Does Anybody Know What Day It Is?
Working With Date Intervals, Multipliers, and Shift Index
By: Teresa Schudrowitz

Abstract
Do SAS date calculations often leave you confused? Do you have to perform date calculations where you need to use a multiple of a date interval? Do you need to advance dates in multiples of 2 months or 4 years? Do you have to return a date value using a non-default starting point, such as years beginning on March 1st rather than January 1st?

If you have ever asked any of these questions than this paper is for you! Throughout this paper we will look at the INTNX interval function to explore the following:

- Single unit date intervals
- Using the multiplier to create a multiunit date intervals
- Using the shift-index to move an interval start date

In addition, we will look at why we receive the results we do when we are using multiunit intervals such as DAY or WEEK. For instance, when going to the beginning of the DAY3 interval for the date Sunday December 31, 2006, why is the date Sunday December 31, 2006? Then, when the date is changed to Sunday April 1, 2007, why is the beginning of the DAY3 interval Saturday March 31, 2007?

Introduction
When working with numeric data calculations are usually straight forward. If the value of x is 50 and you want to advance the value by 10 you merely need to state:

```
z=x+10;
```

This then results in 50 plus 10, providing the result of 60.

While it would be nice if working with date calculations was as simple as 2 + 2 =4, working with dates is not so straight forward. If you have start crate of August 28, 2006, what are you advancing the date by? Days? Weeks? Months? Years? The time interval must be considered.

Functions to work with Dates
SAS has various functions available to assist in date calculations. The two most commonly used are:

INTNX - advance the date a number of intervals
INTCK - determine the number of intervals between two dates

If we look at the INTNX function, this is our date addition. We have a starting date which we want to add x number of periods and receive a new date as a result. The syntax for the INTNX function is:

```
INTNX ('interval', start-from, increment <,'alignment'>)
```

We have a starting date of August 28, 2006. Now we need to determine the interval we want to advance the date by. The available intervals are:

DAY
WEEKDAY
WEEK
TENDAY
SEMIMONTH
MONTH
QTR
SEMIYEAR
YEAR

The DAY interval will advance the date by the specified number of days. For example, if 5 days are specified, then the resulting date will be September 2, 2006. The remaining intervals will advance the

date by the number of periods and, by default, set the date to the first day in the period. For instance, if we were to specify 2 months the resulting date will be October 1, 2006. The following program advances the start date by 5 for each interval.

```
Data _null_;
  start_dt = '28AUG2006'd;
  _5days =intnx('day', start_dt, 5);
  _5weekdays = intnx ('weekday', start_dt, 5);
  _5weeks = intnx('week', Start_dt, 5);
  _5tendays = intnx('tenday', start_dt, 5);
  _5semimonth = intnx('semimonth', start_dt, 5);
  _5months = intnx('month', start_dt, 5);
  _5qtrs = intnx('qtr', start_dt, 5);
  _5semiyears = intnx('semiyear', start_dt, 5);
  _5years = intnx ('year', start_dt, 5);
  format start_dt _5: date9.;
  put start_dt= _5days=
      _5weekdays=  _5weeks=
      _5tendays=   _5semimonth=
      _5months=    _5qtrs=
      _5semiyears= _5years=;
run;
```

When this program completes the following dates will be returned:

_5days=02SEP2006
_5weekdays=04SEP2006
_5weeks=01OCT2006
_5tendays=11OCT2006
_5semimonth=01NOV2006
_5months=01JAN2007
_5qtrs=01OCT2007
_5semiyears=01JAN2009
_5years=01JAN2011

Each of these dates went to the first day in the period. For instance, August 28th is a Monday. When the date is advanced 5 weeks, the resulting date of October 1st is a Sunday. Even though October 1st is a day shy of the 5 weeks, Sunday is the first day of the week interval. The beginning of each interval is at the following points on the calendar:

Interval	Begin Point
DAY	each day
WEEKDAY	each day
WEEK	each Sunday
TENDAY	1st, 11th, and 21st of the month
SEMIMONTH	1st and 16th of the month
MONTH	1st of the month
QTR	1st of January, April, July, and October
SEMIYEAR	1st of January and July
YEAR	1st of January

Single versus Multiunit Intervals

Each of the intervals previously reviewed are single unit intervals. When YEAR is the specified interval the date will advance one year at a time. By applying a multiplier to the interval, a single unit interval can be changed into a multiunit interval:

interval<multiplier>

In this manner 2 month intervals can be established by the following:

MONTH2

Now, rather than the 1st of every month as the interval begin point, the 1st of ever other month is the interval begin point.

The single unit intervals use a set calendar point to begin each interval. While this is also true for multiunit intervals, new questions arise with these types of intervals. In a single unit interval, both October 1st and November 1st are interval begin points. If though, the interval is the multiunit interval of MONTH2, what is the begin point of the 2 month span? Is it October 1st or November 1st?

In answering the begin point question, it is important to remember the basis of SAS dates. SAS dates are stored based on January 1, 1960. Therefore, most multiunit intervals are also based from January 1, 1960. Looking at our MONTH2 interval, the interval starting points are:

January 1, 1960
March 1, 1960
May 1, 1960
July 1, 1960
September 1, 1960
November l, 1960
January 1, 1961
etc.

Since the months in a year can be evenly divided by 2, we can easily determine the interval begin points as January, March, May, July, September, and November, regardless of the year. A similar method may be used whenever the multiplier is evenly divisible into a year. For example, if the interval is MONTH3 the start points would be January, April, July, and October.

The calculation of interval start points becomes less intuitive if the multiplier is not evenly divisible into a year. For example, the begin points for MONTH5 would be as follows:

January 1, 1960
June 1, 1960
November 1, 1960
April 1, 1961
September 1, 1961
February 1, 1962
etc.

Some of the common multiunit intervals which may be used are:

Interval	Begin Point
MONTH4	1st of January, May, and September
MONTH6	1st of January and July
QTR2	1st of January and July
YEAR2	1st of January in even years

Week as a multiunit Interval

When the WEEK interval is used as a multiunit interval it is an exception to the rule of using January 1, 1960 as the basis point. As a single unit interval the starting point for the WEEK interval is Sunday. January 1, 1960 was a Friday. Therefore, SAS uses the Sunday of the week containing January 1, 1960. As a result, the begin point for WEEK<multiplier> is calculated from December 27, 1959.

Shifted Intervals

We have seen how we can advance a date by a specified internal. By default, the date is set to the begin point of the interval. Therefore, if we advance our date by 1 year:

```
new_date = intnx('year', '28AUG2006'd, 1);
```

new_date becomes January 1, 2007. What though, if you require an alternative start point? Your company's fiscal year begins April 1st rather than January 1st. Therefore, when you advance the date one year to the start of the next fiscal year, you want the date to be April 1st not January 1st.

To shift the start point of an interval a shift index will be applied to the interval. The syntax is as follows:

interval<.start-point>

The start-point allows you to shift an interval by a sub period. In our example above the interval is YEAR and the sub period within years is month. To set the period starting point as April 1st we need to shift the begin point of year by four months. Therefore, the interval would be written as:

YEAR.4

Now when we use our new interval to calculate new_date:

```
new_date = intnx('year.4', '28AUG2006'd, 1);
```

new_date now results in April 1, 2007.

When a shift index is used the value of start-point must be equal to or less than the maximum number of sub periods in the interval. For instance, when the interval is year the sub periods are months. Within a year there are twelve months. Therefore, any start-point value between one and twelve is valid. An interval of YEAR.13 would be invalid since a single year does not contain thirteen months.

Summary

The SAS system’s use of a serial date technique, along with accompanying informats, formats, and functions, and a little bit of math, allow the user to solve virtually any date and time problem.