March, 1994

## Advisor Answers

Q: How can I determine the starting date of a week? In a report, I want to have a line in the title bar say:

```
      "Report for Week of" + [ ] + start
```

where start is a variable date that I want to be Monday, regardless of the day that I print the report.

–Wayne A. Smythe (via CompuServe)

A: FoxPro has terrific abilities for manipulating dates. We can add or subtract a number from a date to get the date that number of days after or before the original date. To determine the starting date of a week, we can write a function which uses FoxPro's built-in date handling functions.

The key here is the DOW() function, which returns a number indicating the day of the week for the date passed to it. Sunday is day 1; Saturday is day 7. So, for example:

```
?DOW({03/10/94})
```

returns 5, because March 10, 1994 is a Thursday.

The trick to getting the date of the first day of the week is to subtract the day of the week from the date we're interested in. This will always give us the date of the preceding Saturday. That is,

```
a_date-DOW(a_date)
```

returns the date of the Saturday preceding a_date.

If we're interested in the week beginning Sunday, it's easy at this point. Just add 1 to the result.

```
a_date-DOW(a_date)+1
```

gives the date of the Sunday at the beginning of the week containing a_date. That is, if a_date is a Sunday, it returns a_date; otherwise, it returns the date of the preceding Sunday.

For example,

```
a_date={03/10/94}
```

```
?a_date-DOW(a_date)
03/05/94
?a_date-DOW(a_date)+1
03/06/94
```

When we're interested in weeks starting with a day other than Sunday, it gets a little more complicated. We have to distinguish between those days which fall before and those which fall after the "week starts" day. That is, if we're interested in weeks starting on, say, Wednesday, we can't just take

```
a_date-DOW(a_date)+4
```

because that would put Sunday, Monday and Tuesday into the wrong week. Let's try it using the example above.

```
?a_date-DOW(a_date)+4
03/09/94
```

So far, so good. But what if we start with a Monday?

```
a_date={03/07/94}
?a_date-DOW(a_date)+4
03/09/94
```

This isn't right. If we're interested in weeks starting on Wednesday, March 7, 1994 falls into the week of March 2, not the week of March 9.

So, we need to add or subtract a number based on what week starting day we want and the day of the week of the date we're working with. If the date we're working with falls earlier in the week than the week starting day, we need to subtract the difference; if it falls later, we need to add the week starting day. We can express this as follows, assuming week_start is a variable containing the day number of the week starting day:

```
IF DOW(a_date)>=week_start
  ?a_date-DOW(a_date)+week_start
ELSE
  ?a_date-DOW(a_date)-7+week_start
ENDIF
```

Now lets put all this together to make a function which we can call from reports and screens.

```
FUNCTION startday
* Return the starting date of the week containing the specified
* date. The day of the week on which the week starts may be
* optionally specified. If not, Sunday is assumed.
PARAMETERS the_date,start_day
```

```
* the_date = the date for which the starting date is needed
* start_day = the day of the week on which the week is presumed
*                    to start.
*                    1=Sunday, 7=Saturday.
*                    If omitted, Sunday is presumed.
PRIVATE dow_date
* dow_date = the day of the week of the_date
* check that incoming date is, in fact, a date
IF TYPE("the_date")<>"D"
  * if there's a problem, return the empty date.
  * this could be changed to provide a message.
  RETURN {  /  /  }
ENDIF
* check whether a starting day was specified
IF TYPE("start_day")<>"N" OR .NOT. BETWEEN(start_day,1,7)
  start_day=1
ENDIF
* get the day of the week for the specified date
dow_date=DOW(the_date)
IF dow_date>=start_day
  RETURN the_date-dow_date+start_day
ELSE
  RETURN the_date-dow_date-7+start_day
ENDIF
RETURN
```

To call this function, pass it one or two parameters. The first parameter is the date for which the week start date is needed. The second, optional, parameter is the day of the week on which the week starts.

For example:

```
?startday({03/10/94})
03/06/94
?startday({03/10/94},2)
03/07/94
?startday({03/10/94},4)
03/09/94
?startday({03/10/94},6)
03/04/94
```

If speed is an issue (for example, many starting dates need to be calculated), it may be useful to use IIF() to write an expression instead of calling a function. Tested in FoxPro for Windows, an inline version using IIF() was almost 70 times faster than the function version. The function itself took only about .005 of a second (on a 486/66DX2 with 8MB), so for individual uses, STARTDAY() is quite fast. In tight loops though, the IIF() version will speed things up considerably.

STARTDAY() can be replaced with the following code:

```
IIF(DOW(the_date)>=start_day, ;
    the_date-DOW(the_date)+start_day, ;
    the_date-DOW(the_date)+start_day-7)
```

–Tamar