complete months based on calendar date, so if the start date is {1/7/94} and the end date is {10/6/94}, the function returns "0 Years 8 Months".

We take a slightly different approach this time. (This approach could have been used in the previous function, as well.) First, we compute a guess at the number of years by subtracting. Then, as above, we can divide the calculation into several cases. If the month of the end date is the same as or later than the month of the start date, the number of years is correct and we compute a guess at the number of months by subtracting. If we haven't reached the anniversary month yet (the month of the start date), we subtract one from the year and compute the number of months correcting for the end of the year.

Finally, we need to correct based on the day of the month. If the day of the end date is the same or later than the day of the start date, we're done. If not, we adjust the months. If there were no months (months were computed as 0), we have to adjust both the years and the months.

The last step is to combine the months and years into a neatly formatted string.

```
* CompYrMh.prg
* Compute years and months between any two events.
* Returns string in "nnn Years nn Months" format;
* If input is invalid, returns the empty string.

PARAMETERS dEnd,dStart

IF TYPE("dEnd")<>"D" OR TYPE("dStart")<>"D"
   RETURN ""
ENDIF

IF dEnd={} OR dStart={}
   RETURN ""
ENDIF

PRIVATE nYears,nMonths

* rough guess at years
nYears=YEAR(dEnd)-YEAR(dStart)

* get months and adjust years
DO CASE
CASE MONTH(dEnd)>=MONTH(dStart)
   nMonths=MONTH(dEnd)-MONTH(dStart)
```

```
CASE MONTH(dEnd)<MONTH(dStart)
   nYears=nYears-1
   nMonths=12-(MONTH(dStart)-MONTH(dEnd))
ENDCASE

* now correct for days
IF DAY(dEnd)<DAY(dStart)
       IF nMonths=0
         nYears=nYears-1
         nMonths=11
       ELSE
         nMonths=nMonths-1
       ENDIF
ENDIF

RETURN LTRIM(STR(nYears,3))+" Years "+ ;
       LTRIM(STR(nMonths,2))+" Months"
```

You can take this farther if you want. For example, you may to decide to count a month if you've had at least 15 days, so the example above (start date = {1/7/94}, end date={10/6/94}) would result in "0 years 9 months". You might want the output in a different format or to use "Year" instead of "Years" when nYears is 1. You might want to return years, months and days. All of this is possible with the date functions. As some of my college textbooks used to say, I leave these as exercises for the reader.

–Tamar