

SAS® Routines for Calendar and Clock Elements

Rick Aster, Valley Forge, PA

Abstract

Data values related to time are conventionally measured and reported as combinations of elements of the calendar and clock — for example, a date may be written as the month, day, and year. These composite values can be more complicated to work with than data values that are written as a single measurement. SAS provides a set of functions, formats, and informats that work with specific combinations of calendar and clock elements. International date formats let you work with time data in many different languages. In version 8, you can go beyond the built-in routines and create picture formats to write the date and time of day in any combination of calendar and clock elements.

Linear Time

In theory, linear time is as easy to measure as linear distance. All you need is a unit of measure and a point of reference, and you can make simple numeric measurements of any point in time. For a few applications, time is actually measured this way. The 100-meter event in track and field is an example; the race time is counted in seconds, starting from the starting signal at the beginning of the race.

For most applications, though, time is too large to measure this way. Instead, we use the standards of a calendar and a clock to divide time into units of various sizes: years, weeks, hours, and such. A time measurement, then, typically uses a combination of units. In U.S. business writing, a certain date is written as May 21, 2001 — a month, day, and year. A time of day may be written in three parts, hours, minutes, and a day half, as 1:00 p.m.

The various combinations of units provide a comfortable way of working with time measurements, but they make computation difficult. In order to say whether March 5 comes before or after September 30, you have to refer to a calendar — at least in your head.

Another complicating factor is that the same time measurement can be written in various ways. For example, 5/21/01, 05/21/2001, and 21 May 2001 all refer to the same date.

Time measurements in SAS

To make time measurements easier for a computer program to work with, it is necessary to convert the various conventional forms of notation back into simple linear measurements, which count a single unit from a single point of reference. SAS provides three specific standards to follow in measuring time.

- A SAS *date value* is particularly suited to measuring dates. It counts time as days since the beginning of 1960. A SAS date constant can be written with a day, three-letter month abbreviation, and year, quoted and followed by the letter D, such as '21MAY2001'D.
- A SAS *datetime value* is a better way to measure the date and the time of day in a single value. It counts time as seconds since the beginning of 1960. A SAS datetime constant can be written as a date and time of day, quoted and followed by the letters DT, such as '21MAY2001 13:00'DT.
- A SAS *time value* indicates the time of day. It counts time as seconds since midnight. A SAS time constant is a time of day, quoted and followed by the letter T, such as '13:00'T or '1:00PM'T.

SAS routines convert between these measurements and the calendar and clock elements that are conventionally used for measuring and writing the date and the time of day.

Although a SAS time value is defined as the number of seconds since midnight, that is not very different from a measure of elapsed time in hours, minutes, and seconds from any point of

reference. Therefore, most of the routines that work with SAS time values also work with measures of elapsed time.

Calendar and Clock Elements

The SAS routines that convert time values are functions, informats, and formats. The routines work with the calendar and clock elements listed in Table 1. The functions work only with numeric values for calendar and clock elements; informats and formats also read and write some calendar and clock elements as words.

Functions

Several SAS functions convert between time units. There are functions to construct a SAS date value, SAS time value, and SAS datetime value from component parts. Other functions extract parts. Two functions extract a SAS date value or a SAS time value from a SAS datetime value. Other functions extract specific calendar and clock elements. These functions are listed in Table 2.

The answers that these functions provide are the result of simple, familiar calendar and clock logic. For example, YEAR('31JUL1998'D) is 1998; MONTH('31JUL1998'D) is 7; MDY(7, 31, 1998) is '31JUL1998'D.

Table 1. Parts Used In SAS Routines

Year	Years of the Gregorian calendar, from 1582 to 19999.
2-digit year	Two-digit year numbers in input, year values from 0 to 99, fall into the century defined by the YEARCUTOFF= system option. The value of the YEARCUTOFF= option is the oldest year that can be represented in two digits. For example, with the version 8 default YEARCUTOFF=1950, two-digit years fall into the period from 1950 to 2049. The year number 50 means 1950; the year number 0 means 2000; the year number 49 means 2049. In output, any year can be written with its last two digits.
Quarter	Numbered sequentially in the year.
Month	Numbered sequentially in the year.
Month name	Written with an initial capital letter.
3-letter month abbreviation	The first three letters of the month name. In output, this is written with an initial capital if it is followed by a space, or in all capital letters if it is concatenated to another element of the date.
Month initial	The first letter of the name of the month.
Week	Numbered sequentially in the year, from 0 or 1 to 52 or 53, starting week 1 on the first Sunday of the year.
Weekday number	The day of the week, numbered sequentially from Sunday = 1 to Saturday = 7.
Weekday name	Written with an initial capital letter.
3-letter weekday abbreviation	The first three letters of the weekday name. In output, this is written with an initial capital.
Day	Numbered sequentially in the month.
Day of year	Numbered sequentially in the year.
SAS date	Days elapsed since the beginning of 1960.
Day half	The abbreviations AM and PM.
Hour	The hour of the day, numbered from 0 to 23.
Hour of 12-hour clock	The hour as displayed on the clock: 12, then 1 to 12, then 1 to 11.
Minute	The minute of the hour, from 0 to 59.
Second	The second of the minute, from 0 to 59.
SAS time	Seconds elapsed since midnight.
SAS datetime	Seconds elapsed since the beginning of 1960.
Punctuation	The comma, slash, hyphen, period, and other punctuation symbols are used in output fields and can appear in input fields.

Time computation

The INTNX and INTCK functions do time computation based on fixed time intervals. For example, the INTCK function can tell you the number of days elapsed between two SAS date values; the INTNX function can tell you what day yesterday was from a SAS date value that says what day today is.

The first argument of either function is a time interval code. This is a character value that indicates what time interval and what kind of data value the function works with. For example, the code MONTH indicates that the function works with SAS date values and uses a month as its time interval. This INTNX function call tells you what month occurs 12 months after May 18, 2001:

```
INTNX('MONTH', '18MAY2001'D, 12)
```

Table 2. Functions for Time Conversion

Function Call	Values
MDY (MONTH, DAY, YEAR)	Arguments: MONTH, a month number; DAY, a day of the month; YEAR, a year number. Requirement: The arguments must correspond to a date of the Gregorian calendar. Return value: The SAS date value of the day that is identified by the indicated year, month, and day.
YYQ (YEAR, QUARTER)	Arguments: YEAR, a year number; QUARTER, a quarter number. Return value: The SAS date value of the first day of the indicated quarter.
HMS (HOUR, MINUTE, SECOND)	Arguments: HOUR, an hour number of the 24-hour clock; MINUTE, a minute number, from 0 to 59; SECOND, a second number, from 0 to 59. Return value: The time of day indicated by the hour, minute, and second, as a SAS time value.
DHMS (DATE, HOUR, MINUTE, SECOND)	Arguments: DATE, a SAS date value; HOUR, an hour number of the 24-hour clock, from 0 to 23; MINUTE, a minute number, from 0 to 59; SECOND, a second number, from 0 to 59. Return value: The SAS datetime value of the indicated date and time of day.
DATEPART (DATETIME)	Argument: DATETIME, a SAS datetime value. Return value: The SAS date value of the day indicated in the SAS datetime value. The return value is a whole number.
TIMEPART (DATETIME)	Argument: DATETIME, a SAS datetime value. Return value: The SAS time value of the time of day indicated in the SAS datetime value.
YEAR (DATE)	Argument: DATE, a SAS date value. Return value: The year of the SAS date value.
QTR (DATE)	Argument: DATE, a SAS date value. Return value: The quarter of the SAS date value.
MONTH (DATE)	Argument: DATE, a SAS date value. Return value: The month number of the SAS date value.
WEEKDAY (DATE)	Argument: DATE, a SAS date value. Return value: The number of the day of the week of the SAS date value. Days of the week are counted starting with Sunday = 1.
DAY (DATE)	Argument: DATE, a SAS date value. Return value: The day of the month of the SAS date value.
HOUR (TIME)	Argument: TIME, the time of day indicated by a SAS time value or SAS datetime value. Return value: The hour of the 24-hour clock time. The return value is a whole number.
MINUTE (TIME)	Argument: TIME, the time of day indicated by a SAS time value or SAS datetime value. Return value: The minute of the clock time. The return value is a whole number.
SECOND (TIME)	Argument: TIME, the time of day indicated by a SAS time value or SAS datetime value. Return value: The second of the clock time.

The function returns the SAS date value '01MAY2002'D.

The INTCK function counts the number of intervals elapsed between two time measurements. For example,

```
INTCK('QTR', '01JAN2001'D, '31MAR2001'D)
```

returns 0, because the two dates are in the same quarter.

The available time interval code names are listed in Table 3. These names can be modified with multiplier and shift arguments. For example, WEEK2 (or WEEK2.1) is a two-week interval; WEEK2.2 is also a two-week interval, but starting one week later. The same time intervals are used in SAS/GRAPH to define time axes.

Informats and Formats

The primary use of informats and formats is with text files. An informat converts the value of a text field to a data value that can be used in a SAS program. A format converts a data value to a text value that can be written to a file or displayed.

Informats

SAS informats that read dates produce SAS date values. To read date fields, informats have to find the calendar elements, such as year, month, and day, in the date field, and then do the calendar computations to determine the number of days since the beginning of 1960. The resulting value is a SAS date value. Informats for SAS date values are shown in Table 4.

As an example, consider the way the MMDDYY informat reads the text value 05-21-2001. It first extracts the calendar elements that are contained in the text value: the month 05, the day 21, and the year 2001. Checking the calendar, it finds that the year 2001 starts 14,976 days after the beginning of 1960; in 2001, month 5 (May) starts 120 days after that; and day 21 of the month is 20 days after the beginning of the month. The sum of these numbers, 14,976, 120, and 20, is 15,116, and this is the SAS date value that the informat produces.

Table 3. Time Interval Names

Interval Length	Starting Point	Interval Name	Value
Second		SECOND	SAS datetime
Minute	According to clock	MINUTE	SAS time
Hour		HOUR	
Day	Midnight	DAY	SAS date
		DTDAY	SAS datetime
Selected days of the week ¹	Each selected day	WEEKDAY	SAS date
		DTWEEKDAY	SAS datetime
Week	Sunday	WEEK	SAS date
		DTWEEK	SAS datetime
Third month (8–11 days)	Days 1, 11, 21 of month	TENDAY	SAS date
		DTTENDAY	SAS datetime
Half month (13–16 days)	Days 1, 16 of month	SEMIMONTH	SAS date
		DTSEMIMONTH	SAS datetime
Month (28–31 days)	Day 1 of month	MONTH	SAS date
		DTMONTH	SAS datetime
Quarter (90–92 days)	Jan. 1, Apr. 1, Jul. 1, Oct. 1	QTR	SAS date
		DTQTR	SAS datetime
Half year (181–184 days)	January 1, July 1	SEMIYEAR	SAS date
		DTSEMIYEAR	SAS datetime
Year (365–366 days)	First day of year	YEAR	SAS date
		DTYEAR	SAS datetime

¹ WEEKDAY_n...W, DTWEEKDAY_n...W: Select days of the week with digits between WEEKDAY and W in the interval name. The digits 1 through 7 represent the days Sunday through Saturday. For example, WEEKDAY246W or DTWEEKDAY246W selects Monday, Wednesday, and Friday. WEEKDAY is the same as WEEKDAY23456W, and DTWEEKDAY is the same as DTWEEKDAY23456W, to select the days of the week Monday through Friday.

Table 4. Informat for SAS Date Values

Informat	Width	Field Elements	Examples
DATE w .	$7 \leq w \leq 32$	day, 3-letter month abbreviation, year	01JUN00 01JUN2000
YYMMDD w .	$6 \leq w \leq 32$	year, month, day	1999.12.31
MMDDYY w .	$6 \leq w \leq 32$	month, day, year	5/27/05
DDMMYY w .	$6 \leq w \leq 32$	day, month, year	150708
MONYY w .	$5 \leq w \leq 32$	3-letter month abbreviation, year	mar2000
YYMMN w .	$4 \leq w \leq 32$	year, month (no punctuation)	9711
YYQ w .	$4 \leq w \leq 32$	year, "Q", quarter	2001Q1
JULIAN w .	$5 \leq w \leq 32$	year, day of year	2000 121

Table 5. DATETIME and TIME Informat

Informat	Width	Field Elements	Value
TIME w .	$5 \leq w \leq 32$	hour, minute, optional second	SAS time
DATETIME w .	$13 \leq w \leq 40$	day, 3-letter month abbreviation, year, and hour, minute, optional second or hour of 12-hour clock, minute, optional second, day half	SAS datetime

Table 6. Formats for SAS Date Values

Format	Field Elements	Examples
DATE5.	day, 3-letter month abbreviation	15JUN
DATE7.	day, 3-letter month abbreviation, 2-digit year	14JUN00 15AUG75
DATE9.	day, 3-letter month abbr., year	14JUN2000
MONYY5.	3-letter month abbr., 2-digit year	APR00
MONYY7.	3-letter month abbreviation, year	APR2000
YYMON5.	2-digit year, 3-letter month abbr.	00APR
YYMON7.	year, 3-letter month abbreviation	2000APR
YYMM5.	2-digit year, "M", month	01M07
YYMM7.	year, "M", month	2001M07
YEAR2.	2-digit year	82
YYMMDD2.		00
YEAR4.	year	1582
YYMMDD4.	2-digit year, month	0010
YYMMDD5.	2-digit year, "-", month	00-10
YYMMDD6.	2-digit year, month, day	001006
YYMMDD8.	2-digit year, "-", month, "-", day	00-10-06
YYMMDD10.	year, "-", month, "-", day	2000-10-06
MONTH2.	month	3
MMDDYY2.	month	08
MMDDYY4.	month, day	0818
MMDDYY5.	month, "/", day	08/18
MMDDYY6.	month, day, 2-digit year	081800
MMDDYY8.	month, "/", day, "/", 2-digit year	08/18/00
MMDDYY10.	month, "/", day, "/", year	08/18/2000
DAY2.	day	1
DDMMYY2.	day	01 21
DDMMYY4.	day, month	2103

Format	Field Elements	Examples
DDMMYY5.	day, "/", month	21/03
DDMMYY6.	day, month, 2-digit year	210300
DDMMYY8.	day, "/", month, "/", 2-digit year	21/03/00
DDMMYY10.	day, "/", month, "/", year	21/03/2000
YYQ4.	2-digit year, "Q", quarter	01Q1
YYQ6.	year, "Q", quarter	2001Q1
YYQR6.	year, "Q", quarter as uppercase Roman numeral	01QI
YYQR8.	2-digit year, "Q", quarter as uppercase Roman numeral	2001QI
QTR1.	quarter	3
QTRR3.	quarter as uppercase Roman numeral	III
WEEKDAY1.	weekday number	6
DOWNAME1.	weekday name (first letter)	F
DOWNAME2.	weekday name (first 2 letters)	Fr
DOWNAME3.	3-letter weekday abbreviation	Fri
WEEKDATE3.		Sat
WEEKDATX3.		Sun
DOWNAME9.	weekday name	Monday
WEEKDATE9.		Tuesday
WEEKDATX9.		Wednesday
JULIAN5.	2-digit year, day of year	00121
JULIAN7.	year, day of year	2000121
JULDAY3.	day of year	366
MONNAME1.	month initial	F
MONNAME3.	3-letter month abbreviation	Feb
WORDDATE3.		Apr
WORDDATX3.		Oct
MONNAME9.	month name	September
WORDDATE9.		October
WORDDATX9.		

Format	Field Elements	Examples
WORDDATE12.	3-letter month abbreviation, space, day, comma, space, year	Dec 9, 1777
WORDDATE18.	month name, space, day, comma, space, year	December 9, 1777
WEEKDATE15.	3-letter weekday abbreviation, comma, space, 3-letter month abbreviation, space, day, comma, space, 2-digit year	Tue, Dec 9, 77
WEEKDATE17.	3-letter weekday abbreviation, comma, space, 3-letter month abbreviation, space, day, comma, space, year	Tue, Dec 9, 1777
WEEKDATE23.	weekday name, comma, space, 3-letter month abbreviation, space, day, comma, space, year	Tuesday, Dec 9, 1777
WEEKDATE29.	weekday name, comma, space, month name, space, day, space, year	Tuesday, December 9, 1777
WORDDATX12.	day, space, 3-letter month abbreviation, space, year	9 Dec 1777
WORDDATX18.	day, space, month name, space, year	9 December 1777
WEEKDATX15.	3-letter weekday abbreviation, comma, space, day, space, 3-letter month abbreviation, space, 2-digit year	Tue, 9 Dec 77
WEEKDATX17.	3-letter weekday abbreviation, comma, space, day, space, 3-letter month abbreviation, space, year	Tue, 9 Dec 1777
WEEKDATX23.	weekday name, comma, space, day, space, 3-letter month abbreviation, space, year	Tuesday, 12 Oct 1777
WEEKDATX29.	weekday name, comma, space, day, space, month name, space, year	Tuesday, 12 October 1777

SAS also has informats for SAS datetime values and SAS time values. These informats are shown in Table 5. They generate values measured in seconds in much the same way that SAS date informats generate values measured in days.

Formats

Formats have to do the reverse of what informats do. They use the calendar and clock to compute the elements that are implied in a value, and then write those elements. They write elements either as numerals or as words, usually adding punctuation characters between them. Most informats have a corresponding format of the same name. If a format writes a field, the corresponding informat can read the field and generate the same value.

The way a format writes a value depends on the width argument of the format. At its full width, a format includes all of its elements; with a smaller width, it may be limited to just one or two elements, or it may abbreviate some of them. Table 6 shows formats that write SAS date values with examples of output at various widths.

Some formats can write both SAS time values and SAS datetime values; others work with only one kind of value or the other. The TIME and HOUR formats write only SAS time values. The

DATETIME and DATEAMPM formats write only SAS datetime values. Finally, MMSS is a format intended for use only with measurements of elapsed time. These formats are listed in table 7.

International Date Formats

Most of the world follows the same calendar and clock standards, but different nations have different ways of writing some of the calendar and clock elements. International date formats make it possible to write time measurements in the languages and notational conventions of most of the countries of the world. The corresponding international date informats make it possible to read time measurements in various languages.

The names of international date formats are constructed with a three-letter prefix that indicates a language or dialect; the root DF; and a two- or three-letter suffix that indicates the calendar and clock elements involved. For example, in the format AFRDFDD, the prefix AFR stands for the Afrikaans language. The suffix DD indicates that the format writes the same calendar elements as the DDMMYY format: the day, month, and year numbers.

Table 7. Formats for Other Time Measurements

Format	Field Elements	Examples
HOUR2.	hour	5
TIME2.		17
HHMM2.		23
TOD2.	hour	05
TIME5.	hour, “.”, minute	0:00 23:59
TOD5.	hour, “.”, minute	00:00 23:59
TIME8.	hour, “.”, minute, “.”, second	0:00:00 15:45:00
TOD8.	hour, “.”, minute, “.”, second	00:00:00 15:45:00
HOURw.d	hours ($2 \leq w \leq 20$, $d < w - 2$)	5 23.99999
TIMEw.d	hour, “.”, minute, “.”, seconds ($10 \leq w \leq 20$, $d = w - 9$)	9:38:21.004 23:59:59.9
TODw.d	hour, “.”, minute, “.”, seconds $10 \leq w \leq 20$, $d = w - 9$)	09:38:21.004 23:59:59.9
HHMM4.	hour, minute	9:31 1545
HHMMw.d	hour, “.”, minutes ($5 \leq w \leq 20$, $d = w - 6$)	09:31.25
TIMEAMPM2.	day half	AM PM
TIMEAMPM5.	hour of 12-hour clock, space, day half	11 AM 9 PM
TIMEAMPM8.	hour of 12-hour clock, “.”, minute, space, day half	11:00 AM
TIMEAMPMw.d	hour of 12-hour clock, “.”, minute, “.”, seconds, space, day half ($11 \leq w \leq 20$, $d = w - 12$)	10:32:07.420 AM
DATETIME7.	elements of a DATE7. field	11JAN00
DATEAMPM7.		
DATETIME10.	elements of a DATE7. field, “.”, hour	11JAN00:14
DATEAMPM10.		
DATETIMEw.d	elements of a DATE7. field, “.”, elements of a TOD field ($10 \leq w \leq 40$, $d = w - 17$)	11JAN00:14:30
DATEAMPMw.d	elements of a DATE7. field, “.”, elements of a TIMEAMPM field ($13 \leq w \leq 40$, $d = w - 20$)	11JAN00:02 PM 11JAN00:02:30 PM
MMSS2.	minutes	7
MMSS4.	minutes, seconds	0709
MMSSw.d	minutes, “.”, seconds ($5 \leq w \leq 20$, $d \leq w - 6$)	7:09 7:09.1

Table 8. International Date Format Languages

Prefix	Language Name
AFR	Afrikaans
CAT	Catalan
CRO	Croatian
CSY	Czech
DAN	Danish
DES	Swiss_German
DEU	German
ENG	English
ESP	Spanish
FIN	Finnish
FRA	French
FRS	Swiss_French
HUN	Hungarian
ITA	Italian
MAC	Macedonian
NLD	Dutch
NOR	Norwegian
POL	Polish
PTG	Portuguese
RUS	Russian
SLO	Slovenian
SVE	Swedish
EUR	Selected in the DFLANG= system option

Table 9. International Date Format Suffixes

Suffix	Equivalent Format
DD	DDMMYY
DE	DATE
DN	WEEKDAY
DT	DATETIME
DWN	DOWNNAME
MN	MONNAME
MY	MONYY
WDX	WORDDATX
WKX	WEEKDATX

Table 10. International Date Informat Suffixes

Suffix	Equivalent Informat
DE	DATE
DT	DATETIME
MY	MONYY

The EUR prefix for an international date format does not indicate any specific language. Instead, it lets you set the language of the format by using a system option, DFLANG=. Use this feature to create a SAS program that can write dates in various different languages.

Table 8 shows languages and prefixes that can be used for international date formats. The set of available languages is limited to ones that use the Roman alphabet. Use the prefix as the first part of the format name. Alternatively, use the EUR prefix in the format name, and use the language name as the value of the DFLANG= system option. Table 9 shows the suffixes that can be used for international date formats.

Names of international date informats are constructed in the same way. Table 10 shows the suffixes that can be used for international date informats.

International date formats with the ENG prefix are identical to the corresponding base formats. For example, the format ENGDFMN is the same as the MONNAME format. Formats with other language prefixes write words in those languages, often with different punctuation. Consider, for example, the SAS date value '01JAN2001'D. Given this value, the MONNAME or ENGDFMN format writes the English name of the month, January; the DEUDFMN writes the name of the month in German, Januar; the ITADFMN format writes it in Italian, Gennaio.

Picture Formats

Beginning with version 7, you can use the FORMAT proc to create picture formats based on calendar and clock elements. These formats can combine calendar and clock elements in any combination along with other text.

Table 11. Picture Directives

Calendar or Clock Element	Directive ¹	Examples
The four-digit year	%Y	2001, 2016
The two-digit year ¹	%y	1, 16
	%0y	01, 16
The month number ¹	%m	3, 7, 11
	%0m	03, 07, 11
The month name ²	%B	September
The month name, abbreviated ²	%b	Apr, Aug, Oct
The week number of the year (starting week 1 on the first Sunday of the year) ¹	%U	1, 14, 27, 40
	%0U	01, 14, 27, 40
The day of the year ¹	%j	1, 92, 184
	%0j	001, 092, 184
The day of the month ¹	%d	5, 12, 19, 26
	%0d	05, 12, 19, 26
The number of the day of the week (starting from Sunday=1)	%w	2, 3, 4, 5, 6
The name of the day of the week ²	%A	Sunday
The name of the day of the week, abbreviated ²	%a	Mon, Tue, Wed
The day half ²	%p	AM, PM
The hour of the 24-hour clock ¹	%H	9, 17
	%0H	09, 17
The hour of the 12-hour clock ¹	%I	9, 5
	%0I	09, 05
The minute ¹	%M	0, 15, 30, 45
	%0M	00, 15, 30, 45
The second ¹	%S	0, 15, 30, 45
	%0S	00, 15, 30, 45
The character % (percent sign)	%%	%

¹ Picture directives that contain a 0 write numbers with leading zeroes. Other picture directives write numbers without leading zeroes.

² These calendar and clock elements are localized according to the LANGUAGE= picture option or the DFLANG= system option.

These picture formats are not the same as the older kind of picture formats that write quantitative values. Instead of using pictures that contain digit selectors, their pictures contain code sequences called *picture directives* as part of their text.

The PICTURE Statement

The PICTURE statement contains the name of the new format; a range, usually OTHER; a picture that shows how the calendar and clock elements are used in the format; and one or more picture options. The one required option is the DATATYPE= option. The value for this option is DATE, DATETIME, or TIME, indicating that the format works with a SAS date, SAS datetime, or SAS time value, respectively.

The PICTURE statement is combined with a PROC FORMAT statement to produce a complete step, for example:

```
PROC FORMAT;
  PICTURE WORDYMD
    OTHER = 'Year %Y Month %m Day %d '
    (DATATYPE=DATE);
RUN;
```

This step creates the format WORDYMD, which writes SAS date values.

Picture Directives

When you create a picture format, the picture directives begin with a percent sign. Each picture directive represents a specific calendar or clock element, as shown in table 11. When you use the format, a routine called the *picture processor* substitutes the text of the calendar or clock element for the picture directive. The directives in WORDYMD example generate the year, month number, and day. This format writes dates like this:

```
Year 1999 Month 9 Day 9
```

In the PICTURE statement, it is important to enclose the picture in single quotes, so that the macro processor does not mistake the picture directives for macro calls. Also, write extra spaces at the end of the picture so that it is at least as long as the formatted value it generates. The extra spaces tell the picture processor how much space to allow for the resulting text value.

Localization of Picture Formats

Picture formats can write dates in any of the date format languages. They are localized when they execute according to the value of the DFLANG= system option. That is, they work the same way as international date formats with the EUR prefix. Alternatively, you can permanently set the language of a picture format by using the LANGUAGE= picture option when you create the format. The values for this option are the same as the values you can use with the DFLANG= system option.

Notes

© 2001 Rick Aster. Tables are reproduced from *Professional SAS Programming Logic* by Rick Aster (Breakfast Communications Corp., 2000). © 2000 Rick Aster.

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc.

Contact:

Rick Aster
 Breakfast Communications Corp.
 P.O. Box 176
 Paoli, PA 19301-0176
 Email: ra@globalstatements.com
 Web site: <http://www.globalstatements.com>

