Chapter 3
Date Intervals, Formats, and Functions

Chapter Table of Contents

Chapter 3
Date Intervals, Formats, and Functions

This chapter summarizes the time intervals, date and datetime informats, date and datetime formats, and date and datetime functions available in the SAS system. The use of these features is explained in Chapter 2, “Working with Time Series Data.” The material in this chapter is also contained in the *SAS Language: Reference*. Because these features are useful for work with time series data, documentation of these features is consolidated and repeated here for easy reference.

Time Intervals

This section provides a reference for the different kinds of time intervals supported by the SAS System. How intervals are used is not discussed here; see Chapter 2, “Working with Time Series Data,” for an introduction to the use of time intervals.

Some interval names are for use with SAS date values while other interval names are for use with SAS datetime values. The interval names used with SAS date values are YEAR, SEMIYEAR, QTR, MONTH, SEMIMONTH, TENDAY, WEEK, WEEKDAY, and DAY. The interval names used with SAS datetime or time values are HOUR, MINUTE, and SECOND. Various abbreviations of these names are also allowed, as described in the section "Summary of Interval Types."

Interval names for use with SAS date values can be prefixed with ’DT’ to construct interval names for use with SAS datetime values. The interval names DTYEAR, DTSEMIYEAR, DTQTR, DTMONTH, DTSEMIMONTH, DTTENDAY, DTWEEK, DTWEEKDAY, and DTDAY are used with SAS datetime or time values.

Constructing Interval Names

Multipliers and shift indexes can be used with the basic interval names to construct more complex interval specifications. The general form of an interval name is as follows:

NAMEn.s

The three parts of the interval name are:

NAME	the name of the basic interval type. For example, YEAR specifies yearly intervals.
n	an optional multiplier which specifies that the interval is a multiple of the period of the basic interval type. For example, the interval YEAR2 consists of two-year, or biennial, periods.

s an optional starting subperiod index that specifies that the intervals are shifted to later starting points. For example, YEAR.3 specifies yearly periods shifted to start on the first of March of each calendar year and to end in February of the following year.

Both the multiplier *n* and the shift index *s* are optional and default to 1. For example, YEAR, YEAR1, YEAR.1, and YEAR1.1 are all equivalent ways of specifying ordinary calendar years.

Both the multiplier *n* and the shift index *s* are optional and default to 1. For example, YEAR, YEAR1, YEAR.1, and YEAR1.1 are all equivalent ways of specifying ordinary calendar years.

Shifted Intervals

Different kinds of intervals are shifted by different subperiods.

- YEAR, SEMIYEAR, QTR, and MONTH intervals are shifted by calendar months.
- WEEK, WEEKDAY, and DAY intervals are shifted by days.
- SEMIMONTH intervals are shifted by semi-monthly periods.
- TENDAY intervals are shifted by ten-day periods.
- HOUR intervals are shifted by hours.
- MINUTE intervals are shifted by minutes.
- SECOND intervals are shifted by seconds.

If a subperiod is specified, the shift index cannot be greater than the number of subperiods in the whole interval. For example, you could use YEAR2.24, but YEAR2.25 would be an error because there is no twenty-fifth month in a two-year interval. For interval types that shift by subperiods that are the same as the basic interval type, only multiperiod intervals can be shifted.

For example, MONTH type intervals shift by MONTH subintervals; thus, monthly intervals cannot be shifted since there is only one month in MONTH. However, bimonthly intervals can be shifted, since there are two MONTH intervals in each MONTH2 interval. The interval name MONTH2.2 specifies bimonthly periods starting on the first day of even-numbered months.

Alignment of Intervals

Intervals that represent divisions of a year are aligned with the start of the year (January). MONTH2 periods begin with odd-numbered months (January, March, May, and so on). Likewise, intervals that represent divisions of a day are aligned with the start of the day (midnight). Thus, HOUR8.7 intervals divide the day into the periods 06:00 to 14:00, 14:00 to 22:00, and 22:00 to 06:00.

Intervals that do not nest within years or days are aligned relative to the SAS date or datetime value 0. The arbitrary reference time of midnight on January 1, 1960, is used as the origin for nonshifted intervals, and shifted intervals are defined relative to that reference point. For example, MONTH13 defines the intervals January 1, 1960, February 1, 1961, March 1, 1962, and so forth, and the intervals December 1, 1959, November 1, 1958, and so on before the base date January 1, 1960.

Similarly, WEEK2 interval beginning days are aligned relative to the Sunday of the week of January 1, 1960. The interval specification WEEK6.13 defines six-week periods starting on second Fridays, and the convention of alignment relative to the period containing January 1, 1960 tells where to start counting to find out what dates correspond to the second Fridays of six-week intervals.

See the section “Alignment of SAS Dates” later in this chapter.

Summary of Interval Types

The interval types are summarized as follows.

YEAR
specifies yearly intervals. Abbreviations are YEAR, YEARS, YEARLY, YR, ANNUAL, ANNUALLY, ANNUALS. The starting subperiod *s* is in months.

SEMIYEAR
specifies semiannual intervals (every six months). Abbreviations are SEMIYEAR, SEMIYEARS, SEMIYEARLY, SEMIYR, SEMIANNUAL, SEMIANN.

The starting subperiod *s* is in months. For example, SEMIYEAR.3 intervals are March–August and September–February.

QTR
specifies quarterly intervals (every three months). Abbreviations are QTR, QUARTER, QUARTERS, QUARTERLY, QTRLY, QTRS. The starting subperiod *s* is in months.

MONTH
specifies monthly intervals. Abbreviations are MONTH, MONTHS, MONTHLY, MON.

The starting subperiod *s* is in months. For example, MONTH2.2 intervals are February–March, April–May, June–July, August–September, October–November, and December–January of the following year.

SEMIMONTH
specifies semimonthly intervals. SEMIMONTH breaks each month into two periods, starting on the first and sixteenth day. Abbreviations are SEMIMONTH, SEMIMONTHS, SEMIMONTHLY, SEMIMON.

The starting subperiod *s* is in SEMIMONTH periods. For example, SEMIMONTH2.2 specifies intervals from the sixteenth of one month through the fifteenth of the next month.

TENDAY

specifies ten-day intervals. TENDAY breaks the month into three periods, the first through the tenth day of the month, the eleventh through the twentieth day of the month, and the remainder of the month. (TENDAY is a special interval typically used for reporting automobile sales data.)

The starting subperiod *s* is in TENDAY periods. For example, TENDAY4.2 defines forty-day periods starting at the second TENDAY period.

WEEK

specifies weekly intervals of seven days. Abbreviations are WEEK, WEEKS, WEEKLY.

The starting subperiod *s* is in days, with the days of the week numbered as 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, and 7=Saturday. For example, WEEK.7 means weekly with Saturday as the first day of the week.

WEEKDAY
WEEKDAY17W

specifies daily intervals with weekend days included in the preceding week day. Abbreviations are WEEKDAY, WEEKDAYS.

The WEEKDAY interval is the same as DAY except that weekend days are absorbed into the preceding weekday. Thus there are five WEEKDAY intervals in a calendar week: Monday, Tuesday, Wednesday, Thursday, and the three-day period Friday-Saturday-Sunday.

The default weekend days are Saturday and Sunday, but any one to six weekend days can be listed after the WEEKDAY string and followed by a W. Weekend days are specified as '1' for Sunday, '2' for Monday, and so forth. For example, WEEKDAY67W specifies a Friday-Saturday weekend. WEEKDAY1W specifies a six-day work week with a Sunday weekend. WEEKDAY17W is the same as WEEKDAY.

The starting subperiod *s* is in days.

DAY

specifies daily intervals. Abbreviations are DAY, DAYS, DAILY. The starting subperiod *s* is in days.

HOUR

specifies hourly intervals. Abbreviations are HOUR, HOURS, HOURLY, HR. The starting subperiod *s* is in hours.

MINUTE

specifies minute intervals. Abbreviations are MINUTE, MINUTES, MIN. The starting subperiod *s* is in minutes.

SECOND

specifies second intervals. Abbreviations are SECOND, SECONDS, SEC. The starting subperiod *s* is in seconds.

Examples of Interval Specifications

Table 3.1 shows examples of different kinds of interval specifications.

Table 3.1. Examples of Intervals

Name	Kind of Interval
YEAR	years starting in January
YEAR.10	fiscal years starting in October
YEAR2.7	biennial intervals starting in July of even years
YEAR2.19	biennial intervals starting in July of odd years
YEAR4.11	four-year intervals starting in November of leap years (frequency of U.S. presidential elections)
YEAR4.35	four-year intervals starting in November of even years between leap years (frequency of U.S. midterm elections)
WEEK	weekly intervals starting on Sundays
WEEK2	biweekly intervals starting on first Sundays
WEEK1.1	same as WEEK
WEEK.2	weekly intervals starting on Mondays
WEEK6.3	six-week intervals starting on first Tuesdays
WEEK6.11	six-week intervals starting on second Wednesdays
WEEKDAY	daily with Friday-Saturday-Sunday counted as the same day (five-day work week with a Saturday-Sunday weekend)
WEEKDAY17W	same as WEEKDAY
WEEKDAY67W	daily with Thursday-Friday-Saturday counted as the same day (five-day work week with a Friday-Saturday weekend)
WEEKDAY1W	daily with Saturday-Sunday counted as the same day (six-day work week with a Sunday weekend)
WEEKDAY3.2	three-weekday intervals (with Friday-Saturday-Sunday counted as one weekday) with the cycle three-weekday periods aligned to Monday 4 Jan 1960
HOUR8.7	eight-hour intervals starting at 6 a.m., 2 p.m., and 10 p.m. (might be used for work shifts)

Date and Datetime Informats

Table 3.2 summarizes the SAS date and datetime informats available in SAS software. See Chapter 2, "Working with Time Series Data," for a discussion of the use of date and datetime informats. Refer to *SAS Language: Reference* for a complete description of these informats.

For each informat, Table 3.2 shows an example of a date or datetime value written in the style that the informat is designed to read. The date 17 October 1991 and the time 2:25:32 p.m. are used for the example in all cases. table informats shows the width range allowed by the informat and shows the default width in parenthesis.

Table 3.2. SAS Date and Datetime Informats

Informant Example	Description	Width Range	Default Width
DATE*w.* 17oct91	day, month abbreviation, and year: *ddMONyy*	7-32	7
DATETIME*w.d* 17oct91:14:45:32	date and time: *ddMONyy:hh:mm:ss*	13-40	18
DDMMYY*w.* 17/10/91	day, month, year: *ddmmyy*, *dd/mm/yy*, *dd-mm-yy*, or *dd mm yy*	6-32	6
JULIAN*w.* 91290	year and day of year (Julian dates): *yyddd*	5-32	5
MMDDYY*w.* 10/17/91	month, day, year: *mmddyy*, *mm/dd/yy*, *mm-dd-yy*, or *mm dd yy*	6-32	6
MONYY*w.* Oct91	month abbreviation and year	5-32	5
NENGO*w.* H.03/10/17	Japanese Nengo notation	7-32	10
TIME*w.d* 14:45:32	hours, minutes, seconds: *hh:mm:ss* or hours, minutes: *hh:mm*.	5-32	8
YYMMDD*w.* 91/10/17	year, month, day: *yymmdd*, *yy/mm/dd*, *yy-mm-dd*, or *yy mm dd*	6-32	6
YYQ*w.* 91Q4	year and quarter of year: *yyQq*	4-32	4

Date, Time, and Datetime Formats

The SAS date and datetime formats are summarized in Table 3.3 and Table 3.4. A width value can be specified with each format. The tables list the range of width values allowed and the default width value for each format.

The notation used by a format is abbreviated in different ways depending on the width option used. For example, the format MMDDYY8. writes the date 17 October 1991 as 10/17/91, while the format MMDDYY6. writes this date as 101791. In particular, formats that display the year show two- or four-digit year values depending on the width option. The examples shown in the tables are for the default width.

Refer to *SAS Language: Reference* for a complete description of these formats, including the variations of the formats produced by different width options. See Chapter 2, "Working with Time Series Data," for a discussion of the use of date and datetime formats.

Date Formats

Table 3.3 lists the date formats available in SAS software. For each format, an example is shown of a date value in the notation produced by the format. The date '17OCT91'D is used as the example.

Table 3.3. SAS Date Formats

Format Example	Description	Width Range	Default Width
DATE*w.* 17oct91	day, month abbreviation, year: *ddMONyy*	5-9	7
DAY*w.* 17	day of month	2-32	2
DDMMYY*w.* 17/10/91	day, month, year: *dd/mm/yy*	2-8	8
DOWNAME*w.* Thursday	name of day of the week	1-32	9
JULDAY*w.* 290	day of year	3-32	3
JULIAN*w.* 91290	year and day of year: *yyddd*	5-7	5
MMDDYY*w.* 10/17/91	month, day, year: *mm/dd/yy*	2-8	8
MMYY*w.*	month and year: *mmMyy*	5-32	7

Table 3.3. (continued)

Format Example	Description	Width Range	Default Width
10M1991			
MMYYC*w.* 10:1991	month and year: *mm:yy*	5-32	7
MMYYD*w.* 10-1991	month and year: *mm-yy*	5-32	7
MMYYP*w.* 10.1991	month and year: *mm.yy*	5-32	7
MMYYS*w.* 10/1991	month and year: *mm/yy*	5-32	7
MMYYN*w.* 101991	month and year: *mmyy*	5-32	6
MONNAME*w.* October	name of month	1-32	9
MONTH*w.* 10	month of year	1-32	2
MONYY*w.* OCT91	month abbreviation and year: *MONyy*	5-7	5
QTR*w.* 4	quarter of year	1-32	1
QTRR*w.* IV	quarter in Roman numerals	3-32	3
NENGO*w.* H.03/10/17	Japanese Nengo notation	2-10	10
WEEKDATE*w.* Thursday, October 17, 1991	*day-of-week, month-name dd, yy*	3-37	29
WEEKDATX*w.* Thursday, 17 October 1991	*day-of-week, dd month-name yy*	3-37	29
WEEKDAY*w.* 5	day of week	1-32	1

Table 3.3. (continued)

Format Example	Description	Width Range	Default Width
WORDDATE*w.* October 17, 1991	*month-name dd, yy*	3-32	18
WORDDATX*w.* 17 October 1991	*dd month-name yy*	3-32	18
YEAR*w.* 1991	year	2-32	4
YYMM*w.* 1991M10	year and month: *yyMmm*	5-32	7
YYMMC*w.* 1991:10	year and month: *yy:mm*	5-32	7
YYMMD*w.* 1991-10	year and month: *yy-mm*	5-32	7
YYMMP*w.* 1991.10	year and month: *yy.mm*	5-32	7
YYMMS*w.* 1991/10	year and month: *yy/mm*	5-32	7
YYMMN*w.* 199110	year and month: *yymm*	5-32	7
YYMON*w.* 1991OCT	year and month abbreviation: *yyMON*	5-32	7
YYMMDD*w.* 91/10/17	year, month, day: *yy/mm/dd*	2-8	8
YYQ*w.* 91Q4	year and quarter: *yyQq*	4-6	4
YYQC*w.* 1991:4	year and quarter: *yy:q*	4-32	6
YYQD*w.* 1991-4	year and quarter: *yy-q*	4-32	6
YYQP*w.* 1991.4	year and quarter: *yy.q*	4-32	6

Table 3.3. (continued)

Format Example	Description	Width Range	Default Width
YYQS*w.* 1991/4	year and quarter: *yy/q*	4-32	6
YYQN*w.* 19914	year and quarter: *yyq*	3-32	5
YYQR*w.* 1991QIV	year and quarter in Roman numerals: *yyQrr*	6-32	8
YYQRC*w.* 1991:IV	year and quarter in Roman numerals: *yy:rr*	6-32	8
YYQRD*w.* 1991-IV	year and quarter in Roman numerals: *yy-rr*	6-32	8
YYQRP*w.* 1991.IV	year and quarter in Roman numerals: *yy.rr*	6-32	8
YYQRS*w.* 1991/IV	year and quarter in Roman numerals: *yy/rr*	6-32	8
YYQRN*w.* 1991IV	year and quarter in Roman numerals: *yyrr*	6-32	8

Datetime and Time Formats

Table 3.4 lists the datetime and time formats available. For each format, an example is shown of a datetime value in the notation produced by the format. The datetime value '17OCT91:14:25:32'D is used as the example.

Table 3.4. SAS Datetime and Time Formats

Format Example	Description	Width Range	Default Width
DATETIME*w.d* 17OCT91:14:25:32	*ddMONyy:hh:mm:ss*	7-40	16
HHMM*w.d* 14:25	hour and minute: *hh:mm*	2-20	5
HOUR*w.d* 14	hour	2-20	2
MMSS*w.d* 25:32	minutes and seconds: *mm:ss*	2-20	5
TIME*w.d* 14:25:32	time of day: *hh:mm:ss*	2-20	8
TOD*w.* 14:25:32	time of day: *hh:mm:ss*	2-20	8

Alignment of SAS Dates

SAS date values used to identify time series observations produced by SAS/ETS procedures are normally aligned with the beginning of the time intervals corresponding to the observations. For example, for monthly data for 1994, the date values identifying the observations are 1Jan94, 1Feb94, 1Mar94, . . . , 1Dec94.

However, for some applications it may be preferable to use end of period dates, such as 31Jan94, 28Feb94, 31Mar94, . . . , 31Dec94. For other applications, such as plotting time series, it may be more convenient to use interval midpoint dates to identify the observations.

SAS/ETS procedures provide an ALIGN= option to control the alignment of dates for output time series observations. Procedures supporting the ALIGN= option are ARIMA, DATASOURCE, EXPAND, and FORECAST.

ALIGN=

The ALIGN= option allows the following values:

BEGINNING	Specifies that dates are aligned to the start of the interval. This is the default. BEGINNING can be abbreviated as BEGIN, BEG, or B.
MIDDLE	Specifies that dates are aligned to the interval midpoint. MIDDLE can be abbreviated as MID or M.
ENDING	Specifies that dates are aligned to the end of the interval. ENDING can be abbreviated as END or E.

The ALIGN= option can be specified on the PROC DATASOURCE statement, on the PROC EXPAND statement, on the PROC FORECAST statement, and on the FORECAST statement of the ARIMA procedure.

Date, Time, and Datetime Functions

The SAS system provides functions to perform calculations with SAS date, time, and datetime values. SAS date, time, and datetime functions are used to:

- compute date, time, and datetime values from calendar and time-of-day values.
- compute calendar and time-of-day values from date and datetime values.
- convert between date, time, and datetime values.
- perform calculations involving time intervals.

SAS date, time, and datetime functions are listed in alphabetical order in the following. Refer to *SAS Language: Reference* for a complete description of these functions.

SAS Date, Time, and Datetime Functions

DATE()
returns today's date as a SAS date value.

DATEJUL(*yyddd* **)**
returns the Julian date for a SAS date value.

DATEPART(*datetime* **)**
returns the date part of a SAS datetime value as a date value.

DATETIME()
returns the current date and time of day.

DAY(*date* **)**
returns the day of the month from a SAS date value.

DHMS(*date, hour, minute, second* **)**
returns a SAS datetime value for date, hour, minute, and second values.

HMS(*hour, minute, second* **)**
returns a SAS time value for hour, minute, and second values.

HOUR(*datetime* **)**
returns the hour from a SAS datetime or time value.

INTCK(*interval, date1, date2* **)**
returns the number of boundaries of intervals of the given kind that lie between the two date or datetime values.

INTNX(*interval, date, n* **<**, *'alignment'* **>)**
returns the date or datetime value of the beginning of the interval that is *n* intervals from the interval that contains the given date or datetime value. The optional alignment argument specifies that the returned date is aligned to either the beginning, middle, or end of the interval. Beginning is the default.

JULDATE(*date* **)**
returns the Julian date from a SAS date value.

MDY(*month, day, year* **)**
returns a SAS date value for month, day, and year values.

MINUTE(*datetime* **)**
returns the minute from a SAS time or datetime value.

MONTH(*date* **)**
returns the month of the year from a SAS date value.

QTR(*date* **)**
returns the quarter of the year from a SAS date value.

SECOND(*date* **)**
returns the second from a SAS time or datetime value.

TIME()
returns the current time of day.

TIMEPART(*datetime* **)**
returns the time part of a SAS datetime value.

TODAY()
returns the current date as a SAS date value. (TODAY is another name for the DATE function.)

WEEKDAY(*date* **)**
returns the day of the week from a SAS date value.

YEAR(*date* **)**
returns the year from a SAS date value.

YYQ(*year, quarter* **)**
returns a SAS date value for year and quarter values.