



SAS[®] Certified Specialist Prep Guide

Base Programming Using SAS[®] 9.4



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2019. *SAS® Certified Specialist Prep Guide: Base Programming Using SAS® 9.4*. Cary, NC: SAS Institute Inc.

SAS® Certified Specialist Prep Guide: Base Programming Using SAS® 9.4

Copyright © 2019, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-64295-179-0 (Hard copy)

ISBN 978-1-64295-176-9 (Epub)

ISBN 978-1-64295-177-6 (Mobi)

ISBN 978-1-64295-178-3 (PDF)

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

February 2019

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

P1:certsppg

Contents

| | |
|---|------------|
| <i>How to Prepare for the Exam</i> | <i>vii</i> |
| <i>Accessibility Features of the Prep Guide</i> | <i>xi</i> |

PART 1 SAS Certified Specialist Prep Guide 1

| | |
|---|-----------|
| Chapter 1 • Setting Up Practice Data | 3 |
| Accessing Your Practice Data | 3 |
| Chapter 2 • Basic Concepts | 5 |
| Getting Started | 6 |
| The Basics of the SAS Language | 6 |
| SAS Libraries | 11 |
| Referencing SAS Files | 13 |
| SAS Data Sets | 17 |
| Chapter Quiz | 22 |
| Chapter 3 • Accessing Your Data | 25 |
| SAS Libraries | 25 |
| Viewing SAS Libraries | 28 |
| Chapter Quiz | 31 |
| Chapter 4 • Creating SAS Data Sets | 33 |
| Referencing an External Data File | 34 |
| The IMPORT Procedure | 35 |
| Reading and Verifying Data | 42 |
| Using the Imported Data in a DATA Step | 44 |
| Reading a Single SAS Data Set to Create Another | 45 |
| Reading Microsoft Excel Data with the XLSX Engine | 47 |
| Creating Excel Worksheets | 53 |
| Writing Observations Explicitly | 54 |
| Chapter Quiz | 55 |
| Chapter 5 • Identifying and Correcting SAS Language Errors | 59 |
| Error Messages | 59 |
| Correcting Common Errors | 61 |
| Chapter Quiz | 72 |
| Chapter 6 • Creating Reports | 75 |
| Creating a Basic Report | 76 |
| Selecting Variables | 77 |
| Identifying Observations | 78 |
| Sorting Data | 86 |
| Generating Column Totals | 88 |
| Specifying Titles and Footnotes in Procedure Output | 95 |
| Assigning Descriptive Labels | 100 |
| Using Permanently Assigned Labels | 103 |
| Chapter Quiz | 103 |

| | |
|---|------------|
| Chapter 7 • Understanding DATA Step Processing | 109 |
| How SAS Processes Programs | 109 |
| Compilation Phase | 112 |
| Execution Phase | 115 |
| Debugging a DATA Step | 120 |
| Testing Your Programs | 125 |
| Chapter Quiz | 126 |
| | |
| Chapter 8 • BY-Group Processing | 129 |
| Definitions | 129 |
| Preprocessing Data | 130 |
| FIRST. and LAST. DATA Step Variables | 131 |
| Chapter Quiz | 137 |
| | |
| Chapter 9 • Creating and Managing Variables | 141 |
| Creating Variables | 142 |
| Modifying Variables | 146 |
| Specifying Lengths for Variables | 149 |
| Subsetting Data | 151 |
| Transposing Variables into Observations | 158 |
| Using SAS Macro Variables | 164 |
| Chapter Quiz | 170 |
| | |
| Chapter 10 • Combining SAS Data Sets | 175 |
| How to Prepare Your Data Sets | 176 |
| Methods of Combining SAS Data Sets: The Basics | 177 |
| One-to-One Reading: Details | 178 |
| Concatenating: Details | 182 |
| Match-Merging: Details | 184 |
| Match-Merge Processing | 188 |
| Renaming Variables | 194 |
| Excluding Unmatched Observations | 196 |
| Chapter Quiz | 198 |
| | |
| Chapter 11 • Processing Data with DO Loops | 207 |
| The Basics of DO Loops | 207 |
| Constructing DO Loops | 212 |
| Nesting DO Loops | 215 |
| Iteratively Processing Observations from a Data Set | 217 |
| Conditionally Executing DO Loops | 218 |
| Chapter Quiz | 220 |
| | |
| Chapter 12 • SAS Formats and Informats | 225 |
| Applying SAS Formats and Informats | 225 |
| The FORMAT Procedure | 229 |
| Defining a Unique Format | 231 |
| Associating User-Defined Formats with Variables | 233 |
| Chapter Quiz | 238 |
| | |
| Chapter 13 • SAS Date, Time, and Datetime Values | 241 |
| SAS Date and Time Values | 241 |
| Reading Dates and Times with Informats | 243 |
| Example: Using Dates and Times in Calculations | 247 |
| Displaying Date and Time Values with Formats | 248 |
| Chapter Quiz | 251 |

| | |
|--|------------|
| Chapter 14 • Using Functions to Manipulate Data | 253 |
| The Basics of SAS Functions | 254 |
| SAS Functions Syntax | 255 |
| Converting Data with Functions | 256 |
| Manipulating SAS Date Values with Functions | 263 |
| Modifying Character Values with Functions | 277 |
| Modifying Numeric Values with Functions | 300 |
| Nesting SAS Functions | 303 |
| Chapter Quiz | 304 |
| | |
| Chapter 15 • Producing Descriptive Statistics | 307 |
| The MEANS Procedure | 307 |
| The FREQ Procedure | 317 |
| Chapter Quiz | 331 |
| | |
| Chapter 16 • Creating Output | 335 |
| The Output Delivery System (ODS) | 336 |
| Creating HTML Output with ODS | 338 |
| Creating PDF Output with ODS | 347 |
| Creating RTF Output with ODS | 352 |
| Creating EXCEL Output with ODS | 354 |
| The EXPORT Procedure | 356 |
| Chapter Quiz | 359 |

PART 2 Workbook 363

| | |
|--|------------|
| Chapter 17 • Practice Programming Scenarios | 365 |
| Scenario 1 | 366 |
| Scenario 2 | 366 |
| Scenario 3 | 367 |
| Scenario 4 | 368 |
| Scenario 5 | 368 |
| Scenario 6 | 369 |
| Scenario 7 | 370 |
| Scenario 8 | 371 |
| Scenario 9 | 372 |
| Scenario 10 | 373 |

PART 3 Quiz Answer Keys and Scenario Solutions 375

| | |
|---|------------|
| Appendix 1 • Chapter Quiz Answer Keys | 377 |
| Chapter 2: Basic Concepts | 377 |
| Chapter 3: Accessing Your Data | 378 |
| Chapter 4: Creating SAS Data Sets | 379 |
| Chapter 5: Identifying and Correcting SAS Language Errors | 380 |
| Chapter 6: Creating Reports | 381 |
| Chapter 7: Understanding DATA Step Processing | 383 |
| Chapter 8: BY-Group Processing | 384 |
| Chapter 9: Creating and Managing Variables | 384 |
| Chapter 10: Combining SAS Data Sets | 386 |
| Chapter 11: Processing Data with DO Loops | 387 |

| | |
|--|------------|
| Chapter 12: SAS Formats and Informat | 388 |
| Chapter 13: SAS Date, Time, and Datetime Values | 389 |
| Chapter 14: Using Functions to Manipulate Data | 390 |
| Chapter 15: Producing Descriptive Statistics | 390 |
| Chapter 16: Creating Output | 391 |
| Appendix 2 • Programming Scenario Solutions | 393 |
| Scenario 1 | 394 |
| Scenario 2 | 395 |
| Scenario 3 | 396 |
| Scenario 4 | 398 |
| Scenario 5 | 401 |
| Scenario 6 | 402 |
| Scenario 7 | 404 |
| Scenario 8 | 405 |
| Scenario 9 | 407 |
| Scenario 10 | 408 |
| Index | 411 |

How to Prepare for the Exam

Requirements and Details

Requirements

To complete examples in this book, you must have access to SAS windowing environment, SAS Enterprise Guide, or SAS Studio.

Exam Objectives and Updates to This Book

The current exam objectives and a list of any updates to this book are available at www.sas.com/certify. Exam objectives are subject to change.



Take a Practice Exam

Practice exams are available for purchase through SAS and Pearson VUE. For more information about practice exams, see www.sas.com/base_programmer_cert.

Registering for the Exam

To register for the SAS 9.4 Base Programming – Performance-Based Exam, see the SAS Global Certification website at www.sas.com/certify.

Additional Resources for Learning SAS Programming

| From SAS Software | |
|-------------------|--|
| Help | <ul style="list-style-type: none"> • SAS[®]9: Select Help ⇒ SAS Help and Documentation. • SAS Enterprise Guide: Select Help ⇒ SAS Enterprise Guide Help. • SAS Studio: Select the Help icon . |
| Documentation | <ul style="list-style-type: none"> • SAS[®]9: Select Help ⇒ SAS Help and Documentation. • SAS Enterprise Guide: Access online documentation on the web. • SAS Studio: Select the Help icon  and then click Help. |

| On the Web | |
|------------------------------|--|
| Base SAS Glossary | support.sas.com/baseglossary |
| Bookstore | www.sas.com/books |
| Certification | www.sas.com/certify |
| Communities | communities.sas.com |
| Knowledge Base | support.sas.com/notes |
| Learning Center | www.sas.com and click Learn . Then select Get Started with SAS . |
| SAS Documentation | support.sas.com/documentation documentation.sas.com |
| SAS Global Academic Program | www.sas.com and click Learn . Then select For Students and Educators . |
| SAS OnDemand | support.sas.com/ondemand/ |
| Syntax Quick Reference Guide | support.sas.com/content/dam/SAS/support/en/books/data/base-syntax-ref.pdf |
| Training | www.sas.com/training |
| Technical Support | support.sas.com . Then select Technical Support . |

Syntax Conventions

In this book, SAS syntax looks like this example:

```
DATA output-SAS-data-set
    (DROP=variables(s) | KEEP=variables(s));
SET SAS-data-set <options>;
BY variable(s);
RUN;
```

Here are the conventions that are used in the example:

- **DATA**, **DROP=**, **KEEP=**, **SET**, **BY**, and **RUN** are in uppercase bold because they must be spelled as shown.
- *output-SAS-data-set*, *variable(s)*, *SAS-data-set*, and *options* are in italics because each represents a value that you supply.
- <*options*> is enclosed in angle brackets because it is optional syntax.

- DROP= and KEEP= are separated by a vertical bar (|) to indicate that they are mutually exclusive.

The example syntax that is shown in this book includes only what you need to know in order to prepare for the certification exam. For complete syntax, see the appropriate SAS reference guide.

x *How to Prepare for the Exam*

Accessibility Features of the Prep Guide

Overview

The *SAS Certified Specialist Prep Guide: Base Programming Using SAS 9.4* is a test preparation document that uses the following environments and products:

- SAS windowing environment
- SAS Enterprise Guide
- SAS Studio or SAS University Edition

Accessibility Documentation Help

The following table contains accessibility information for the listed products:

Accessibility Documentation Links

| Product or Environment | Where to Find Accessibility Documentation |
|--|--|
| Base SAS (Microsoft Windows, UNIX, and z/OS) | support.sas.com/baseaccess |
| SAS Enterprise Guide | support.sas.com/documentation/onlinedoc/guide/index.html |
| SAS Studio | support.sas.com/studioaccess |

Documentation Format

Contact accessibility@sas.com if you need this document in an alternative digital format.

Chapter 2

Basic Concepts

| | |
|--|-----------|
| Getting Started | 6 |
| The Basics of the SAS Language | 6 |
| SAS Statements | 6 |
| Global Statements | 6 |
| DATA Step | 6 |
| PROC Step | 7 |
| SAS Program Structure | 7 |
| Processing SAS Programs | 8 |
| Log Messages | 9 |
| Results of Processing | 9 |
| SAS Libraries | 11 |
| Definition | 11 |
| Predefined SAS Libraries | 11 |
| Defining Libraries | 11 |
| How SAS Files Are Stored | 12 |
| Storing Files Temporarily or Permanently | 12 |
| Referencing SAS Files | 13 |
| Referencing Permanent SAS Data Sets | 13 |
| Referencing Temporary SAS Files | 13 |
| Rules for SAS Names | 14 |
| VALIDVARNAME=System Option | 14 |
| VALIDMEMNAME=System Option | 16 |
| When to Use VALIDMEMNAME=System Option | 17 |
| SAS Data Sets | 17 |
| Overview of Data Sets | 17 |
| Descriptor Portion | 17 |
| SAS Variable Attributes | 18 |
| Data Portion | 20 |
| SAS Indexes | 21 |
| Extended Attributes | 21 |
| Chapter Quiz | 22 |

Ready to take your SAS[®] and JMP[®] skills up a notch?



Be among the first to know about new books,
special events, and exclusive discounts.

support.sas.com/newbooks

Share your expertise. Write a book with SAS.

support.sas.com/publish

 sas.com/books
for additional books and resources.


THE POWER TO KNOW.®

Getting Started

In the SAS 9.4 Base Programming – Performance-Based exam, you are not tested on the details of running SAS software in the various environments. However, you might find such information useful when working with the practice data.

You can access a brief overview of the windows and menus in the SAS windowing environment, SAS Enterprise Guide, and SAS Studio at <http://video.sas.com/>. From **Categories** select **How To Tutorials** ⇒ **Programming**. Select the video for your SAS environment. Other tutorials are available from the SAS website.

The Basics of the SAS Language

SAS Statements

A *SAS statement* is a type of SAS language element that is used to perform a particular operation in a SAS program or to provide information to a SAS program. SAS statements are free-format. This means that they can begin and end anywhere on a line, that one statement can continue over several lines, and that several statements can be on the same line. Blank or special characters separate words in a SAS statement.

TIP You can specify SAS statements in uppercase or lowercase. In most situations, text that is enclosed in quotation marks is case sensitive.

Here are two important rules for writing SAS programs:

- A SAS statement ends with a semicolon.
- A statement usually begins with a SAS keyword.

There are two types of SAS statements:

- statements that are used in DATA and PROC steps
- statements that are global in scope and can be used anywhere in a SAS program

Global Statements

Global statements are used anywhere in a SAS program and stay in effect until changed or canceled, or until the SAS session ends. Here are some common global statements: TITLE, LIBNAME, OPTIONS, and FOOTNOTE.

DATA Step

The *DATA step* creates or modifies data. Input for a DATA can include raw data or a SAS data set. Output from a DATA step can include a SAS data set or a report. A *SAS data set* is a data file that is formatted in a way that SAS can understand.

For example, you can use DATA steps to do the following:

- put your data into a SAS data set
- compute values

- check for and correct errors in your data
- produce new SAS data sets by subsetting, supersetting, merging, and updating existing data sets

PROC Step

The *PROC step* analyzes data, produces output, or manages SAS files. The input for a PROC (procedure) step is usually a SAS data set. Output from a PROC step can include a report or an updated SAS data set.

For example, you can use PROC steps to do the following:

- create a report that lists the data
- analyze data
- create a summary report
- produce plots and charts

SAS Program Structure

A SAS program consists of a sequence of steps. A program can be any combination of DATA or PROC steps. A step is a sequence of SAS statements.

Here is an example of a simple SAS program.

Example Code 1 A Simple SAS Program

```

title1 'June Billing';           /* #1 */
data work.junefee;             /* #2 */
  set cert.admitjune;
  where age>39;
run;                             /* #3 */
proc print data=work.junefee; /* #4 */
run;

```

- 1 The TITLE statement is a global statement. Global statements are typically outside steps and do not require a RUN statement.
- 2 The DATA step creates a new SAS data set named Work.JuneFee. The SET statement reads in the data from Cert.AdmitJune. The new data set contains only those observations whose value for Age is greater than 39.
- 3 If a RUN or QUIT statement is not used at the end of a step, SAS assumes that the beginning of a new step implies the end of the previous step. If a RUN or QUIT statement is not used at the end of the last step in a program, SAS Studio and SAS Enterprise Guide automatically submit a RUN and QUIT statement after the submitted code.
- 4 The PROC PRINT step prints a listing of the new SAS data set. A PROC step begins with a PROC statement, which begins with the keyword PROC.

Output 2.1 PRINT Procedure Output**June Billing**

| Obs | ID | Name | Sex | Age | Date | Height | Weight | ActLevel | Fee |
|-----|------|--------------|-----|-----|----------|--------|--------|----------|--------|
| 1 | 2575 | Quigley, M | F | 40 | 06/06/10 | 69 | 163 | HIGH | 124.80 |
| 2 | 2589 | Wilcox, E | F | 41 | 06/17/10 | 67 | 141 | HIGH | 149.75 |
| 3 | 2523 | Johnson, R | F | 43 | 06/17/10 | 63 | 137 | MOD | 149.75 |
| 4 | 2584 | Takahashi, Y | F | 43 | 06/18/10 | 65 | 123 | MOD | 124.80 |
| 5 | 2571 | Nunnelly, A | F | 44 | 06/19/10 | 66 | 140 | HIGH | 149.75 |
| 6 | 2578 | Cameron, L | M | 47 | 06/20/10 | 72 | 173 | MOD | 124.80 |
| 7 | 2568 | Eberhardt, S | F | 49 | 06/21/10 | 64 | 172 | LOW | 124.80 |
| 8 | 2539 | LaMance, K | M | 51 | 06/22/10 | 71 | 158 | LOW | 124.80 |
| 9 | 2595 | Warren, C | M | 54 | 06/15/10 | 71 | 183 | MOD | 149.75 |
| 10 | 2579 | Underwood, K | M | 60 | 06/11/10 | 71 | 191 | LOW | 149.75 |

Processing SAS Programs

When a SAS program is submitted for execution, SAS first validates the syntax and then compiles the statements. DATA and PROC statements signal the beginning of a new step. The beginning of a new step also implies the end of the previous step. At a step boundary, SAS executes any statement that has not been previously executed and ends the step.

Example Code 2 Processing SAS Programs

```

data work.admit2;           /* #1 */
  set cert.admit;
  where age>39;
proc print data=work.admit2; /* #2 */
run;                       /* #3 */

```

- 1 The DATA step creates a new SAS data set named Work.Admit2 by reading Cert.Admit. The DATA statement is the beginning of the new step. The SET statement is used to read data. The WHERE statement conditionally reads only the observations where the value of the variable Age is greater than 39.
- 2 The PROC PRINT step prints the new SAS data set named Work.Admit2. The PROC PRINT statement serves as a step boundary in this example because a RUN statement was not used at the end of the DATA step. The PROC step also implies the end of the DATA step.
- 3 The RUN statement ends the PROC step.

TIP The RUN statement is not required between steps in a SAS program. However, it is a best practice to use a RUN statement because it can make the SAS program easier to read and the SAS log easier to understand when debugging.

Log Messages

The SAS log collects messages about the processing of SAS programs and about any errors that occur. Each time a step is executed, SAS generates a log of the processing activities and the results of the processing.

When SAS processes the sample program, it produces the log messages shown below. Notice that you get separate sets of messages for each step in the program.

Log 2.1 SAS Log Messages for Each Program Step

```

5  data work.admit2;
6      set cert.admit;
7      where age>39;
8  run;

NOTE: There were 10 observations read from the data set CERT.ADMIT.
      WHERE age>39;
NOTE: The data set WORK.ADMIT2 has 10 observations and 9 variables.
NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds

9  proc print data=work.admit2;
NOTE: Writing HTML Body file: sashtml.htm
10 run;

NOTE: There were 10 observations read from the data set WORK.ADMIT2.
NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.35 seconds
      cpu time           0.24 seconds

```

Results of Processing

The DATA Step

Suppose you submit the sample program below:

```

data work.admit2;
    set cert.admit;
    where age>39;
run;

```

When the program is processed, it creates a new SAS data set, Work.Admit2, containing only those observations with age values greater than 39. The DATA step creates a new data set and produces messages in the SAS log, but it does not create a report or other output.

The PROC Step

If you add a PROC PRINT step to this same example, the program produces the same new data set as before, but it also creates the following report:

```

data work.admit2;
    set cert.admit;
    where age>39;
run;

```

```
proc print data=work.admit2;
run;
```

Figure 2.1 PRINT Procedure Output

| The SAS System | | | | | | | | | |
|----------------|------|--------------|-----|-----|------|--------|--------|----------|--------|
| Obs | ID | Name | Sex | Age | Date | Height | Weight | ActLevel | Fee |
| 1 | 2523 | Johnson, R | F | 43 | 31 | 63 | 137 | MOD | 149.75 |
| 2 | 2539 | LaMance, K | M | 51 | 4 | 71 | 158 | LOW | 124.80 |
| 3 | 2568 | Eberhardt, S | F | 49 | 27 | 64 | 172 | LOW | 124.80 |
| 4 | 2571 | Nunnelly, A | F | 44 | 19 | 66 | 140 | HIGH | 149.75 |
| 5 | 2575 | Quigley, M | F | 40 | 8 | 69 | 163 | HIGH | 124.80 |
| 6 | 2578 | Cameron, L | M | 47 | 5 | 72 | 173 | MOD | 124.80 |
| 7 | 2579 | Underwood, K | M | 60 | 22 | 71 | 191 | LOW | 149.75 |
| 8 | 2584 | Takahashi, Y | F | 43 | 29 | 65 | 123 | MOD | 124.80 |
| 9 | 2589 | Wilcox, E | F | 41 | 16 | 67 | 141 | HIGH | 149.75 |
| 10 | 2595 | Warren, C | M | 54 | 7 | 71 | 183 | MOD | 149.75 |

Other Procedures

SAS programs often invoke procedures that create output in the form of a report, as is the case with the FREQ procedure:

```
proc freq data=sashelp.cars;
table origin*DriveTrain;
run;
```

Figure 2.2 FREQ Procedure Output

| The FREQ Procedure | | | | | |
|--|-------------------------------|------------|-------|--------|-------|
| Frequency Percent Row Pct Col Pct | Table of Origin by DriveTrain | | | | |
| | Origin | DriveTrain | | | Total |
| | | All | Front | Rear | |
| Asia | 34 | 99 | 25 | 158 | |
| | 7.94 | 23.13 | 5.84 | 36.92 | |
| | 21.52 | 62.66 | 15.82 | | |
| | 36.96 | 43.81 | 22.73 | | |
| Europe | 36 | 37 | 50 | 123 | |
| | 8.41 | 8.64 | 11.68 | 28.74 | |
| | 29.27 | 30.08 | 40.65 | | |
| | 39.13 | 16.37 | 45.45 | | |
| USA | 22 | 90 | 35 | 147 | |
| | 5.14 | 21.03 | 8.18 | 34.35 | |
| | 14.97 | 61.22 | 23.81 | | |
| | 23.91 | 39.82 | 31.82 | | |
| Total | 92 | 226 | 110 | 428 | |
| | 21.50 | 52.80 | 25.70 | 100.00 | |

Other SAS programs perform tasks such as sorting and managing data, which have no visible results except for messages in the log. (All SAS programs produce log messages, but some SAS programs produce only log messages.)

```
proc sort data=cert.admit;
by sex;
run;
```

Log 2.2 SAS Log: COPY Procedure Output

```
11  proc sort data=cert.admit;
12      by sex;
13  run;

NOTE: There were 21 observations read from the data set
      CERT.ADMIT.
NOTE: The data set CERT.ADMIT has 21 observations and 9
      variables.
NOTE: PROCEDURE SORT used (Total process time):
      real time          0.01 seconds
      cpu time           0.00 seconds
```

SAS Libraries

Definition

A *SAS library* contains one or more files that are defined, recognized, and accessible by SAS, and that are referenced and stored as a unit. One special type of file is called a *catalog*. In SAS libraries, catalogs function much like subfolders for grouping other members.

Predefined SAS Libraries

By default, SAS defines several libraries for you:

Sashelp

a permanent library that contains sample data and other files that control how SAS works at your site. This is a Read-Only library.

Sasuser

a permanent library that contains SAS files in the Profile catalog and that stores your personal settings. This is also a convenient place to store your own files.

Work

a temporary library for files that do not need to be saved from session to session.

You can also define additional libraries. When you define a library, you indicate the location of your SAS files to SAS. After you define a library, you can manage SAS files within it.

Note: If you are using SAS Studio, you might encounter the Webwork library. Webwork is the default output library in interactive mode. For more information about the Webwork library, see *SAS Studio: User's Guide*.

Defining Libraries

To define a library, you assign a library name to it and specify the location of the files, such as a directory path.

You can also specify an engine, which is a set of internal instructions that SAS uses for writing to and reading from files in a library.

You can define SAS libraries using programming statements. For information about how to write LIBNAME statements to define SAS libraries, see [Assigning Librefs on page 25](#).

TIP Depending on your operating environment and the SAS/ACCESS products that you license, you can create libraries with various engines. Each engine enables you to read a different file format, including file formats from other software vendors.

When you delete a SAS library, the pointer to the library is deleted, and SAS no longer has access to the library. However, the contents of the library still exist in your operating environment.

How SAS Files Are Stored

A SAS library is the highest level of organization for information within SAS.

For example, in the Windows and UNIX environments, a library is typically a group of SAS files in the same folder or directory.

The table below summarizes the implementation of SAS libraries in various operating environments.

Table 2.1 *Environments and SAS Libraries*

| Environment | Library Definition |
|---------------|---|
| Windows, UNIX | a group of SAS files that are stored in the same directory. Other files can be stored in the directory, but only the files that have SAS file extensions are recognized as part of the SAS library. |
| z/OS | a specially formatted host data set in which only SAS files are stored. |

Storing Files Temporarily or Permanently

Depending on the library name that you use when you create a file, you can store SAS files temporarily or permanently.

Table 2.2 *Temporary and Permanent SAS Libraries*

| | |
|--|---|
| Temporary SAS libraries last only for the current SAS session. | If you do not specify a library name when you create a file, the file is stored in the temporary SAS library, Work. If you specify the library name Work, then the file is stored in the temporary SAS library. When you end the session, the temporary library and all of its files are deleted. |
|--|---|

Permanent SAS libraries are available to you during subsequent SAS sessions.

To store files permanently in a SAS library, specify a library name other than the default library name Work.

In the example, when you specify the library name Cert when you create a file, you are specifying that the file is to be stored in a permanent SAS library.

Referencing SAS Files

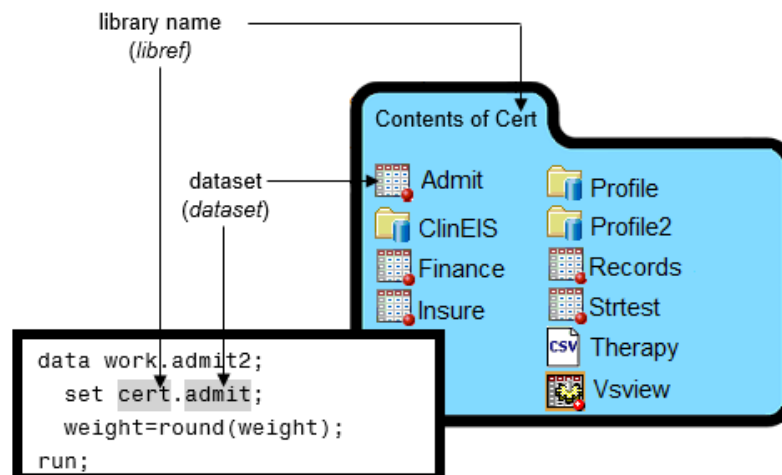
Referencing Permanent SAS Data Sets

To reference a permanent SAS data set in your SAS programs, use a two-level name consisting of the library name and the data set name:

libref.dataset

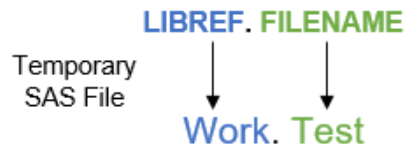
In the two-level name, *libref* is the name of the SAS library that contains the data set, and *data set* is the name of the SAS data set. A period separates the libref and data set name.

Figure 2.3 Two-Level Permanent SAS Name



Referencing Temporary SAS Files

To reference temporary SAS files, you can specify the default libref Work, a period, and the data set name. For example, the two-level name, Work.Test, references the SAS data set named Test that is stored in the temporary SAS library Work.

Figure 2.4 Two-Level Temporary SAS Library Name

Alternatively, you can use a one-level name (the data set name only) to reference a file in a temporary SAS library. When you specify a one-level name, the default libref Work is assumed. For example, the one-level name Test references the SAS data set named Test that is stored in the temporary SAS library Work.

Figure 2.5 One-Level Temporary SAS Library Name

Rules for SAS Names

By default, the following rules apply to the names of SAS data sets, variables, and libraries:

- They must begin with a letter (A-Z, either uppercase or lowercase) or an underscore (_).
- They can continue with any combination of numbers, letters, or underscores.
- They can be 1 to 32 characters long.
- SAS library names (librefs) can be 1 to 8 characters long.

These are examples of valid data set names and variable names:

- Payroll
- LABDATA2015_2018
- _EstimatedTaxPayments3

VALIDVARNAME=System Option

SAS has various rules for variable names. You set these rules using the VALIDVARNAME= system option. VALIDVARNAME specifies the rules for valid SAS variable names that can be created and processed during a SAS session.

Syntax, VALIDVARNAME=

VALIDVARNAME= *V7|UPCASE|ANY*

V7 specifies that variable names must follow these rules:

- SAS variable names can be up to 32 characters long.
- The first character must begin with a letter of the Latin alphabet (A - Z, either uppercase or lowercase) or an underscore (_). Subsequent characters can be letters of the Latin alphabet, numerals, or underscores.
- Trailing blanks are ignored. The variable name alignment is left-justified.
- A variable name cannot contain blanks or special characters except for an underscore.
- A variable name can contain mixed-case letters. SAS stores and writes the variable name in the same case that is used in the first reference to the variable. However, when SAS processes a variable name, SAS internally converts it to uppercase. Therefore, you cannot use the same variable name with a different combination of uppercase and lowercase letters to represent different variables. For example, **cat**, **CaT**, and **CAT** all represent the same variable.
- Do not assign variables the names of special SAS automatic variables (such as **_N_** and **_ERROR_**) or variable list names (such as **_NUMERIC_**, **_CHARACTER_**, and **_ALL_**) to variables.

UPCASE specifies that the variable name follows the same rules as *V7*, except that the variable name is uppercase, as in earlier versions of SAS.

ANY specifies that SAS variable names must follow these rules:

- The name can begin with or contain any characters, including blanks, national characters, special characters, and multi-byte characters.
 - The name can be up to 32 bytes long.
 - The name cannot contain any null bytes.
 - Leading blanks are preserved, but trailing blanks are ignored.
 - The name must contain at least one character. A name with all blanks is not permitted.
 - A variable name can contain mixed-case letters. SAS stores and writes the variable name in the same case that is used in the first reference to the variable. However, when SAS processes a variable name, SAS internally converts it to uppercase. Therefore, you cannot use the same variable name with a different combination of uppercase and lowercase letters to represent different variables. For example, **cat**, **CaT**, and **CAT** all represent the same variable.
-

Note: If you use characters other than the ones that are valid when **VALIDVARNAME=V7**, then you must express the variable name as a name literal and set **VALIDVARNAME=ANY**. If the name includes either a percent sign (%) or an ampersand (&), then use single quotation marks in the name literal to avoid interaction with the SAS macro facility.

CAUTION:

Throughout SAS, using the name literal syntax with SAS member names that exceed the 32-byte limit or have excessive embedded quotation marks might cause unexpected results. The **VALIDVARNAME=ANY** system option enables compatibility with other DBMS variable (column) naming conventions, such as allowing embedded blanks and national characters.

VALIDMEMNAME=System Option

You can use the VALIDMEMNAME= system option to specify rules for naming SAS data sets.

Syntax, VALIDMEMNAME=

VALIDMEMNAME= COMPATIBLE | EXTEND

Important: COMPATIBLE is the default system option for VALIDMEMNAME=.

COMPATIBLE specifies that a SAS data set name must follow these rules:

- The length of the names can be up to 32 characters long.
- Names must begin with a letter of the Latin alphabet (A- Z, a - z) or an underscore. Subsequent characters can be letters of the Latin alphabet, numerals, or underscores.
- Names cannot contain blanks or special characters except for an underscore
- Names can contain mixed-case letters. SAS internally converts the member name to uppercase. Therefore, you cannot use the same member name with a different combination of uppercase and lowercase letters to represent different variables. For example, **customer**, **Customer**, and **CUSTOMER** all represent the same member name. How the name is saved on disk is determined by the operating environment.

EXTEND specifies that the data set name must follow these rules:

- Names can include national characters.
 - The name can include special characters, except for the / \ * ? " < > | : - characters.
 - The name must contain at least one character.
 - The length of the name can be up to 32 bytes.
 - Null bytes are not allowed.
 - Names cannot begin with a blank or a ‘.’ (period).
 - Leading and trailing blanks are deleted when the member is created.
 - Names can contain mixed-case letters. SAS internally converts the member name to uppercase. Therefore, you cannot use the same member name with a different combination of uppercase and lowercase letters to represent different variables. For example, **customer**, **Customer**, and **CUSTOMER** all represent the same member name. How the name appears is determined by the operating environment.
-

Note: If VALIDMEMNAME=EXTEND, SAS data set names must be written as a SAS name literal. If you use either a percent sign (%) or an ampersand (&), then you must use single quotation marks in the name literal in order to avoid interaction with the SAS macro facility.

CAUTION:

Throughout SAS, using the name literal syntax with SAS member names that exceed the 32-byte limit or that have excessive embedded quotation marks might cause unexpected results. The intent of the VALIDMEMNAME=EXTEND system option is to enable compatibility with other DBMS member naming conventions, such as allowing embedded blanks and national characters.

When to Use VALIDMEMNAME=System Option

Use VALIDMEMNAME= EXTEND system option when the characters in a SAS data set name contain one of the following:

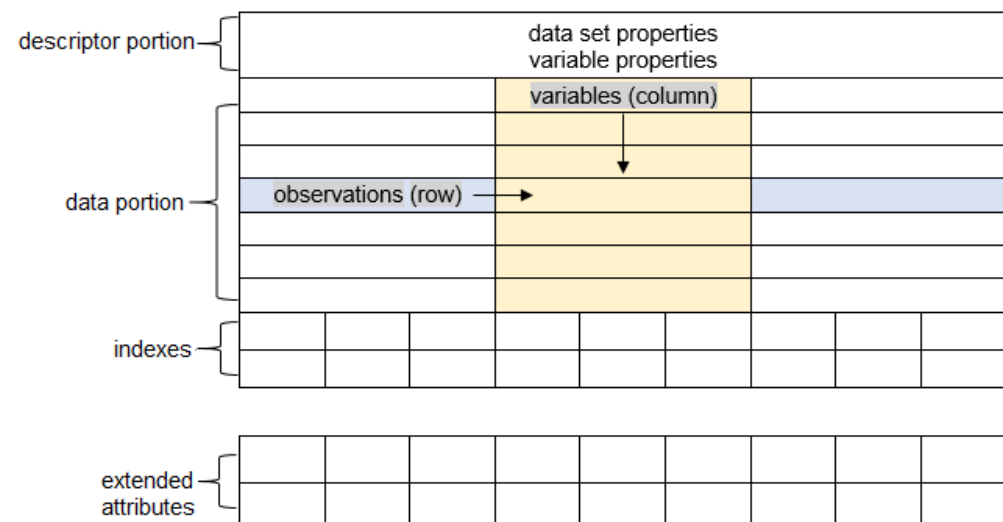
- international characters
- characters supported by third-party databases
- characters that are commonly used in a filename

SAS Data Sets

Overview of Data Sets

A *SAS data set* is a file that consists of two parts: a descriptor portion and a data portion. Sometimes a SAS data set also points to one or more indexes, which enable SAS to locate rows in the data set more efficiently. (The data sets that are shown in this chapter do not contain indexes.) Extended attributes are user-defined attributes that further define a SAS data set.

Figure 2.6 Parts of a SAS Data Set



Descriptor Portion

The descriptor portion of a SAS data set contains information about the data set, including the following:

- the name of the data set
- the date and time that the data set was created
- the number of observations
- the number of variables

The table below lists part of the descriptor portion of the data set Cert.Insure, which contains insurance information for patients who are admitted to a wellness clinic.

Table 2.3 Descriptor Portion of Attributes in a SAS Data Set

| | |
|---------------------|---------------------|
| Data Set Name: | CERT.INSURE |
| Member Type: | DATA |
| Engine: | V9 |
| Created: | 07/03/2018 10:53:05 |
| Observations: | 21 |
| Variables: | 7 |
| Indexes: | 0 |
| Observation Length: | 64 |

SAS Variable Attributes

The descriptor portion of a SAS data set contains information about the properties of each variable in the data set. The properties information includes the variable's name, type, length, format, informat, and label.

When you write SAS programs, it is important to understand the attributes of the variables that you use. For example, you might need to combine SAS data sets that contain same-named variables. In this case, the variables must be the same type (character or numeric). If the same-named variables are both character variables, you still need to check that the variable lengths are the same. Otherwise, some values might be truncated.

The following table uses Cert.Insure data and the VALIDVARNAME=ANY system option. The SAS variable has several attributes that are listed here:

Table 2.4 Variable Attributes

| Variable Attribute | Definition | Example | Possible Values |
|--------------------|--|-------------------------|---------------------|
| Name | identifies a variable. A variable name must conform to SAS naming rules. See “Rules for SAS Names” for SAS names rules. | Policy Total Name | Any valid SAS name. |

| Variable Attribute | Definition | Example | Possible Values |
|--------------------|---|--|---|
| Type | identifies a variable as numeric or character. Character variables can contain any values. Numeric variables can contain only numeric values (the numerals 0 through 9, +, -, ., and E for scientific notation). | Char Num Char | Numeric and character |
| Length | refers to the number of bytes used to store each of the variable's values in a SAS data set. Character variables can be up to 32,767 bytes long. All numeric variables have a default length of 8 bytes. Numeric values are stored as floating-point numbers in 8 bytes of storage. | 5 8 14 | 2 to 8 bytes 1 to 32,767 bytes for character |
| Format | affects how data values are written. Formats do not change the stored value in any way; they merely control how that value is displayed. SAS offers a variety of character, numeric, and date and time formats. | \$98.64 | Any SAS format If no format is specified, the default format is BEST12. for a numeric variable, and \$w. for a character variable. |
| Informat | reads data values in certain forms into standard SAS values. Informats determine how data values are read into a SAS data set. You must use informats to read numeric values that contain letters or other special characters. | 99 | Any SAS informat The default informat for numeric is w.d and for character is \$w. |
| Label | refers to a descriptive label up to 256 characters long. A variable label, which can be printed by some SAS procedures, is useful in report writing. | Policy Number Total Balance Patient Name | Up to 256 characters |

The following output is the descriptor portion of Cert.Insure.

Output 2.2 Descriptor Portion of Cert.Insure

| Alphabetic List of Variables and Attributes | | | | | | |
|---|------------|------|-----|-----------|----------|---------------|
| # | Variable | Type | Len | Format | Informat | Label |
| 7 | BalanceDue | Num | 8 | 6.2 | | |
| 4 | Company | Char | 11 | | | |
| 1 | ID | Char | 4 | | | |
| 2 | Name | Char | 14 | | | Patient Name |
| 5 | PctInsured | Num | 8 | | | |
| 3 | Policy | Char | 5 | | | Policy Number |
| 6 | Total | Num | 8 | DOLLAR8.2 | COMMA10. | Total Balance |

Data Portion

Data Portion Overview

The data portion of a SAS data set is a collection of data values that are arranged in a rectangular table. In the example below, the company **MUTUALITY** is a data value, Policy **32668** is a data value, and so on.

Figure 2.7 Parts of a SAS Data Set: Data Portion

| ID | Patient Name | Policy Number | Company | PctInsured | Total Balance | BalanceDue |
|------|----------------|---------------|-----------|------------|---------------|------------|
| 2458 | Murray, W | 32668 | MUTUALITY | 100 | 98.64 | 0.00 |
| 2462 | Almers, C | 95824 | RELIABLE | 80 | 780.23 | 156.05 |
| 2501 | Bonaventure, T | 87795 | A&R | 80 | 47.38 | 9.48 |
| 2523 | Johnson, R | 39022 | ACME | 50 | 122.07 | 61.04 |

Observations (Rows)

Observations (also called rows) in the data set are collections of data values that usually relate to a single object. The values **2458**, **Murray W**, **32668**, **MUTUALITY**, **100**, **98.64**, and **0.00** are comprised in a single observation in the data set shown below.

Figure 2.8 Parts of a SAS Data Set: Observations

| | ID | Patient Name | Policy Number | Company | PctInsured | Total Balance | BalanceDue |
|-------------|------|----------------|---------------|-----------|------------|---------------|------------|
| Observation | 2458 | Murray, W | 32668 | MUTUALITY | 100 | 98.64 | 0.00 |
| | 2462 | Almers, C | 95824 | RELIABLE | 80 | 780.23 | 156.05 |
| | 2501 | Bonaventure, T | 87795 | A&R | 80 | 47.38 | 9.48 |
| | 2523 | Johnson, R | 39022 | ACME | 50 | 122.07 | 61.04 |

This data set has 21 observations, each containing information about an individual. To view the full descriptor portion of this data set, see [Table 2.3 on page 18](#). A SAS data set can store any number of observations.

Variables (Columns)

Variables (also called columns) in the data set are collections of values that describe a particular characteristic. The values 2458, 2462, 2501, and 2523 are comprised in the variable ID in the data set shown below.

Figure 2.9 Parts of a SAS Data Set: Variables

Variables

| ID | Patient Name | Policy Number | Company | PctInsured | Total Balance | BalanceDue |
|------|----------------|---------------|-----------|------------|---------------|------------|
| 2458 | Murray, W | 32668 | MUTUALITY | 100 | 98.64 | 0.00 |
| 2462 | Almers, C | 95824 | RELIABLE | 80 | 780.23 | 156.05 |
| 2501 | Bonaventure, T | 87795 | A&R | 80 | 47.38 | 9.48 |
| 2523 | Johnson, R | 39022 | ACME | 50 | 122.07 | 61.04 |

This data set contains seven variables: ID, Name, Policy, Company, PctInsured, Total, and BalanceDue. A SAS data set can store thousands of variables.

Missing Values

Every variable and observation in a SAS data set must have a value. If a data value is unknown for a particular observation, a missing value is recorded in the SAS data set. A period (.) is the default value for a missing numeric value, and a blank space is the default value for a missing character value.

Figure 2.10 Parts of a SAS Data Set: Missing Data Values

| ID | Patient Name | Policy Number | Company | PctInsured | Total Balance | BalanceDue |
|------|----------------|---------------|-----------|------------|---------------|------------|
| 2458 | Murray, W | 32668 | MUTUALITY | 100 | 98.64 | 0.00 |
| 2462 | Almers, C | 95824 | RELIABLE | 80 | 780.23 | 156.05 |
| 2501 | Bonaventure, T | 87795 | A&R | . | 47.38 | 9.48 |
| 2523 | Johnson, R | 39022 | ACME | 50 | 122.07 | 61.04 |

Missing Value

SAS Indexes

An index is a separate file that you can create for a SAS data file in order to provide direct access to a specific observation. The index file has the same name as its data file and a member type of INDEX. Indexes can provide faster access to specific observations, particularly when you have a large data set. The purpose of SAS indexes is to optimize WHERE expressions and to facilitate BY-group processing. For more information, see “[Specifying WHERE Expressions](#)” and see [Chapter 8, “BY-Group Processing](#)”.

Extended Attributes

Extended attributes are user-defined metadata that is defined for a data set or for a variable (column). Extended attributes are represented as name-value pairs.

TIP You can use PROC CONTENTS to display data set and variable extended attributes.

Chapter Quiz

Select the best answer for each question. Check your answers using the answer key in the appendix.

1. How many observations and variables does the data set below contain?

| Name | Sex | Age |
|----------|-----|-----|
| Picker | M | 32 |
| Fletcher | | 28 |
| Romano | F | . |
| Choi | M | 42 |

- 3 observations, 4 variables
 - 3 observations, 3 variables
 - 4 observations, 3 variables
 - cannot tell because some values are missing
2. How many program steps are executed when the program below is processed?

```
data user.tables;
  set work.jobs;run;
proc sort data=user.tables;
  by name; run;
proc print data=user.tables;
run;
```

- three
 - four
 - five
 - six
3. What type of variable is the variable AcctNum in the data set below?

| AcctNum | Gender |
|---------|--------|
| 3456_1 | M |
| 2451_2 | |
| Romano | F |
| Choi | M |

- numeric
- character
- can be either character or numeric
- cannot tell from the data shown

4. What type of variable is the variable `Wear` based on the justification of the text in the data set below?

| Brand | Wear |
|-------|------|
| Acme | 43 |
| Ajax | 34 |
| Atlas | . |

- numeric
 - character
 - can be either character or numeric
 - cannot tell from the data shown
5. With the system option `VALIDVARNAME=ANY`, which of the following variable names is valid?
- 4BirthDate
 - \$Cost
 - Tax-Rate
 - all of the above
6. Which of the following files is a permanent SAS file?
- Work.PrdSale
 - Cert.MySales
 - Certxl.Quarter1
 - b and c only
 - a, b, and c
7. In a `DATA` step, how can you reference a temporary SAS data set named `Forecast`?
- Forecast
 - Work.Forecast
 - Sales.Forecast (after assigning the libref `Sales`)
 - a and b only
8. What is the default length for the numeric variable `Balance`?

| Name | Balance |
|----------|---------|
| Adams | 105.73 |
| Geller | 107.89 |
| Martinez | 97.45 |
| Noble | 182.50 |

- 5
- 6
- 7
- 8

9. How many statements does the following SAS program contain?

```
proc print data=cert.admit label double;  
  var ID Name Sex Age; where Sex=F;  
  label Sex='Gender'; run;
```

- a. three
 - b. four
 - c. five
 - d. six
10. What is a SAS library?
- a. a collection of SAS files, such as SAS data sets and catalogs
 - b. in some operating environments, a physical collection of SAS files
 - c. a group of SAS files in the same folder or directory
 - d. all of the above