Contents

About This Book .......................................................................................................................... vii
Acknowledgments ...................................................................................................................... xi
Chapter 1: Introduction to Dates and Times in SAS ................................................................. 1
  1.1 How Does It Work? (January 1, 1960, and Midnight as Zero) .......................................... 1
  1.2 Internal Representation ....................................................................................................... 2
  1.3 External Representation (Basic FORMAT Concepts) ......................................................... 2
  1.4 Date and Time as Numeric Constants in SAS .................................................................... 3
  1.5 Length and Numeric Requirements for Date, Time, and Datetime ................................. 5
  1.6 General SAS Options for Dates ....................................................................................... 7
Chapter 2: Displaying SAS Date, Time, and Datetime Values as Dates and Times as We Know Them ................................................................................................................. 9
  2.1 How Do I Use a Format? ................................................................................................... 10
  2.2 How Many Built-In Formats Are There for Dates and Times? ..................................... 13
  2.3 Date Formats, Justification, and ODS .......................................................................... 13
  2.4 Detailed Discussion of Each Format ............................................................................. 14
    2.4.1 Date Formats ............................................................................................................. 14
    2.4.2 Time Formats .......................................................................................................... 37
    2.4.3 Datetime Formats .................................................................................................... 41
  2.5 Creating Custom Date Formats Using the VALUE Statement of PROC FORMAT .......... 47
  2.6 Creating Custom Date Formats Using the PICTURE Statement of PROC FORMAT ...... 48
  2.7 Creating Custom Formats Using PROC FCMP for Processing .................................... 52
  2.8 The PUT() Function and Formats .................................................................................. 55
Chapter 3: Converting Dates and Times into SAS Date, Time, and Datetime Values .......................................................... 57
  3.1 Avoiding the Two-Digit Year Trap ................................................................. 57
  3.2 Using Informats ............................................................................................ 59
  3.3 The INFORMAT Statement .......................................................................... 59
     3.3.1 Using Informats with the INPUT Statement ........................................... 60
     3.3.2 Informats with the INPUT() Function .................................................... 61
     3.3.3 When the Informat Does Not Match the Data Being Read .................... 62
  3.4 Listing and Discussion of Informats ............................................................ 64
     3.4.1 Date Informats ....................................................................................... 64
     3.4.2 Time Informats ..................................................................................... 73
     3.4.3 Datetime Informats ............................................................................... 78
     3.4.4 The "ANYDATE" Series of Informats ..................................................... 81
     3.4.5 So Why Not Just Use the "ANYDATE" Series of Informats? ................... 86

Chapter 4: ISO 8601 Dates, Times, Datetimes, Durations, and Functions 91
  4.1 What Is ISO 8601? ....................................................................................... 91
  4.2 ISO 8601 Formats ....................................................................................... 92
     4.2.1 ISO Date Formats ............................................................................... 93
     4.2.2 ISO Time Formats ............................................................................. 93
     4.2.3 ISO Datetime Formats ..................................................................... 99
  4.3 ISO 8601 Informats ..................................................................................... 103
     4.3.1 ISO Date Informats ............................................................................ 104
     4.3.2 ISO Time Informats .......................................................................... 105
     4.3.3 ISO Datetime Informats .................................................................... 108
  4.4 Time Zone Functions .................................................................................. 111
     4.4.1 Introduction ....................................................................................... 111
     4.4.2 The TIMEZONE= Option .................................................................. 111
     4.4.3 List of Time Zone Functions ................................................................. 112
Contents

4.5 ISO 8601 Durations and Intervals .................................................................................................................. 116
  4.5.1 ISO Duration and Interval Representations ............................................................................................. 116
  4.5.2 ISO 8601 Duration and Interval Formats ............................................................................................... 117
  4.5.3 ISO 8601 Duration and Interval Informats ............................................................................................. 121
  4.5.4 CALL IS8601_CONVERT ........................................................................................................................ 123

4.6 Conclusion ..................................................................................................................................................... 136

Chapter 5: Date and Time Functions .................................................................................................................. 137

  5.1 Current Date and Time Functions ............................................................................................................. 137
  5.2 Extracting Pieces from SAS Date, Time, and Datetime Values .................................................................. 138
  5.3 Creating Dates, Times, and Datetimes from Numbers or Other Information ......................................... 140
    5.3.1 Introduction ......................................................................................................................................... 140
    5.3.2 List of Functions and Their Descriptions .......................................................................................... 140
  5.4 Calculating Elapsed Time, and the HOLIDAY() Function ......................................................................... 145
    5.4.1 Calculating Elapsed Time with DATDIF() and YRDIF() .................................................................. 145
  5.5 The Basics of SAS Intervals ....................................................................................................................... 149
    5.5.1 The Interval Calculation Functions: INTCK() and INTNX() ............................................................. 151
  5.6 Modifying SAS Intervals ........................................................................................................................... 159
  5.7 Creating Your Own SAS Intervals ............................................................................................................. 169
  5.8 Interval Functions about Intervals ............................................................................................................. 176
    5.8.1 INTFIT(argument-1,argument-2,type) ............................................................................................ 177
    5.8.2 INTFMT('interval','size') ................................................................................................................ 178
    5.8.3 INTGET(argument1,argument2,argument3) .................................................................................... 179
    5.8.4 INTSHIFT('interval') ....................................................................................................................... 180
    5.8.5 INTTEST('interval') ......................................................................................................................... 181
  5.9 Retail Calendar Intervals and Seasonality ................................................................................................. 181
    5.9.1 Retail Calendar Intervals .................................................................................................................. 181
    5.9.2 Seasonality Functions ....................................................................................................................... 183

Chapter 6 Deeper into Dates and Times with SAS ............................................................................................. 185

  6.1 Macro Variables and Dates ......................................................................................................................... 185
    6.1.1 Automatic Macro Variables .............................................................................................................. 185
    6.1.2 Putting Dates into Titles .................................................................................................................... 186
    6.1.3 Using %SYSFUNC() to Create Dates, Times, and Datetimes in Macro Variables ............................ 187
    6.1.4 Using Dates in Macros ...................................................................................................................... 189
Chapter 1: Introduction to Dates and Times in SAS

1.1 How Does It Work? (January 1, 1960, and Midnight as Zero) .................... 1
1.2 Internal Representation .............................................................................. 2
1.3 External Representation (Basic FORMAT Concepts) ............................... 2
1.4 Date and Time as Numeric Constants in SAS ......................................... 3
1.5 Length and Numeric Requirements for Date, Time, and Datetime .......... 5
1.6 General SAS Options for Dates............................................................... 7

In the years that I've been working with SAS and teaching students how to use it, I find that two things consistently confuse those who are new to SAS. First is the default way that the DATA step works. Its implied DO-until-end-of-data generates many "How do I tell it how much data to read and when to stop?" questions. The second most confusing concept in SAS is that of how dates (and times) work within the software. I've seen many misuses of character strings masquerading as dates and/or times over the years, as well as unexpected results due to a failure to understand this fundamental part of SAS.

However, the way that SAS reads, stores, and displays dates and times is only the tip of the iceberg when it comes to the power and flexibility of SAS in handling this information. There is so much more than just having numbers represent date and time values. We'll start with the basics in the first three chapters, and then progress to some more advanced uses of those date and time values, taking advantage of many of the features available in SAS for that purpose.

1.1 How Does It Work? (January 1, 1960, and Midnight as Zero)

SAS counts dates, times, and datetime values separately. The date counter started at zero on January 1, 1960. Any day before January 1, 1960, is a negative number, and any day after that is a positive number. Every day at midnight, the date counter is increased by one. The time counter is measured in seconds and runs from zero (at midnight) to 86,400 (the number of seconds in a day), when it resets to zero. SAS calculates datetimes as the number of seconds since midnight, January 1, 1960. Why January 1, 1960? The founders of SAS wanted to use the approximate birth date of the IBM 370 system to represent the beginning of the modern computing era, and they chose January 1, 1960, as an easy-to-remember approximation.
In deciding whether to use a date, a time, or a datetime, you should consider how you are going to use it. Datetimes are always date and time combined; therefore, if you will not always have a time available for each date, you should strongly consider using separate date and time variables and then calculate a datetime variable from the two components when needed. Normally, attempting to create a datetime without both a date and a time will cause an error, and the result will be a missing value for the datetime. However, in specific circumstances, it is possible to create a datetime value from a date and a missing time (see Section 4.3.3, "ISO Datetime Informats," for an example). In these cases, the time will be set automatically to midnight (0 seconds of the given date). You may want your datetime value to be missing when there is a date but no time available. In these specific circumstances, it is important to keep track of date and time separately. Many programs that handle dates (such as databases and spreadsheets) maintain their dates and times as a numeric value relative to some fixed point in time, although the date that represents zero is different across each package, and packages may vary in how they keep track of time of day. Ultimately, this makes calculating durations easy, and working with dates and times stored in this fashion becomes a matter of addition, subtraction, multiplication, and division.

1.2 Internal Representation

SAS stores dates as integers, while the datetime and time counters are stored as real numbers to account for fractional seconds. The origin of the algorithm used for SAS date processing comes from a January 14, 1980, Computerworld article by Dr. Bhairav Joshi of SUNY-Geneseo. The earliest date that SAS can handle with this algorithm is January 1, 1582 (essentially the implementation date of the current Gregorian calendar system). The latest date is far enough into the future that at least five digits will be required to display the year.

Dates as stored by SAS don't do us much good in the real world. The statement "I was born on -242" won't mean much to anyone else. However, "I was born on May 4, 1959," can easily be translated into something that most people can understand, or it can be used as is. Fortunately, SAS has a number of built-in facilities to perform automatic translation between the internal numbers stored in SAS and dates and times and their representation as understood by the rest of the world. These built-in tools include formats and informats (introduced in Section 1.3 and covered extensively in Chapters 2, 3, and 4), date and time constants (Section 1.4), and functions (Chapter 5).

1.3 External Representation (Basic FORMAT Concepts)

Formats perform an automatic translation between the actual value and the value to be displayed. Formats display the date, time, and datetime values in a fashion that is much more easily understood. Formats do not change the values themselves; they are just a way to display the values in any output.

When you have dates or times and want to translate them into SAS date and time values, you will use informats. Although you will need a statement, procedure, or a function to actually create the
SAS values, informats describe what the data look like so that SAS can translate it correctly for storage. We will discuss formats and informats in detail in Chapters 2, 3, and 4 because there are dozens of them. Three of the most commonly used formats that work with SAS date, time, and datetime values are used in the following section.

### 1.4 Date and Time as Numeric Constants in SAS

We've talked about internal and external representation of dates and times. How do you put a specific date into a program as a constant? Formats only change the way the values are displayed in output, so you can't use them. Informats need a function or a SAS statement to translate the characters they are given, so you could use them, but then you would always need to use the INPUT() function to create a SAS date in a DATA step or PROC SQL. The INPUT function takes a series of characters that you give it and translates it using the informat that describes what the series of characters look like. That's very inefficient if you just want one specific date.

```sas
date = INPUT("04AUG2013",DATE9.);
```

Look at the program in Example 1.1 to see how date, time, and datetime constants are written into a SAS program. Take note of the quotation marks around the values for date, time, and datetime and the letters that follow each closing quote.

#### Example 1.1: Date Constants

```sas
DATA date_constants;
  date = '04aug2013'd;  /* This is a date constant */
  time = '07:15:00't;   /* This is a time constant */
  datetime = '07aug1904:21:31:00'dt;  /* This is a datetime constant */
RUN;

TITLE "Unformatted Constants";
PROC PRINT DATA=date_constants;
  VAR date time datetime;
RUN;

TITLE "Formatted Constants";
PROC PRINT DATA=date_constants;
  VAR date time datetime;
  FORMAT date worddate32. time timeampm9. datetime datetime19.;  /* Format the constants */
RUN;
```

The quotes are used to create a literal value. You may use a pair of single or double quotes to specify the literal value. Dates have to be written as `ddmonyyyy`; times as `hh:mm:ss` (add a decimal point and more digits to represent fractional seconds if necessary); and datetimes as the date...
The first PROC PRINT statement displays the date, time, and datetime values we created with our constants without formats, so we can see the values as they are stored in the data set.

<table>
<thead>
<tr>
<th>Unformatted Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
</tr>
<tr>
<td>19574</td>
</tr>
</tbody>
</table>

The second PROC PRINT shows the effect of associating the variable DATE with the WORDDATE. format, the variable TIME with the TIMEAMPM. format, and the variable DATETIME with the DATETIME. format.

<table>
<thead>
<tr>
<th>Formatted Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
</tr>
<tr>
<td>August 4, 2013</td>
</tr>
</tbody>
</table>

Without the formats, you can see that the date constants we used to create the values stored in the data set are displayed as their actual SAS date, time, and datetime values. They don't make much sense to us until a format is associated with the variable.

What happens if you forget to put the "D," "T," or "DT" after your date constant? In Example 1.2, the "D," "T," and "DT" have been removed from the same date, time, and datetime in Example 1.1.

Example 1.2: Incorrect Date Constants

```sas
DATA bad_date_constants;
  date = '04aug2013';  /* This is NOT a date constant */
  time = '07:15:00';   /* This is NOT a time constant */
  datetime = '07aug1904:21:31:00';  /* This is NOT a datetime constant */
RUN;
TITLE "Unformatted Constants";
PROC PRINT DATA=bad_date_constants;
```
Chapter 1: Introduction to Dates and Times in SAS

```
VAR date time datetime;
RUN;
```

Now we print out the values without formats. While the problem may not be apparent at first glance, this result does not look like the unformatted SAS date, time, and datetime values in the previous example.

<table>
<thead>
<tr>
<th>Unformatted constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
</tr>
<tr>
<td>04aug2013</td>
</tr>
</tbody>
</table>

Now let's try to add one day to the date, and a minute (60 seconds) to both the time and datetime. Here is a partial log of what happens when we try this with the code in Example 1.2.

```
12 DATA bad_date_constants;
13 date = '04aug2013' + 1;
14 time = '07:15:00' + 60;
15 datetime = '07aug1904:21:31:00' + 60;
16 RUN;
```

NOTE: Character values have been converted to numeric values at the places given by: (Line):(Column).
13:8  14:8  15:12
NOTE: Invalid numeric data, '04aug2013' , at line 13 column 8.
NOTE: Invalid numeric data, '07:15:00' , at line 14 column 8.
NOTE: Invalid numeric data, '07aug1904:21:31:00' , at line 15 column 12.

0 date=. time=. datetime=. _ERROR_=1 _N_=1

The "invalid numeric data" note in the log tells you that you tried to use a character value to do something that requires a numeric value. The boldface last line tells you that you have missing values for all three variables, because you were trying to do math with a character value. Remember that SAS dates, times, and datetimes are always stored as numbers. When you see "invalid numeric data" where you intended to use a date constant, it is highly probable that your date constant is missing its identifying "D," "T," or "DT."

### 1.5 Length and Numeric Requirements for Date, Time, and Datetime

You can take advantage of the fact that dates are stored as integers to save space when you create variables to store them. Instead of using the default length of 8 for numeric variables, set the length of the numeric variables where you are storing dates to 4. This will safely store dates from January 1, 1582 (the earliest date SAS can handle), to October 23, 7701. A length of 5 is overkill, although that would extend the ending date another 534,773,760 days! A length of 3 will not accurately store
dates outside the range of January 1, 1960, and September 13, 1960. If you declare your date variables to be a length of 4, you will be able to store two dates in the space it would take to store one if you were using the SAS default length for numeric variables. This can save you a great deal of storage space in a large data warehouse.

Times may present a bit of a problem, because you may need to store fractional seconds. The rule is simple enough: If you want to store time values with fractional seconds, you must use a length of 8 to store them accurately. Otherwise, the length of 4 is long enough to store every possible time value from midnight to midnight down to the second. In these cases, not using the default length will allow you to store two times in the same amount of space as one.

Datetime values require more space, because a length of 4 will not store a datetime value with accuracy, regardless of whether you want fractional seconds. The number is just too big. As long as you are not storing fractional seconds, a length of 6 will store datetimes that accurately represent values from midnight on January 1, 1582, to 3:04:31 p.m. on April 9, 6315. Changing the range from the default of 8 to 6 for datetime values results in a 25 percent savings in space, which still may be significant depending on how much data you have. Of course, if you are going to maintain decimal places in your datetime values, you must use the default length of 8.

I have just provided the absolute minimum lengths required for accuracy. DO NOT attempt to save additional space by shrinking the variable lengths beyond 4, 6, or 8 as listed. You will lose precision, which could lead to unexpected results. Example 1.3 shows what can happen if you do not use enough bytes to store your date values. This example uses the value 19941, which represents the date of August 6, 2014, and it is in variables of lengths 3, 4, and 5.

**Example 1.3: The Effect of LENGTH Statements on Dates**

```sas
DATA date_length;
LENGTH len3 3 len4 4 len5 5;
len3 = 19941;
len4 = 19941;
len5 = 19941;
FORMAT len3 len4 len5 mmddyy10.;
RUN;
```

As the table below shows, when you try to store a date in fewer than 4 bytes, you do not get the correct value. Using a length of 4 to store your dates and times (without fractional seconds) is still a significant (50 percent) savings in the amount of storage required. You will create inaccuracies in your data if you try to save more than that. Saving additional space is not worth the risk of inaccurate data.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>len3</td>
<td>len4</td>
<td>len5</td>
</tr>
<tr>
<td>08/05/2014</td>
<td>08/06/2014</td>
<td>08/06/2014</td>
</tr>
</tbody>
</table>
1.6 General SAS Options for Dates

Two options influence the default date and time stamp that SAS places on pages of output and the SAS log. The DATE/NODATE option causes the start date and time of the SAS job (or session) to appear on each page of the SAS log and SAS output. These values are obtained from the operating system clock and are displayed as 24-hour clock time, followed by the day of the week, month, day, and four-digit year. If you are running SAS interactively, then the date and time are printed only on the output, not the log. By default, the DATE system option is in effect when you start SAS. However, if you do not want this default display, then use the NODATE option. You probably don't want SAS to display its default date stamp if you are going to put your own date and/or time stamp in the title or in a footnote (see Chapter 6).

As mentioned in the previous paragraph, if the DATE option is enabled, SAS prints the date and time that the current SAS session started on each page. If you want a more exact date and time on those pages, you can use the DTRESET system option, which will cause SAS to retrieve the date and time from the operating system clock each time a page is written. That date and time will then be placed on the page instead of the time that the SAS job started. Since the time is displayed in hours and minutes, you will only see it change every minute. The DTRESET option can be useful in interactive applications or SAS programs that may have been running for days or weeks, where knowing when the output was generated is more important than knowing when the SAS session began. Since the DTRESET option affects the default SAS date and time stamp, it works only if the DATE option is enabled. When you use the NODATE option, using DTRESET will have no effect because you aren't using the SAS date and time stamp on your output.
Index

Symbols and Numerics

& (ampersand) 192
= (equals sign) 207
$ (dollar sign) 61
%%% date directive 49

A

%a date directive 49
%A date directive 49
AFR language prefix 214
ampersand (&) 192
ANYDTDTEw: informat 82–83
   ANYDTDTMw: informat and 84–85
   ANYDTTMEw: informat and 85–86
   DATESTYLE= system option and 81, 82–83, 85–86
troubleshooting 239
ANYDTDTMw: informat 81, 84–85
automatic macro variables 186–189

B

%b date directive 49
%B date directive 49
B8601Clw.d informat 78, 108
B8601Da.w: format 93
B8601Da.w: informat 104
B8601Djw.d informat 79, 108
B8601Dnw: format 99
B8601Dt.w.d format 99–100
B8601Dtw.d informat 109
B8601Dx.w.d format 100–101
B8601Dzw.d format 102
B8601Dzw.d informat 110
B8601Lzw.d format 96
B8601Tmw.d format 94
B8601Tmw.d informat 105
B8601Ttx.w.d format 94–95
B8601Tzw.d format 97–98
B8601Tzw.d informat 105–106

C

calculations
   INTCK() function and 151–156
   INTNX() function and 156–159
   number of days between dates 145–149
   number of years between dates 146–147
   CALL IS8601_CONVERT 123–136
   CALL SYMGET() function 192–193
   CALL SYMPUT() function 192–193
   case sensitivity 48–49
   CAT language prefix 214
   CATS() function 14
   CATT() function 14
   CATX() function 14
   character constants 61
   character strings
      DATETIME informat and 63
      PUT() function and 55–56
   character variables
      INPUT() function and 61–62
      INPUTC() function 61–62
      PUT() function and 55–56
      PUTN() function and 55–56
   COMPRESS() function 14
   constants, date and time as 3–5
   CONVERT statement, EXPAND procedure
      converting to higher frequency 202–203
      METHOD= option 206
      OBSERVED= option 206–212
   CRO language prefix 214
   CSY language prefix 214

D

%d date directive 49, 51
DAN language prefix 214
DATA step
   FORMAT statement 10–137
   %LET statement and 191, 192
   DATADIF() function 145–146
   DATALINES statement 61
   DATATYPE= option 48, 51
Index

date directives, picture format 50
DATE() function 187–188
&DATE macro variable 187–188
DATE system option 7
DATEAMPMw.d format 42–43, 233
DATEPART() function 140
datetime formats and 41–42
troubleshooting 241, 242–243, 245
dates
automatic macro variables 186–189
CALL SYMPUT() function and 192–193
as constants 3–5
counters for 1
creating character strings 55–56
custom formats 47–55
datet ime values and 13–14, 41–42
default justification 13
Excel and 227–228
external representation of 2–3
formats for 14–37, 92–103
graphing 194–200
Hebrew formats 226
informats for 59, 61, 64–73, 81–86, 103–111
internal representation of 2
international formats and informats 212–220
interval definitions 149–151
interval functions 151–159
Japanese formats 226
Japanese informats 226–227
quick reference 231–233, 235–237
shifting intervals 159–168
%SYSFUNC() macro function and 187–189
Taiwanese formats 226
Taiwanese informats 226–227
in titles 186–187
troubleshooting 239–251
width specification 5–6, 13
YEARCUTOFF= system option and 57–59
DATESTYLE= system option
ANYDTDTEw. informat and 82
ANYDTDTMw. informat and 84
ANYDTTMEw. informat and 85–86
character strings and 63
DATETIMEw.d format 43–44, 79, 213, 233
DATEw. format 15
DTDATEw. format and 44
international format for 213
quick reference 231
DATEw. informat 64
ANYDTDTEw. informat 82
ANYDTDTMw. informat 84
ANYDTTMEw. informat 85–86
DATJUL() function 140–141
DAY() function 138
DAY interval 150
INTCK() function 152
INTNX() function 158
shift point 160
DAYw. format 15, 231
datetime values
automatic macro variables and 186–189
custom formats 47–55
date formats and 13–14, 41–42
default justification 13
Excel and 227–228
external representation of 2–3
formats for 41–46, 99–103
informats for 61, 78–81, 81–82, 108–111
internal representation of 2
international formats and informats 212–220
interval definitions 150–151
interval functions 151–159
quick reference 234, 235–237
shifting intervals 159–1682
%SYSFUNC() macro function and 187–189
width specification 5–6, 37, 41–42
YEARCUTOFF= system option and 57–59
DATETIMEw. informat
ANYDTDTEw. informat and 82
ANYDTDTMw. informat and 84
ANYDTTMEw. informat and 85–86
character strings and 63
DATETIMEw.d format 43–44, 79, 213, 233
DATEw. format 15
DTDATEw. format and 44
international format for 213
quick reference 231
DATEw. informat 64
ANYDTDTEw. informat 82
ANYDTDTMw. informat 84
ANYDTTMEw. informat 85–86
DATJUL() function 140–141
DAY() function 138
DAY interval 150
INTCK() function 152
INTNX() function 158
shift point 160
DAYw. format 15, 231
DB2 databases 227–228
DDMMYYB. format 17, 231
DDMMYYC. format 17, 231
DDMMYYD. format 17, 231
DDMMYYN. format 231
DDMMYYP. format 17, 231
DDMMYYYS. format 231
DDMMYYw. format 15–16, 30–31
  international format for 213
  quick reference 231
YYMMDDw. format and 30–31
DDMMYYw. informat 64–65
  ANYDSTDTEw. informat and 82
  ANYDSTDTMw. informat and 84
  ANYDTDTMew. informat and 85–86
DATASTYLE= system option 81–82
DDMMYYw. format 16–17
  MMYYw. format and 20
  YYMMyw. format and 30
  YYQyw. format and 33–34
DES language prefix 214
DEU language prefix 214
DFLANG= system option 212
DHMS() function 141–142, 247–248
dollar sign ($) 61
don t (.)
  See period
DOWNAMEw. format 17
  international format for 213
  quick reference 231
DTDATE9. format 44
DTDATEw. format 44
  DATEw format and 15
  quick reference 233
DTDAY interval 150
  INTNX() function 157
  INTNX() function 158
  shift point 160
DTHOUR interval 151, 161
DTMINUTE interval 151, 161
DTMONTH interval 151
  INTNX() function 158
  shift point 161
DTMONYYw. format 44
  MONYYw. format and 21
  quick reference 233
DTQTR interval 151
  INTNX() function 157, 158
  shift point 161
DTRESET system option 7
DTSECOND interval 151, 161
DTSEMIMONTH interval 151
  INTNX() function 158
  shift point 161
DTSEMIYEAR interval 151
  INTNX() function 157, 158
  shift point 161
DTTENDDAY interval 151
  INTNX() function 158
  shift point 161
DTWEEK interval 150
  INTNX() function 157
  shift point 161
DTWEEKDAY interval 151
  INTNX() function 157
  shift point 161
DTWKDATXw. format 45
  quick reference 233
DTYEAR interval 151
  INTNX() function 157, 158
  shift point 161
DTYEARw. format 45, 233
DTYYQCw. format 46, 233
durations, ISO 8601 116–136

E
E8601DAw. format 93
E8601DAw. informat 104
E8601DNw. format 99
E8601DTw.d format 100
E8601DTw.d informat 109–110
E8601DXw.d format 101–102
E8601DZw.d format 102–103
E8601DZw.d inform 110–111
E8601LZw.d format 96–97
E8601LZw.d informat 107
E8601TMw.d format 94
E8601TMw.d inform 105
E8601TXw.d format 95–96
E8601TZw.d format 98–99
E8601TZw.d informat 106–107
equals sign (=) 207
_ERROR_ automatic variable 6–63
ESP language prefix 214
"EUR" formats 213–214
"EUR" informat 213–214
EURDFDD. format 213
EURDFDEw. format 213
EURDFDIN. format 213
EURDFDTw. format 213
EURDFDWN. format 213
EURDFMN. format 213
EURDFMYw. format 213
EURDFWDX. format 213
EURDFWKX. format 213
Excel (Microsoft) 227–228
EXPAND procedure
capabilities 200–202
CONVERT statement 206–212
converting to higher frequency 202–203
FROM= option 206, 209–212
ID statement 204–205
interpolating missing values 205–206
TO= option 206, 209–212
external representation, date and time 2–3
datetime values 99–103
"EUR" 213–214
external representation of date, time and 2–3
graphing dates and 194–200
Hebrew 226
ISO 8601 92–103, 117–121
Japanese 226
"NLS" 214–220
PUT() function and 55–56
quick reference 231–233
Taiwanese 226
for time 37–41
using wrong 62–63
FRA language prefix 214
FROM= option, EXPAND statement 206, 209–212
FRS language prefix 214
FUNCTION statement 52–55
functions
  calculating intervals 145–149
creating date, time 140–145
current date, time 137–138
extraction 138–140
G
graphing dates 194–200
Gregorian year
  JULIANw. informat 65
  PDJULG4. informat 66
  PDJULGw. informat 22
  PDJULIw. format 22
  PDJULIw. informat 66
H
%H date directive 49, 223
HDATEw. format 226
HEBDATEx. format 226
Hebrew date formats 226
HHHMSSw. informat 73–75
HHMMw.d format 38, 233
HMS() function 143
HOLIDAY() function 148–149, 170
HOUR() function 140
HOUR interval 151
  INTNX() function 158
  shift point 161

F
%F date directive 223
FCMP procedure 52–55
FIN language prefix 214
FOOTNOTE statement 186
FORMAT procedure
  PICTURE statement 47–52
  troubleshooting 246
  VALUE statement 47–48
FORMAT statement
date directives and 50
functionality 10–13
INFORMAT statement and 59–60
troubleshooting 243–244
formats 10–13, 59
  custom 47–55
date constants and 4–5
  for dates 14–37
  for datetime 41–46

H
%H date directive 49, 223
HDATEw. format 226
HEBDATEx. format 226
Hebrew date formats 226
HHHMSSw. informat 73–75
HHMMw.d format 38, 233
HMS() function 143
HOLIDAY() function 148–149, 170
HOUR() function 140
HOUR interval 151
  INTNX() function 158
  shift point 161
HOURw.d format 38–39, 233
HUN language prefix 214

I

%I date directive 49, 223
ID statement, EXPAND procedure 204–205
imputed dates 52–55
INFORMAT statement 59–61, 59–63
informat
about 59
ANYDT variants 81–86
"EUR" 213–214
for dates 64–73
for datetime 78–81
for datetime values 108–111
for time 73–78
Hebrew 226
INFORMAT statement 59–63
ISO 8601 103–111
ISO 8601 duration and interval 121–123
Japanese and Taiwanese 226–227
"NLS" 220
using wrong 62–63
INPUT() function
functionality 3
informat and 59, 61–62
troubleshooting 240, 241
INPUT statement 60–61
INPUTC() function 61–62
INPUTN() function 61–62
INTCINDEX() function 183
INTCK() function 159, 156–162
calculating intervals 167–168
WORKINGDAYS interval 175–176
INTCYCLE() function 183
internal representation, date and time 2
interval multipliers
graphs and 194–200
shifting intervals and 162–163
intervals
basics of 149–151
creating 169–176
custom 162–163
INTCK() function 151–156
interval functions 176–181
INTNX() function 156–159
ISO 8601 116–136
measuring 167–168
number of days between dates 145–149
number of years between dates 146–147
retail calendar 181–183
shifting 159–168
INTFIT() function 177–178
INTFMT() function 178–179
INTGET() function 179–180
INTINDEX() function 183
INTNX() function 151, 156–160, 167–168
INTSEAS() function 183
INTSHIFT() function 180–181
INTTEST() function 181
ISO 8601 91–92
durations and intervals 116–136
forms 92–103
informat 103–111
ITA language prefix 214

J

%j date directive 49, 223
Japanese date formats 226
Japanese date informats 226–227
JDATEMYDw. informat 226–227
Jewish calendar 226
JNENGOw. informat 227
Joshi, Bhairav 2
JULDATE() function 138
JULDATE7() function 138
JULDAYw. function 17–18, 233
Julian date
JULDATE() function 138
JULDATE7() function 138
JULDAYw. format 17–18
JULIANw. format 18
JULIANw. informat 65
PDJULI1. format 23
PDJULG. informat 66
PDJULG4. informat 66
PDJULGw. format 22
PDJULIW. format 22–23
PDJULIW. informat 66
JULIANw. format 231
JULIANw. informat 65
   ANYDTDTEw. informat and 82
   ANYDTDTMw. informat and 84
   ANYDTTMEw. informat and 85–86
justification
date formats 13–14
PDJULIw. format and 22

K
Kanji representation 227

L
%LEFT() macro function 192
LENGTH statement 5–6
%LET statement 191, 192
LISTING destination 13, 19
literal values
   quotation marks and 3
   YEARCUTOFF= system option and 57–59
LOCALE= system option 111, 214–215

M
%m date directive 49, 223
%M date directive 49
MAC language prefix 214
macro functions, date and time in 186–189
macro variables
   CALL SYMPUT() function and 192–193
dates and 185–193
   quotation marks and 186
MAKEDATE() function 52–55
MDY() function 143, 247
MDYAMPMw. format 46, 233
MDYAMPMw.d format 79–80
METHOD= option, CONVERT statement (EXPAND) 206
Microsoft Excel 227–228
MINGUOw. format 226
MINGUOw. informat 227
MINUTE() function 140
MINUTE interval 151
   INTNX() function 158
   shift point 161
missing values
   DATESTYLE= system option 81
   EXPAND procedure and 205–206
   symbol for 50
   wrong informats and 62–63
MMDDYw. format 18
   MMDDYYwxw. format and 19
   quick reference 231
   YYMMDDw. format and 30–31
MMDDYY10. format 11–13, 18
MMDDYYB. format 19, 231
MMDDYYC. format 19, 231
MMDDYYD. format 231
MMDDYYN. format 231
MMDDYYP. format 19, 231
MMDDYYYS. format 231
MMDDYYWw. informat 65–66
   ANYDTDTEw. informat and 82
   ANYDTDTMw. informat and 84
   DATESTYLE= system option and 81–82
MMDDYYxw. format 19
   MMYYwxw. format and 20
   YYMMwxw. format and 30
   YYQwxw. format and 33–34
MMSSw.d format 39, 233
MMYYC. format 20, 231
MMYYD. format 20, 231
MMYYN. format 20, 232
MMYYP. format 20, 232
MMYYYS. format 232
MMYYYw. format 19–20, 231
MMYYxw. format 20
MONTH() function 138
MONTH interval 150
element 202–203
   INTCK() function 152
   INTNX() function 158
   shift point 160, 165–166
MONTHw. format 21, 232
MONNYw. format 21
   DTMONNYw. format and 44
   international format for 213
   quick reference 232
Index 259

MONYYw. informat 66
ANYDTDTEw. informat and 82
ANYDTDMw. informat and 84
ANYDTTMEmw. informat and 85–86
explanation of 66
MSEC8. informat 76

N

$N8601BABw.d format 119
$N8601Bw. informat 121–122
$N8601Bw.d format 118
$N8601EAw. format 120
$N8601EHw. format 120
$N8601Ew. format 119
$N8601EXw. format 121
$N8601Rw. informat 122–123
National Language Support (NLS) 111, 212
NENGOw. format 227
NLD language prefix 214
NLDATE. format 235
NLDATE() function 221–225
NLDATEL. format 215, 235
NLDATEM. format 235
NLDATEMD. format 215, 235
NLDATEMDL. format 215, 235
NLDATEMDM. format 216, 235
NLDATEMDS. format 216, 235
NLDATEMDT. format 217
NLDATEML. format 217
NLDATEMN. format 216, 235
NLDATES. format 216, 235
NLDATEmw. format 215
NLDATEmw. format 235
NLDATENw. format 216
NLDATENm. format 216, 235
NLDATEMYm. format 216, 235
NLDATEMYml. format 216, 235
NLDATEMYMM. format 216, 235
NLDATEMYMS. format 216, 235
NLDATERYQ. format 216, 235
NLDATERYQL. format 217, 236
NLDATERYQM. format 217, 236
NLDATERYQS. format 217, 236
NLDATERYR. format 217, 236
NLDATEMYw. format 217, 236
NLDATEMw() function 221–225
NLDATEMw. format 217
NLDATEmw. format 217
NLS (National Language Support) 111, 212
"NLS" formats 214–220
"NLS" informats 220
NLTIME. format 236
NLTIME() function 221–225
NLTIMEAP. format 236
NODATE system option 7
NOR language prefix 214
numeric variables
date, time as 3–5
functions from 140–145
INPUT() function and 61–62
LENGTH statement and 5–6
NWKDOM() function 144, 170

O

%o date directive 223
OBSERVED= option, CONVERT statement
(EXPAND) 206–212
OBSERVED=AVERAGE option, CONVERT statement (EXPAND) 207–212
OBSERVED-BEGINNING option, CONVERT statement (EXPAND) 207–212
OBSERVED=DERIVATIVE option, CONVERT statement (EXPAND) 207–212
OBSERVED-END option, CONVERT statement (EXPAND) 207–212
OBSERVED=MIDDLE option, CONVERT statement (EXPAND) 207–212
OBSERVED=TOTAL option, CONVERT statement (EXPAND) 207–212
ODS destinations 13–14, 214–215
OPTIONS INTERVALDS= statement 169–176
OPTIONS statement
DATE/NODATE system option 7
LOCALE= system option 212
OS TIME macro 76, 78
OUTLIB= option 52–55
%p date directive 49, 223
PDF destination 14
PDJULG. informat 66
PDJULG4. informat 66
PDJULGw. format 22
PDJULI1. format 23
PDJULIw. format 22–23
PDJULIw. informat 23, 66
PDTIME4. informat 76
period (.)
  format syntax and 16
  in informats 59
  missing values and 50
PICTURE statement, FORMAT procedure 47–52, 246
POL language prefix 214
PRINT procedure 11–12
PROC step 11
PTG language prefix 214
PUT() function 14, 17, 28, 55–56, 61–62, 221, 244–245
PUTN() function 14, 17, 28, 55–56

%QSYSFUNC() macro function 187
QTR() function 138
QTR interval 150
  INTNX() function 158
  shift point 160
QTRRw. format 23–24, 232
QTRw. format 23, 232

&R RAWDATE macro variable 187
retail calendar intervals 181–183
RMFDUR4. informat 76
RMFSTAMP8. informat 80
RTF destination 14
RUS language prefix 214

%S date directive 49, 223
sampling frequency
  converting to higher 202–203
  converting to lower 203–205
  OBSERVED= option and 207–212
SASDATEFMT= system option 228–229
SAS/ETS 183, 200
SAS/GRAPH 194–200
seasonality functions 183
SECOND() function 140
SECOND interval 151
  INTNX() function 158
  shift point 161
SEMIMONTH interval 150
  INTNX() function 158
  shift point 160
SEMIYEAR interval 150
  INTNX() function 158
  shift point 160
shift operators, intervals and 159–162
SLO language prefix 214
SMFSTAMP8. informat 80
SMRSTAMP8. informat 80
SQL procedure 193
STIMER system option 76
STIMERw. informat 76–77
STRIP() function 14
SVE language prefix 214
&SYSDATE automatic macro variable 185, 187
&SYSDATE9 automatic macro variable 186, 187
&SYSDAY automatic macro variable 186, 187
%SYSFUNC() macro function 187–189
&SYSTIME automatic macro variable 186, 187

Taiwanese date formats 226
Taiwanese date informats 226–227
TENDAY interval 150
  INTNX() function 158
  shift point 160
time
  automatic macro variables 186–189
  CALL SYMPUT() function and 192–193
  as constants 3–5
counters for 2
custom formats 47–55
default justification 13
Excel and 227–228
external representation of 2–3
formats for 37–41, 92–103
informs for 73–78, 81–82, 103–111
internal representation of 2
international formats and inormats 212–220
interval definitions 150–151
interval functions 151–159
quick reference 233, 235–237
shifting inervals 159–168
%SYSFUNC() macro function and 187–189
width specification 5–6, 37
TIME() function 138
time zone functions 111–115
TIMEAMPMM11. format 11–13
TIMEAMPMPw.d format 40
clock values and 37
quick reference 233
TIMEPART() function 41–42, 140
troubleshooting 242–243
TIMEw. informat 77–78
ANDTDTEw. informat and 82
ANYYDSTD前辈w. informat and 84
ANYYDRTTTMEw. informat and 85–86
TIMEw.d format 39–40
HHMMyw.d format and 38
quick reference 233
TIMEZONE= option 93, 111–112
TITLE statement 186
titles, date in 186–187
TO= option, EXPAND procedure 206, 209–212
TODAY() function 137
TODSTAMP8. informat 78
TODw.d format 40–41
clock values in 37
quick reference 233
troubleshooting dates 239–251
TU4. informat 78
two-digit year
  extraction functions and 138–140
  YEARCUTOFF= system options and 57–59, 79
TZONEDSTNAME() function 114
TZONEDSTTOFF() function 114
TZONEDSTNAME() function 112
TZONENAME() function 112–113
TZONEOFF() function 113, 246
TZONES2U() function 113
TZONESTTNAME() function 114–115
TZONESTTOFF() function 115
TZONEU2S() function 115, 246
U
U algorithm 25, 35, 69, 70, 139
%u date directive 223
%U date directive 49, 223
V
V algorithm 26–27, 36–37, 69–71, 139, 223
%V date directive 223
VALUE statement, FORMAT procedure 47–48
variables
  See character variables; macro variables;
  numeric variables
W
W algorithm 27, 37, 69–72, 139, 223
%w date directive 49, 223
%W date directive 223
WEEK() function 139
WEEK informat 69–70
WEEK interval 150
  INTCK() function 152
  INTNX() function 157, 159
    shift point 160, 161–162, 163–164
  WEEKDATEw. format 11–12, 24
    quick reference 232
  WEEKDATXw. format and 24–25
  WEEKDATXw. format 24–25
  DTWKDATXw. format and 445
    international format for 213–214
    quick reference 232
  WEEKDAY() function 139
  WEEKDAY interval 150
    INTNX() function 157
    shift point 160
  WEEKDAYw. format 25
    international format for 213
quick reference 232
WEEKUw. format 25–26
    quick reference 232
WEEKUw. informat 70–71
WEEKV interval 182–183
WEEKVw. format 26–27
    quick reference 232
    WEEKWw. format and 27
WEEKVw. informat 71–72
WEEKWw. format 27, 232
WHERE clause, SQL procedure 228
width specification
    for date formats 5–6, 13–14
    for datetime formats 5–6, 37, 41–42
    for time formats 5–6, 37
    formats and 13–14
    informats and 59
WORDDATEw. format 28
    FORMAT statement and 50
    quick reference 232
WORDDATXw. format 28–29, 249
    international format for 213
    quick reference 232
WORKINGDAYS interval 175–176

Y
%y date directive 49, 223
%Y date directive 49, 51, 223
Y2K problem 185
YEAR() function 139
YEAR interval 150
    INTCK() function 152
    INTNX() function 158
    shift point 160, 161–162, 165–166
YEAR10. interval 166–167
YEARCUTOFF= system option
    DATJUL() function and 140–141
    extraction functions and 138–140
    Japanese/Taiwanese date informats 226–227
    MDY() function 143
    PDJULGw. format and 22
    PDJULI. format 23
two-digit year and 57–59, 79
YYQ() function 144–145
YEARV interval 182
YEARw. format 29
    DTYEARw. format and 45
    quick reference 232
YMDDTTMw.d informat 80–81
YRDIF() function 146–147, 155–159
YYMM. format 232
YYMMMC. format 30, 232
YYMMMD. format 30, 232
YYMMDD. format 32, 232
YYMMDDD. format 32, 232
YYMMDDN. format 32, 232
YYMMDDP. format 32, 232
YYMMDDS. format 232
YYMMDD Dw. format 30–31
    DATESTYLE= system option 81–82
    quick reference 232
YYMMDDw. informat 66–67
    ANYDTDTEw. informat and 82
    ANYDTDTMw. informat and 84
    ANYDTTMEw. informat and 85–86
    DATESTYLE= system option and 81–82
YYMMDDxw. format 31–32
    YYMMxw. format and 30
    YYQxw. format and 33–34
YYMMMN. format 30, 232
YYMMNW. informat 67
YYMMMP. format 30, 232
YYMMMS. format 232
YYMMW. format 29
YYMMxw. format 30
YYMONW. format 32, 232
YYQ() function 144–145
YYQC. format 33
    DDTYQCw. format and 46
    quick reference 233
YYQD. format 34, 233
YYQN. format 33, 233
YYQP. format 34, 233
YYQR. format 35, 233
YYQRD. format 35, 233
YYQRN. format 35, 233
YYQRP. format 35, 233
YYQRS. format 233
YYQRw. format 34, 233
YYQRw. format 34–35
YYQS. format 233
YYQw. format 32–33, 233
YYQw. informat 67–68
   ANYDTDTEw. informat and 82
   ANYDTDTMw. informat and 84
   ANYDTTMEw. informat and 85–86
YYQxw. format 33–34
YYWEEKU. format 233
YYWEEKUw. format 35–36
YYWEEKV. format 233
YYWEEKVw. format 36
YYWEEKWw. format 37, 233
About This Book

Purpose
This book is designed to provide a detailed look at how the SAS date facility works, including an in-depth look at intervals and the interval functions, ISO 8601 date and datetime handling, and the NLS formats and informats. It is intended to serve as both a reference and a teaching tool. Ultimately, this book will allow the reader to become more confident in their daily work with dates, times, and datetimes in SAS.

Is This Book for You?
This book is aimed at beginning to intermediate SAS programmers, or those who work with ISO 8601 data, intervals, and/or reporting in multiple languages.

What’s New in This Edition
This new edition includes updated information to reflect the changes in version 9 of SAS; an expanded discussion of intervals, including the ability to define your own intervals; a section on how SAS works with the ISO 8601 date standards; and a troubleshooting appendix for beginners.

Scope of This Book
This book does not cover the SAS/ETS product, except for an overview of the EXPAND procedure.

About the Examples

Software Used to Develop the Book's Content
SAS version 9.4 (TS level 1M0) was used to produce all the examples in this book.

Example Code and Data
Many of the examples used in this book have accompanying code and data.
About This Book

You can access the example code and data for this book by linking to its author page at http://support.sas.com/publishing/authors. Select the name of the author. Then, look for the cover thumbnail of this book, and select Example Code and Data to display the SAS programs that are included in this book.

For an alphabetical listing of all books for which example code and data is available, see http://support.sas.com/bookcode. Select a title to display the book’s example code.

If you are unable to access the code through the website, send e-mail to saspress@sas.com.

Output and Graphics Used in This Book

Tables in this book were generated using ODS RTF, while graphics were generated as PNG files directly in SAS using the GPLOT and SGPLOT procedures. Screen captures were used to show the VIEWTABLE displays.

Additional Help

Although this book illustrates many analyses regularly performed in businesses across industries, questions specific to your aims and issues may arise. To fully support you, SAS Institute and SAS Press offer you the following help resources:

- For questions about topics covered in this book, contact the author through SAS Press:
  - Send questions by email to saspress@sas.com; include the book title in your correspondence.
  - Submit feedback on the author’s page at http://support.sas.com/author_feedback.

- For questions about topics in or beyond the scope of this book, post queries to the relevant SAS Support Communities at https://communities.sas.com/welcome.

- SAS Institute maintains a comprehensive website with up-to-date information. One page that is particularly useful to both the novice and the seasoned SAS user is its Knowledge Base. Search for relevant notes in the “Samples and SAS Notes” section of the Knowledge Base at http://support.sas.com/resources.

- Registered SAS users or their organizations can access SAS Customer Support at http://support.sas.com. Here you can pose specific questions to SAS Customer Support; under Support, click Submit a Problem. You will need to provide an email address to which replies can be sent, identify your organization, and provide a customer site number or license information. This information can be found in your SAS logs.
Meet the Author

Derek Morgan is a senior SAS programmer in the pharmaceutical industry who has been programming professionally in SAS for over 27 years. He spent 23 of those years at Washington University in St. Louis, where he received an A.B. in biology in 1985 and his first introduction to SAS as a student. During his career he has used SAS to create interactive data entry and management systems and to build and maintain research databases for analysis. In the late 1980s, he created a macro library to allow the use of nonproportional fonts in tables and listings on PostScript printers. He has taught introductory SAS programming and has presented many papers at local, regional, and national SAS Users Group conferences. Derek is married and has one son, and in his spare time he plays electric bass around the St. Louis area.

Learn more about this author by visiting his author page at http://support.sas.com/publishing/authors/morgan.html. There you can download free book excerpts, access example code and data, read the latest reviews, get updates, and more.

Keep in Touch

We look forward to hearing from you. We invite questions, comments, and concerns. If you want to contact us about a specific book, please include the book title in your correspondence.

Contact the Author through SAS Press

- By e-mail: saspress@sas.com
- Via the Web: http://support.sas.com/author_feedback

Purchase SAS Books

For a complete list of books available through SAS, visit sas.com/store/books.

- Phone: 1-800-727-0025
- E-mail: sasbook@sas.com

Subscribe to the SAS Training and Book Report

Receive up-to-date information about SAS training, certification, and publications via email by subscribing to the SAS Training & Book Report monthly eNewsletter. Read the archives and subscribe today at http://support.sas.com/community/newsletters/training!
Publish with SAS
SAS is recruiting authors! Are you interested in writing a book? Visit http://support.sas.com/saspress for more information.
Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore for additional books and resources.