

SAS[®] Technical Report P-258 Using the REPORT Procedure in a Nonwindowing Environment

Release 6.07



SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., SAS® Technical Report P-258, *Using the REPORT Procedure in a Nonwindowing Environment, Release 6.07*, Cary, NC: SAS Institute Inc., 1993. XXX pp.

SAS® Technical Report P-258, Using the REPORT Procedure in a Nonwindowing Environment, Release 6.07

Copyright © 1993 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-55544-572-1

All rights reserved. Printed in the United States of America. 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.

1st printing, October 1993

The SAS® System is an integrated system of software providing complete control over data access, management, analysis, and presentation. Base SAS software is the foundation of the SAS System. Products within the SAS System include SAS/ACCESS®, SAS/AF®, SAS/ASSIST®, SAS/CALC®, SAS/CONNECT®, SAS/CPE®, SAS/DMI®, SAS/EIS®, SAS/ENGLISH®, SAS/ETS®, SAS/FSP®, SAS/GRAPH®, SAS/IML®, SAS/IMS-DL/I®, SAS/INSIGHT®, SAS/LAB®, SAS/OR®, SAS/PH-Clinical®, SAS/QC®, SAS/REPLAY-CICS®, SAS/SHARE®, SAS/STAT®, SAS/TOOLKIT®, SAS/TUTOR®, SAS/DB2™, SAS/GIS™, SAS/IMAGE™, SAS/NVISION™, SAS/SESSION™, and SAS/SQL-DS™ software. Other SAS Institute products are SYSTEM 2000® Data Management Software, with basic SYSTEM 2000, CREATE™, Multi-User™, QueX™, Screen Writer™, and CICS interface software; NeoVisuals® software; JMP®, JMP IN®, JMP Serve®, and JMP *Design*® software; SAS/RTERM® software; and the SAS/C® Compiler and the SAS/CX® Compiler; and Emulus™ software. MultiVendor Architecture™ and MVA™ are trademarks of SAS Institute Inc. SAS Video Productions™ and the SVP logo are service marks of SAS Institute Inc. SAS Institute also offers SAS Consulting®, Ambassador Select™, and On-Site Ambassador™ services. *Authorline*®, *Observations*®, *SAS Communications*®, *SAS Training*®, *SAS Views*®, the SASware Ballot®, and *JMPer Cable*™ are published by SAS Institute Inc. All trademarks above are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

The Institute is a private company devoted to the support and further development of its software and related services.

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

Doc P4, 081993

Contents

Reference Aids vii

Acknowledgments ix

Using This Book xi

Background 1

Introduction 1

Invoking PROC REPORT in the Nonwindowing Environment 1

Specifying the Input Data Set 1

Creating the SAS Data Set BUDGET2 2

Chapter 1 · Shaping the Report Layout 5

Introduction 5

Listing All Observations and All Variables 8

Specifying and Ordering Columns to Display 11

Ordering Rows 13

Consolidating Multiple Observations into One Row 16

Using Variable Values as Column Headers 19

Chapter 2 · Reports with Statistics 23

Introduction 23

Placing Statistics above or below a Variable 25

Reporting Multiple Statistics for One Variable 27

Showing Frequency Counts 29

Chapter 3 · Basic Report Enhancements 33

Introduction 33

Adding Titles 34

Adding Footnotes 36

Changing the Linesize 38

Changing the Pagesize 40

Selecting Observations to Process 42

Controlling Space between Columns 44

Specifying Formats 46

Specifying Column Widths 48

Suppressing the Display of an Item 50

Wrapping the Value of a Character Variable across Multiple Rows 52

Presenting One Variable in Two Different Ways 54

Storing a Report Definition 56

Using a Stored Report Definition 59

Chapter 4 · Customizing Column Headers 61

Introduction 61

Customizing the Text in Column Headers 62

Controlling Line Breaks in Column Headers 64

Extending a Column Header to Fill a Column 66

Adding a Blank Line beneath Column Headers 70

Underlining Column Headers 71

Spanning Multiple Columns with One Customized Column Header 73

Chapter 5 · Adding Variables that Are Not in the Input Data Set 75

Introduction 75

Adding a Variable Computed from a Group, Order, Computed, or Display Variable 78

Adding a Variable Computed from a Statistic 80

Adding a Variable Computed from a Variable that Shares a Column with an Across Variable 82

Adding a Character Variable 85

Chapter 6 · Working with Groups of Observations 87

Introduction 87

Producing a Separate Page for Each Group 93

Inserting a Blank Line between Groups 97

Inserting Multiple Blank Lines between Groups 99

Summarizing Variables and Statistics for Groups 101

Overlining Summary Information 103

Underlining Summary Information 105

Suppressing the Value of the Group Variable in the Summary Information 107

Changing the Value of a Character Variable in the Summary Information 109

Specifying Text and Variable Values between Groups 111

Controlling the Amount of Space Used to Write a Character Variable 114

Controlling the Amount of Space Used to Write a Numeric Variable 117

Controlling the Horizontal Position of Text between Groups 120

Initializing Variables for Each Group 122

Placing Conditional Text between Groups 125

Controlling Whether Statements in a Compute Block Execute between Groups 128

Chapter 7 · Working with the Report as a Whole 131

Introduction 131

Summarizing Variables and Statistics for the Whole Report 136

Overlining Summary Information 138

Underlining Summary Information 140

Inserting a Blank Line after Summary Information at the Beginning of a Report 142

Inserting Multiple Blank Lines after Summary Information at the Beginning of a Report 144

Changing the Value of a Character Variable in the Summary Information 146

Specifying Text and Variable Values at the Beginning or End of a Report 148

Controlling the Horizontal Position of Text at the Beginning or End of a Report 151

Initializing Variables at the Beginning of a Report 153

Placing Conditional Text at the Beginning or End of a Report 156

Controlling Whether Statements in a Compute Block Execute at the Beginning and End of a Report 159

Chapter 8 · Calculating Percentages 163

Introduction 163

Calculating Percentages within Groups 166

Calculating Percentages over All Groups 169

Calculating Cumulative Percentages within Groups 172

Calculating Cumulative Percentages for the Whole Report 175

Chapter 9 · The REPORT Language 179

Overview 180

REPORT Procedure Syntax 180

PROC REPORT Statement 181

BREAK Statement 188

BY Statement 191

CALL DEFINE Statement 192

COLUMN Statement 193

COMPUTE Statement 198

DEFINE Statement 201

ENDCOMP Statement 214

FREQ Statement 214

LINE Statement 215

RBREAK Statement 217

WEIGHT Statement 220

Examples 220

Chapter 10 · How PROC REPORT Builds a Report 243

Introduction 243

Sequence of Events 243

Building a Report that Uses Groups and a Report Summary 245

Building a Report that Uses DATA Step Variables 248

Glossary 257

Index 261

Reference Aids

Displays

- 9.1 Report from which Users Can Access a More Detailed Report 240
- 9.2 Detailed Report Loaded with the EXECUTE Command 241

Figures

- 10.1 Initializing Values for the First Detail Row 246
- 10.2 Filling in Values from Left to Right 246
- 10.3 Adding a Computed Variable to the Row 247
- 10.4 First Complete Detail Row 247
- 10.5 Preliminary Summary Line 248
- 10.6 Final Summary Line 248
- 10.7 Initializing DATA Step Variables 251
- 10.8 Preliminary Summary Line for the Break at the Beginning of the Report 251
- 10.9 Final Summary Line for the Break at the Beginning of the Report 252
- 10.10 Preliminary Summary Line for the Break before the First Group of Observations 252
- 10.11 Final Summary Line for the Break before the First Group of Observations 253
- 10.12 Initializing Values for the First Detail Row 253
- 10.13 Filling in Values from Left to Right 253
- 10.14 Adding the First Computed Variable to the Row 254
- 10.15 First Complete Detail Row 255
- 10.16 Preliminary Summary Line for the Break before the Second Group of Observations 255
- 10.17 Final Summary Line for the Break before the Second Group of Observations 256

Output

- 9.1 Stacking Items with the COLUMN Statement 195
- 9.2 Creating a Header That Spans Multiple Columns 196
- 9.3 Using Aliases in the COLUMN Statement 197
- 9.4 An Across Variable with No Items Above or Below It 202
- 9.5 An Across Variable with an Analysis Variable above It 203
- 9.6 Using an Analysis Variable with Group Variables 204
- 9.7 Using an Analysis Variable with Order Variables 205

9.8	Creating Groups	208
9.9	Displaying Order Variables	209
9.10	Grouping First by DEPT, Then by DATE	212
9.11	Grouping First by DATE, Then by DEPT	212
9.12	Grouping (across) First by DATE, Then by DEPT	212
9.13	Grouping (across) First by DEPT, Then by DATE	212
9.14	Selecting and Ordering Items to Display	221
9.15	Ordering Rows by DEPT, ACCOUNT, and DATE	222
9.16	Creating a Single Row for Each Department	223
9.17	Creating a Column Header for Each Value of DATE	224
9.18	Default Placement of Long Rows	225
9.19	Wrapping Values onto Consecutive Rows	226
9.20	Ignoring Group Variables with Missing Values	227
9.21	Including Observations with Missing Values of DEPT	227
9.22	Customizing Column Headers	228
9.23	Creating a Report with Panels	229
9.24	Adding the Computed Variable BALANCE to the Report	230
9.25	Summing BUDGET, ACTUAL, and BALANCE for Each Department and Double-Spacing between Accounts	232
9.26	Summing BUDGET, ACTUAL, and BALANCE at End of the Report	233
9.27	Customizing Summary Text for Each Department	235
9.28	Wrapping the Value of COMMENT across Multiple Rows	236
9.29	Suppressing the Display of BUDGET and ACTUAL	237
10.1	Three Different Meanings of BUDGET.SUM.	244
10.2	Report with Groups and a Report Summary	246
10.3	Report with DATA Step Variables	250

Tables

9.1	Attribute Descriptions	193
9.2	Report Items that Can Share Columns	213

Acknowledgments

We are grateful to Philip Gallagher, Daryl R. Hoffman, Steve Marcus, Ray Pass, Bob Snyder, and Brad Stachoviak, users of the SAS System, who participated in the review cycles for this book.

Using This Book

Purpose

Technical Report P-258, Using the REPORT Procedure in a Nonwindowing Environment provides both usage and reference information for using the REPORT procedure in a PROC step without opening the REPORT window. The information pertains to Release 6.07 and subsequent releases.

The usage portion of the book is divided into Chapters that explain different aspects of report building. In each chapter is a series of sections, each one dealing with an isolated task. The sections are designed to be as independent of each other as possible. The usage material does not attempt to cover all the components of the REPORT procedure.

The reference portion of the book provides complete descriptions of the aspects of the REPORT language that you use in a nonwindowing environment. The reference material does not attempt to teach you how to use the procedure.

“Using This Book” contains important information that will assist you as you use this book. It discusses the level of experience you need to use the REPORT procedure, describes the organization of the book, explains various conventions used throughout the book, and lists other books that may be useful to you.

This book differs from most SAS documentation in the way it organizes and presents information. For example, the usage part of the book presents discrete examples that generally focus on one particular task and leave the rest of the report as simple as possible. To help you get the most out of the book, read “How to Use This Book” later in this chapter.

Audience

Using the REPORT Procedure in a Nonwindowing Environment is for users who understand the concepts listed in the following section, “Prerequisites.” The book assumes that the user has no knowledge of PROC REPORT. However, the material in the book is also useful to people who have experience using the procedure.

Prerequisites

The following table summarizes the SAS System concepts you need to understand in order to use *Using the REPORT Procedure in a Nonwindowing Environment*.

You need to know how to	Refer to
invoke the SAS System at your site.	instructions provided by the SAS Software Consultant at your site.
allocate SAS data libraries and assign librefs. You should have an understanding of the concepts of SAS data libraries, SAS data sets, and SAS catalogs.	<i>SAS Language: Reference, Version 6, First Edition.</i>
use base SAS software. You need varying amounts of familiarity with the SAS System, depending on the level of complexity of the reports you want to produce. Because you can use DATA step programming in PROC REPORT, the more familiar you are with the DATA step, the more flexibility you have in producing reports. However, you can produce moderately sophisticated reports with very little knowledge of the DATA step.	<i>SAS Language: Reference, Version 6, First Edition</i>

How to Use This Book

This section provides an overview of the information in this book and describes what you should read in particular situations.

Organization

Using the REPORT Procedure in a Nonwindowing Environment contains three parts: background, usage, and reference. This section describes the purpose of each part and lists the chapters in each one.

Background

The background part of the book provides information that you need in order to use either the usage or reference part of the book. It explains how to invoke PROC REPORT in a nonwindowing environment, how to specify the data set to process, and how to create the data set that most of the examples use.

Usage

The usage part of the book is a series of task-oriented chapters that explain how to accomplish particular tasks. Each chapter presents tasks related to one aspect of report writing. For the most part the chapters and the tasks within chapters are independent of one another. However, later chapters all assume that you have read Chapter 1.

Chapter 1, “Shaping the Report Layout”

lists questions to consider when you design your report. It describes how to create the layout you choose for your report.

Chapter 2, “Reports with Statistics”

shows how to use statistics in your reports.

Chapter 3, “Basic Report Enhancements”

describes how to enhance your report in simple but effective ways. Some of the tasks use parts of the SAS System that you may already be familiar with (like WHERE clauses and TITLE statements); others use features specific to the REPORT procedure.

Chapter 4, “Customizing Column Headers”

explains how to customize your column headers.

Chapter 5, “Adding Variables that Are Not in the Input Data Set”

explains how to create a column in the report for a variable that is not in the input data set.

Chapter 6, “Working with Groups of Observations”

explains how to create one row of a report that represents a group of observations, how to summarize information for a group of observations, and how to place customized text between groups.

Chapter 7, “Working with the Report as a Whole”

explains how to summarize information for the whole report and how to place customized text at the beginning or end of the report.

Chapter 8, “Calculating Percentages”

explains how to calculate percentages and cumulative percentages for groups of observations and for the report as a whole.

Reference

The reference part of the book contains information about the syntax of the REPORT procedure and about the way the procedure builds a report.

Chapter 9, “The REPORT Language”

provides detailed information about the statements, arguments, and options you can use with PROC REPORT. Examples at the end of the chapter show how to use a variety of features. These examples are more integrated than the examples in the usage material. Usage examples generally illustrate a single task. Reference examples use more of a variety of features in one report.

Chapter 10, “How PROC REPORT Builds a Report”

explains the sequence of events in building a report. The chapter steps you through the creation of two reports: one fairly simple and one moderately complex.

Chapter Structure

Chapters 1 through 8 each contain an introduction followed by a series of task-oriented sections. The introduction explains the concepts you need in order to understand the rest of the chapter. The sections following the introduction each describe a single task related to report writing. These sections are designed to contain enough information to accomplish the task, but they do not contain detailed information about the features you use or how they interact with other features. The discussion of each task includes the following subsections:

Implementation describes how to accomplish the task.

Report	shows the report the example produces. Unlike many SAS books, this one puts the output before the sample code so that you can more readily find the report and see if it includes the feature you are looking for.
Sample Program	contains the SAS code that produces the report. The features that accomplish the main task are shaded.
Building Blocks	lists other sections of the book that the example builds on. The list includes chapter numbers for sections outside the current chapter. In the usage part of the book, tasks are kept as separate as possible. However, all reports rely on some basic layout described in Chapter 1. More advanced tasks may rely on several other tasks as well.

Note: Some sections contain multiple reports and multiple pieces of sample code.

Chapter 9 describes the use of the REPORT language. The introduction to this chapter provides the high-level syntax for the REPORT procedure. A description of each statement follows. The statements are documented in alphabetic order except for the PROC REPORT statement, which appears first. The description of a statement includes the following sections:

Syntax	provides the detailed syntax for the statement.
Requirements	describes any required arguments.
Options	describes optional arguments. Options appear in alphabetic order.

Reference Aids

Using the REPORT Procedure in a Nonwindowing Environment has been organized to make information readily accessible. The following features will help you find the information you need.

Table of Contents	lists part and chapter titles and major subheadings with page numbers.
Glossary	defines the major terms used in this book.
Index	provides page numbers for specific tasks, statements, arguments, and options, as well as cross-references where appropriate.

Conventions

This section explains the various conventions used in presenting text, syntax, and examples in this book.

Typographical Conventions

Using the REPORT Procedure in a Nonwindowing Environment uses several type styles for presenting information. The following list explains the meaning of the typographical conventions used in this book.

roman	is the standard type style used for most text in this book.
UPPERCASE ROMAN	is used for SAS language elements in text.
bold	is used for headings.
<i>italic</i>	is used for terms that are defined in text, for emphasis, for user-supplied values (in text or syntax), and for references to publications.
monospace	is used for character variable values and items in windows (except pushbuttons). It is also used to show SAS System titles, labels, and footnotes in text.
code	is used to show example code. In most cases, this book uses lowercase type for SAS code, with the exception of some title characters. SAS System messages appear in mixed case.
<i>oblique</i>	is used for user-supplied values in example code.

Syntax Conventions

Type styles have special meanings when used in the presentation of PROC REPORT syntax in this book. The following list explains the style conventions for the syntax sections.

UPPERCASE BOLD	identifies SAS keywords such as the names of statements and procedures.
UPPERCASE ROMAN	identifies arguments and values that are literals.
<i>italic</i>	identifies arguments or values that you supply.

The following symbols are also syntax conventions:

< > (angle brackets)	identify optional arguments. Any argument not enclosed in angle brackets is required.
(vertical bar)	indicates that you can choose one value from a group. Values separated by bars are mutually exclusive.
. . . (ellipsis)	indicates that the argument or group of arguments following the ellipsis can be repeated any number of times. If the ellipsis and the following argument are enclosed in angle brackets, they are optional.

The following examples illustrate the syntax conventions described in this section. These examples contain selected syntax elements, not complete syntax.

```
DEFINE item /<usage>
           <attribute(s)>
           <option(s)>
           <justification>
           <'column-header-1' <. . . '>' column-header-n'>;
```

- **DEFINE** is in uppercase bold because it is a SAS keyword, the name of a statement. You must spell it as shown. The remaining elements in this statement are arguments.
- *item* is in italic because you supply the name of any item in the report. It is not enclosed in angle brackets because it is required.
- *usage* represents a list of options from which you can choose only one option. *usage* is enclosed in angle brackets because it is optional.
- *attribute(s)* and *option(s)* represent lists of options from which you can choose one or more options. Both are enclosed in angle brackets because they are optional.
- *justification* represents a list of options from which you can choose only one option. *justification* is enclosed in angle brackets because it is optional.
- *column-header-1* is enclosed in angle brackets because it is an optional argument. The words are in italic because they are values you must supply. In this case, *column-header-1* is the text for the first line of the header for the column containing *item*.
- *column-header-n* is enclosed in angle brackets because it is an optional argument. The words are in italic because they are values you must supply. In this case, *column-header-n* is the text for the *n*th line of the header for the column containing *item*. The ellipsis indicates that you can use as many column headers as you want.
- The ending semicolon (;) is not enclosed in angle brackets because it is required.
- A forward slash (/) must precede the first argument.

```
ORDER=DATA | FORMATTED | FREQ | INTERNAL
```

All words in this piece of syntax are uppercase roman because they are literals. You must use the exact spelling that appears in the syntax statement. The vertical bars indicate that the values to the right of the equal sign are mutually exclusive. If you use this option, you must choose one value.

Conventions for Examples and Output

The output in this book was created using the following OPTIONS statement, which does not appear in the individual examples:

```
options pageno=1 linesize=72 pagesize=60 nodate;
```

Some examples use options in the PROC REPORT statement to override the settings of the LINESIZE= and PAGESIZE= options.

All examples define the USER libref for the SAS data library containing the input data set. The LIBNAME statement does not appear in the individual examples. The user libref enables us to use a one-level name for the input data set. For information on the USER libref, see Chapter 6, “SAS Files,” in *SAS Language: Reference*.

Additional Documentation

SAS Institute provides many publications about products of the SAS System and how to use them on specific hosts. For a complete list of SAS publications, refer to the current *Publications Catalog*. The catalog is produced twice a year. You can order a free copy of the catalog by writing to the address or calling the telephone number below:

SAS Institute Inc.
Book Sales Department
SAS Campus Drive
Cary, NC 27513
Telephone: 919-677-8000
Fax: 919-677-8166

In addition to *Using the REPORT Procedure in a Nonwindowing Environment*, you will find these other documents helpful when using the REPORT procedure:

- *SAS Language: Reference, Version 6, First Edition* (order #A56076) provides detailed reference information about all elements of the SAS language except procedures. While you are using PROC REPORT, you may find the chapters on statements, formats, functions, and the SAS Text Editor particularly useful.
- *SAS Procedures Guide, Version 6, Third Edition* (order #A56080) provides detailed reference information about other procedures in base SAS software. In particular, you may find the chapters on the CATALOG, DATASETS, and FORMAT procedures useful.
- *SAS Guide to the REPORT Procedure: Usage and Reference, Version 6, First Edition* (order #A56088) explains how to use PROC REPORT in a windowing environment.
- *SAS Technical Report P-222, Changes and Enhancements to Base SAS Software, Release 6.07* (order #59139) describes changes to base SAS software from Release 6.06 to 6.07.
- *SAS Guide to Macro Processing, Version 6, Second Edition* (order #A56041) provides information about the Macro facility. You can use macro processing to further customize your reports.
- SAS documentation for your host operating system provides information about the operating-system-specific features of the SAS System for your operating system.

Background

Introduction 1

Invoking PROC REPORT in the Nonwindowing Environment 1

Specifying the Input Data Set 1

Creating the SAS Data Set BUDGET2 2

Introduction

The REPORT procedure combines features from the PRINT, MEANS, and TABULATE procedures with features of DATA step report writing into a powerful report-writing tool. You can use the REPORT procedure in one of three ways:

- in a windowing environment with a prompting facility that guides you as you build a report.
- in a windowing environment without the prompting facility.
- in a nonwindowing environment. In this case you submit a series of statements with the PROC REPORT statement, just as you do in other SAS procedures.

This book describes how to use PROC REPORT in a nonwindowing environment. You can store the report definitions of any reports that you create. You can then use those report definitions in either the windowing or nonwindowing environment.

Use of the windowing environment is described in *SAS Guide to the REPORT Procedure*.

Invoking PROC REPORT in the Nonwindowing Environment

To ensure that PROC REPORT sends your output to the SAS procedure output rather than the REPORT window, use the NOWINDOWS (NOWD) option in the PROC REPORT statement.

Note: If you omit the option and your output goes to the REPORT window, you can terminate PROC REPORT by selecting **File** from the action bar at the top of the REPORT window. Select **Quit** from the pull-down menu that appears. PROC REPORT opens a dialog box that asks you to confirm or cancel the QUIT command. Select the **OK** button to confirm the command.

Specifying the Input Data Set

You build a report from the data in a SAS data set. Use the DATA= option in the PROC REPORT statement to identify your input data set. The value for the DATA= option is the name of the data set to process.

Creating the SAS Data Set BUDGET2

Most of the examples in this book use the SAS data set BUDGET2. The data set contains two quarters of financial data for a company that produces videotapes. The data set contains the following variables:

DATE	is the last date of the quarter.
DEPT	is the name of a department in the company.
ACCOUNT	is the name of an account within a department.
BUDGET	is the amount of money budgeted for an account for the quarter.
ACTUAL	is the amount of money spent by an account during the quarter.

This data set is similar to the data set BUDGET that is used throughout *SAS Guide to the REPORT Procedure*.

This section contains the DATA step that creates the data set BUDGET2. In order to store the data set permanently, you must assign a libref to a permanent SAS data library and use that libref in the DATA statement. Here we use the special libref USER so that you can refer to the data set by a one-level name. Note that the DATA step assigns formats to the variables BUDGET, ACTUAL, and DATE.

This data set is available in transport format in the Online Customer Support Facility. For information on how to use this facility, see SAS Technical Report U-116.

```
libname user 'SAS-data-library';
data user.budget2;
  input date:date7. dept $ 9-18 @20 account $ @29 budget @36 actual;
  actual=actual/100;
  format budget actual dollar11.2 date date7.;
  cards;
31mar92 Staff      fulltime 130000 12764268
30jun92 Staff      fulltime 165000 16634575
31mar92 Staff      parttime 40000 4385012
30jun92 Staff      parttime 60000 5601896
31mar92 Equipment lease    40000 4000000
30jun92 Equipment lease    40000 4000000
31mar92 Equipment purchase 40000 4828238
30jun92 Equipment purchase 20000 1776915
31mar92 Equipment tape     8000 682942
30jun92 Equipment tape     12000 1142673
31mar92 Equipment sets     7500 834268
30jun92 Equipment sets     7500 807962
31mar92 Equipment maint    10000 754213
30jun92 Equipment maint    12000 1067529
31mar92 Equipment rental   4000 399887
30jun92 Equipment rental   6000 548294
31mar92 Facilities rent    24000 2400000
30jun92 Facilities rent    24000 2400000
31mar92 Facilities utils   5000 422329
30jun92 Facilities utils   3500 344481
31mar92 Facilities supplies 2750 221655
30jun92 Facilities supplies 2750 274248
```

```

31mar92 Travel leases 3500 304515
30jun92 Travel leases 4500 388965
31mar92 Travel gas 800 53726
30jun92 Travel gas 1200 98493
31mar92 Other advert 30000 3247698
30jun92 Other advert 30000 3732564
31mar92 Other talent 13500 1298673
30jun92 Other talent 19500 1842464
31mar92 Other musicfee 3000 255050
30jun92 Other musicfee 5000 487595
;

```

The following PROC REPORT step creates the listing of the data set that appears in Output 1:

```

libname user 'SAS-data-library';
proc report data=budget2 nowindows;
  title 'The SAS Data Set BUDGET2';
run;

```

Output 1
The SAS Data Set
BUDGET2

The SAS Data Set BUDGET2					1
DATE	DEPT	ACCOUNT	BUDGET	ACTUAL	
31MAR92	Staff	fulltime	\$130,000.00	\$127,642.68	
30JUN92	Staff	fulltime	\$165,000.00	\$166,345.75	
31MAR92	Staff	parttime	\$40,000.00	\$43,850.12	
30JUN92	Staff	parttime	\$60,000.00	\$56,018.96	
31MAR92	Equipment	lease	\$40,000.00	\$40,000.00	
30JUN92	Equipment	lease	\$40,000.00	\$40,000.00	
31MAR92	Equipment	purchase	\$40,000.00	\$48,282.38	
30JUN92	Equipment	purchase	\$20,000.00	\$17,769.15	
31MAR92	Equipment	tape	\$8,000.00	\$6,829.42	
30JUN92	Equipment	tape	\$12,000.00	\$11,426.73	
31MAR92	Equipment	sets	\$7,500.00	\$8,342.68	
30JUN92	Equipment	sets	\$7,500.00	\$8,079.62	
31MAR92	Equipment	maint	\$10,000.00	\$7,542.13	
30JUN92	Equipment	maint	\$12,000.00	\$10,675.29	
31MAR92	Equipment	rental	\$4,000.00	\$3,998.87	
30JUN92	Equipment	rental	\$6,000.00	\$5,482.94	
31MAR92	Facilities	rent	\$24,000.00	\$24,000.00	
30JUN92	Facilities	rent	\$24,000.00	\$24,000.00	
31MAR92	Facilities	utils	\$5,000.00	\$4,223.29	
30JUN92	Facilities	utils	\$3,500.00	\$3,444.81	
31MAR92	Facilities	supplies	\$2,750.00	\$2,216.55	
30JUN92	Facilities	supplies	\$2,750.00	\$2,742.48	
31MAR92	Travel	leases	\$3,500.00	\$3,045.15	
30JUN92	Travel	leases	\$4,500.00	\$3,889.65	
31MAR92	Travel	gas	\$800.00	\$537.26	
30JUN92	Travel	gas	\$1,200.00	\$984.93	
31MAR92	Other	advert	\$30,000.00	\$32,476.98	
30JUN92	Other	advert	\$30,000.00	\$37,325.64	
31MAR92	Other	talent	\$13,500.00	\$12,986.73	
30JUN92	Other	talent	\$19,500.00	\$18,424.64	
31MAR92	Other	musicfee	\$3,000.00	\$2,550.50	
30JUN92	Other	musicfee	\$5,000.00	\$4,875.95	

Chapter 1 Shaping the Report Layout

Introduction 5

Statements that Determine the Layout 5

Ways You Can Use a Data Set Variable 6

Statistics You Can Use 7

When a Usage Option Is Unnecessary 7

Listing All Observations and All Variables 8

Specifying and Ordering Columns to Display 11

Ordering Rows 13

Consolidating Multiple Observations into One Row 16

Using Variable Values as Column Headers 19

Introduction

Report writing is simplified if you approach it with a clear understanding of what you want your report to look like. The most important thing to figure out is the layout of the report. To determine the layout ask yourself the following kinds of questions:

- What do I want to display in each column of the report?
- In what order do I want the columns to appear?
- Do I want to display a column for each value of a particular variable?
- Do I want a row for every observation in the data set?
- Do I want to consolidate information for groups of observations into one row?
- In what order do I want the rows to appear?

Once you understand the basics of structuring a report, you can go on to enhance it. This chapter shows you how to create the most fundamental types of reports. Later chapters show you how to enhance them. Enhancements include controlling formats, column width, and spacing; adding statistics and variables to the report that are not in the data set; summarizing information for groups of observations or for the whole data set.

Statements that Determine the Layout

Two statements determine the layout of the report:

- The COLUMN statement lists the items that appear in the columns of the report and describes the arrangement of the columns. You do not need this statement if you want to include all variables in the input data set in the same order as they occur in the data set.
- The DEFINE statement defines the characteristics of an item in the report. These characteristics include the column header, the format to use to display values, and how PROC REPORT uses the item in the report.

A report *item* can be

- a data set variable
- a statistic calculated by the procedure
- a variable you compute based on other items in the report.

In discussing report layouts, this book often refers to report items by the way they are used. For instance, we talk about group variables, analysis variables, computed variables, and statistics.

Ways You Can Use a Data Set Variable

Much of a report's layout is determined by the way you use data set variables in the report. You tell PROC REPORT how to use a variable by specifying a usage option in the DEFINE statement for that variable.

If you want the values of the variable to ...	then use this usage option in the DEFINE statement ...	For an example, see this section later in this chapter ...
appear as they do in the input data set	DISPLAY	“Listing All Observations and All Variables”
form column headers	ACROSS	“Using Variable Values as Column Headers”
determine the order of the rows in the report	ORDER	“Ordering Rows”
consolidate into one row all observations from the input data set that have the same value for the variable	GROUP	“Consolidating Multiple Observations into One Row”
calculate a statistic for all observations that have a unique combination of values for all group variables*	ANALYSIS	“Consolidating Multiple Observations into One Row”

*If the report does not contain groups, the value of an analysis variable is the value of a single observation.

The layout of a report depends not only on the usage of each item in the report but also on the interactions between the positions and usages of the items in the report. These interactions are explained in “Interactions of Position and Usage” in the documentation for the DEFINE statement in Chapter 9.

Statistics You Can Use

If you use the ANALYSIS option, you must also specify a statistic in the DEFINE statement. Available statistics are

N	RANGE	T
NMISS	SUM	PRT
MEAN	USS	SUMWGT
STD	CSS	VAR
MIN	STDERR	
MAX	CV	

Explanations of these statistics are in the introduction to Chapter 2.

Naming a statistic in the DEFINE statement implies the ANALYSIS option, so you never need to specify ANALYSIS, although specifying it may make your code easier for novice users to understand.

When a Usage Option Is Unnecessary

By default, PROC REPORT assigns a usage to each data set variable.

- Character variables are display variables.
- Numeric variables are analysis variables used to calculate the SUM statistic.

If you want the default usage, you do not need to specify a usage option in the DEFINE statement. If you do not need to specify any other characteristics for the item, you can omit the DEFINE statement entirely.

Note: In this book, if any item in a report requires a usage option, we show DEFINE statements and usage options for all items. However, we never highlight a usage option that reflects the default behavior.

Listing All Observations and All Variables

Implementation

By default, PROC REPORT includes all data set variables in the report. By default, character variables are display variables. A report that contains one or more display variables has a row for every observation in the data set. Therefore, the method of listing all observations in the data set depends on whether the report contains any character variables.

- If the report contains any character variables, run PROC REPORT with the NOWINDOWS and DATA= options.
- If the report contains only numeric variables
 - run PROC REPORT with the NOWINDOWS and DATA= options
 - define at least one of the variables as a display variable.

Report 1: A Report with Character and Numeric Variables

A Report with Character and Numeric Variables					1
DATE	DEPT	ACCOUNT	BUDGET	ACTUAL	
31MAR92	Staff	fulltime	\$130,000.00	\$127,642.68	
30JUN92	Staff	fulltime	\$165,000.00	\$166,345.75	
31MAR92	Staff	parttime	\$40,000.00	\$43,850.12	
30JUN92	Staff	parttime	\$60,000.00	\$56,018.96	
31MAR92	Equipment	lease	\$40,000.00	\$40,000.00	
30JUN92	Equipment	lease	\$40,000.00	\$40,000.00	
31MAR92	Equipment	purchase	\$40,000.00	\$48,282.38	
30JUN92	Equipment	purchase	\$20,000.00	\$17,769.15	
31MAR92	Equipment	tape	\$8,000.00	\$6,829.42	
30JUN92	Equipment	tape	\$12,000.00	\$11,426.73	
31MAR92	Equipment	sets	\$7,500.00	\$8,342.68	
30JUN92	Equipment	sets	\$7,500.00	\$8,079.62	
31MAR92	Equipment	maint	\$10,000.00	\$7,542.13	
30JUN92	Equipment	maint	\$12,000.00	\$10,675.29	
31MAR92	Equipment	rental	\$4,000.00	\$3,998.87	
30JUN92	Equipment	rental	\$6,000.00	\$5,482.94	
31MAR92	Facilities	rent	\$24,000.00	\$24,000.00	
30JUN92	Facilities	rent	\$24,000.00	\$24,000.00	
31MAR92	Facilities	utils	\$5,000.00	\$4,223.29	
30JUN92	Facilities	utils	\$3,500.00	\$3,444.81	
31MAR92	Facilities	supplies	\$2,750.00	\$2,216.55	
30JUN92	Facilities	supplies	\$2,750.00	\$2,742.48	
31MAR92	Travel	leases	\$3,500.00	\$3,045.15	
30JUN92	Travel	leases	\$4,500.00	\$3,889.65	
31MAR92	Travel	gas	\$800.00	\$537.26	
30JUN92	Travel	gas	\$1,200.00	\$984.93	
31MAR92	Other	advert	\$30,000.00	\$32,476.98	
30JUN92	Other	advert	\$30,000.00	\$37,325.64	
31MAR92	Other	talent	\$13,500.00	\$12,986.73	
30JUN92	Other	talent	\$19,500.00	\$18,424.64	
31MAR92	Other	musicfee	\$3,000.00	\$2,550.50	
30JUN92	Other	musicfee	\$5,000.00	\$4,875.95	

Sample Program 1

```
proc report data=budget2 nowindows;
    title 'A Report with Character and Numeric Variables';
run;
```

Report 2: A Report with Only Numeric Variables

By default, PROC REPORT treats numeric variables as analysis variables used to calculate the SUM statistic. Therefore, if the report contains only numeric variables and you do not explicitly define any of them, PROC REPORT sums the value of each variable over all observations in the data set and produces a one-line summary report of the sums, like page 1 of this report. If, on the other hand, you define at least one of the variables as a display variable, PROC REPORT produces a report that lists all observations and variables, like page 2 of this report.

A Report with Only Numeric Variables:		1
All Analysis Variables (by Default)		
BUDGET	ACTUAL	
\$775,000.00	\$780,011.28	

A Report with Only Numeric Variables:		2
Defining One Variable as a Display Variable		
BUDGET	ACTUAL	
\$130,000.00	\$127,642.68	
\$165,000.00	\$166,345.75	
\$40,000.00	\$43,850.12	
\$60,000.00	\$56,018.96	
\$40,000.00	\$40,000.00	
\$40,000.00	\$40,000.00	
\$40,000.00	\$48,282.38	
\$20,000.00	\$17,769.15	
\$8,000.00	\$6,829.42	
\$12,000.00	\$11,426.73	
\$7,500.00	\$8,342.68	
\$7,500.00	\$8,079.62	
\$10,000.00	\$7,542.13	
\$12,000.00	\$10,675.29	
\$4,000.00	\$3,998.87	
\$6,000.00	\$5,482.94	
\$24,000.00	\$24,000.00	
\$24,000.00	\$24,000.00	
\$5,000.00	\$4,223.29	
\$3,500.00	\$3,444.81	
\$2,750.00	\$2,216.55	
\$2,750.00	\$2,742.48	
\$3,500.00	\$3,045.15	
\$4,500.00	\$3,889.65	
\$800.00	\$537.26	
\$1,200.00	\$984.93	
\$30,000.00	\$32,476.98	
\$30,000.00	\$37,325.64	
\$13,500.00	\$12,986.73	
\$19,500.00	\$18,424.64	
\$3,000.00	\$2,550.50	
\$5,000.00	\$4,875.95	

Sample Program 2

```
proc report data=budget2(keep=budget actual) nowindows;
  title 'A Report with Only Numeric Variables: ';
  title2 'All Analysis Variables (by Default)';
run;
proc report data=budget2(keep=budget actual) nowindows;
  define budget / display;
  define actual / analysis sum;

  title 'A Report with Only Numeric Variables: ';
  title2 'Defining One Variable as a Display Variable';
run;
```

Program Notes

- The KEEP= data set option in the PROC REPORT statement selects the variables to process. Data set options work with PROC REPORT as they do with the rest of the SAS System. For information on data set options, see Chapter 15, “SAS Data Set Options,” in *SAS Language: Reference, Version 6, First Edition*.

Specifying and Ordering Columns to Display

Implementation

- List the items in the desired order in the COLUMN statement.

If you do not use a COLUMN statement, a report contains a column for each variable in the input data set. The columns are in the same order as the variables are in the data set. When you use the COLUMN statement, only items you list in the statement appear in the report. They appear in the order you list them in the COLUMN statement.

Note: You can select variables with the DROP= and KEEP= data set options, but you cannot control the order of the columns with these options. If you do not use a COLUMN statement, the variables always appear in the order they have in the input data set. For information on data set options, see Chapter 15, “SAS Data Set Options,” in *SAS Language: Reference*.

Report

Specifying and Ordering Columns to Display			1
ACCOUNT	DATE	BUDGET	
fulltime	31MAR92	\$130,000.00	
fulltime	30JUN92	\$165,000.00	
parttime	31MAR92	\$40,000.00	
parttime	30JUN92	\$60,000.00	
lease	31MAR92	\$40,000.00	
lease	30JUN92	\$40,000.00	
purchase	31MAR92	\$40,000.00	
purchase	30JUN92	\$20,000.00	
tape	31MAR92	\$8,000.00	
tape	30JUN92	\$12,000.00	
sets	31MAR92	\$7,500.00	
sets	30JUN92	\$7,500.00	
maint	31MAR92	\$10,000.00	
maint	30JUN92	\$12,000.00	
rental	31MAR92	\$4,000.00	
rental	30JUN92	\$6,000.00	
rent	31MAR92	\$24,000.00	
rent	30JUN92	\$24,000.00	
utils	31MAR92	\$5,000.00	
utils	30JUN92	\$3,500.00	
supplies	31MAR92	\$2,750.00	
supplies	30JUN92	\$2,750.00	
leases	31MAR92	\$3,500.00	
leases	30JUN92	\$4,500.00	
gas	31MAR92	\$800.00	
gas	30JUN92	\$1,200.00	
advert	31MAR92	\$30,000.00	
advert	30JUN92	\$30,000.00	
talent	31MAR92	\$13,500.00	
talent	30JUN92	\$19,500.00	
musicfee	31MAR92	\$3,000.00	
musicfee	30JUN92	\$5,000.00	

Sample Program

```
proc report data=budget2 nowindows;  
  column account date budget;  
  
  title 'Specifying and Ordering Columns to Display';  
run;
```

Ordering Rows

Implementation

- Use the ORDER option in one or more DEFINE statements to specify the sequence of the rows in the report.
- To modify the default ordering sequence, use these additional options in the DEFINE statement:
 - ORDER=
 - DESCENDING

By default, PROC REPORT orders the rows of a report according to the values of the order variables. If the report contains multiple order variables, the REPORT procedure first orders rows according to the values of the first order variable in the COLUMN statement. Within each value for the first order variable, the procedure orders rows according to the values of the second order variable in the COLUMN statement, and so forth. If you do not use a COLUMN statement, PROC REPORT processes the order variables according to their positions in the input data set.

The ORDER= option specifies the sort order for a variable. The default is ORDER=FORMATTED.

► **Caution** *Default for the ORDER= Option*

In other SAS procedures, the default is ORDER=INTERNAL. The default for the option in PROC REPORT may change in a future release to be consistent with other procedures. Therefore, in production jobs where it is important to order report items by their formatted values, specify ORDER=FORMATTED even though it is currently the default. Doing so ensures that PROC REPORT will continue to produce the reports you expect even if the default changes. ▲

If you want to order observations by their ...	then use ORDER= ...
order in the input data set	DATA
formatted value (ascending)	FORMATTED
frequency count (ascending)	FREQ
nonformatted value (ascending). This sort sequence is the same one that PROC SORT would use. It is particularly useful for displaying dates chronologically.	INTERNAL

The DESCENDING option reverses the sort sequence for an item.

The ORDER= option also controls

- the ordering of across variables (see “Using Variable Values as Column Headers” later in this chapter)

- the ordering of group variables (see “Consolidating Multiple Observations into One Row” later in this chapter).

Note: The ORDER option determines a variable’s usage and gives that variable a role in determining the sequence of rows in the report. The ORDER= option specifies the method to use to determine that sequence (formatted value, frequency count, and so forth). Do not use the ORDER= option unless the DEFINE statement you use it in also contains the ORDER, GROUP, or ACROSS option.

Report

In this report DATE, DEPT, and ACCOUNT are order variables. PROC REPORT orders rows by

1. ascending internal values of DATE (SAS date values)
2. ascending formatted values of DEPT
3. descending formatted values of ACCOUNT.

PROC REPORT does not repeat the value of an order variable from one row to the next when the same value applies.

Ordering Rows					1	
DATE	DEPT	ACCOUNT	BUDGET	ACTUAL		
31MAR92	Equipment	tape	\$8,000.00	\$6,829.42		
		sets	\$7,500.00	\$8,342.68		
		rental	\$4,000.00	\$3,998.87		
		purchase	\$40,000.00	\$48,282.38		
		maint	\$10,000.00	\$7,542.13		
		lease	\$40,000.00	\$40,000.00		
	Facilities	utils	\$5,000.00	\$4,223.29		
		supplies	\$2,750.00	\$2,216.55		
		rent	\$24,000.00	\$24,000.00		
	Other	talent	\$13,500.00	\$12,986.73		
		musicfee	\$3,000.00	\$2,550.50		
		advert	\$30,000.00	\$32,476.98		
	Staff	parttime	\$40,000.00	\$43,850.12		
		fulltime	\$130,000.00	\$127,642.68		
	Travel	leases	\$3,500.00	\$3,045.15		
		gas	\$800.00	\$537.26		
	30JUN92	Equipment	tape	\$12,000.00	\$11,426.73	
			sets	\$7,500.00	\$8,079.62	
rental			\$6,000.00	\$5,482.94		
purchase			\$20,000.00	\$17,769.15		
maint			\$12,000.00	\$10,675.29		
lease			\$40,000.00	\$40,000.00		
Facilities		utils	\$3,500.00	\$3,444.81		
		supplies	\$2,750.00	\$2,742.48		
		rent	\$24,000.00	\$24,000.00		
Other		talent	\$19,500.00	\$18,424.64		
		musicfee	\$5,000.00	\$4,875.95		
		advert	\$30,000.00	\$37,325.64		
Staff		parttime	\$60,000.00	\$56,018.96		
		fulltime	\$165,000.00	\$166,345.75		
Travel		leases	\$4,500.00	\$3,889.65		
		gas	\$1,200.00	\$984.93		

Sample Program

```
proc report data=budget2 nowindows;
  define date    / order order=internal;
  define dept    / order;
  define account / order descending;
  define budget  / analysis sum;
  define actual  / analysis sum;

  title 'Ordering Rows';
run;
```

Consolidating Multiple Observations into One Row

Implementation

- Use the GROUP option in one or more DEFINE statements to identify variables to use to form groups.
- Specify a statistic in the DEFINE statement for each analysis variable. (By default, all numeric variables are analysis variables used to calculate the SUM statistic.)

A group is a set of observations that has a unique combination of values for all group variables. PROC REPORT tries to consolidate, or summarize, each group in one row of the report. What values does it use in a row that represents a group of observations? The values of the group variables are, by definition, common to all observations in a group, so PROC REPORT uses those values in the row. However, each observation may have different values for analysis variables. Therefore, each analysis variable has a statistic associated with it. (The default is SUM). The value of an analysis variable for a group is the value of that statistic calculated for all observations in the group. For a list of available statistics, see “Statistics You Can Use” in the introduction to this chapter.

By default, PROC REPORT orders the rows of a report according to the formatted values of the group variables. If the report contains multiple group variables, the REPORT procedure first orders rows according to the values of the first group variable in the COLUMN statement. Within each value for the first group variable, the procedure orders rows according to the values of the second group variable in the COLUMN statement, and so forth. If you do not use a COLUMN statement, PROC REPORT processes the group variables according to their positions in the input data set. For information on using the ORDER= option to alter the sort order for a group variable, see “Ordering Rows” earlier in this chapter.

If PROC REPORT cannot create groups, it returns a message and displays group variables the same way that it displays order variables. For more information about creating groups, see Chapter 6, “Working with Groups of Observations,” and the discussion of the GROUP option and “Interactions of Position and Usage” in the documentation for the DEFINE statement in Chapter 9.

Note: Other base SAS procedures implement the same concept with the CLASS statement.

Report 1: Groups Based on the Value of a Single Variable

In this report DEPT is a group variable. Each row of the report represents one group. A group is all observations with a unique value of DEPT. For example, the first row represents all observations for which the value of DEPT is **Equipment**.

BUDGET is an analysis variable used to calculate the SUM statistic. The value of BUDGET in the first row is the sum of BUDGET for all observations where DEPT='Equipment'.

Groups Based on the Value of a Single Variable		1
DEPT	BUDGET	
Equipment	\$207,000.00	
Facilities	\$62,000.00	
Other	\$101,000.00	
Staff	\$395,000.00	
Travel	\$10,000.00	

Sample Program 1

```
proc report data=budget2 nowindows;
  column dept budget;

  define dept / group;
  define budget / analysis sum;

  title 'Groups Based on the Value of a Single Variable';
run;
```

Report 2: Groups Based on the Value of Multiple Variables

In this report both DEPT and ACCOUNT are group variables. Each row of the report represents one group. A group is all observations with the same values of DEPT and ACCOUNT. For example, the first row represents all observations for which the value of DEPT is **Equipment** and the value of ACCOUNT is **lease**.

BUDGET is an analysis variable used to calculate the SUM statistic. The value of BUDGET in the first row is the sum of BUDGET for all observations where DEPT='Equipment' and ACCOUNT='lease'.

PROC REPORT does not repeat the value of a group variable from one row to the next when the same value applies.

Groups Based on the Value of Multiple Variables			1
DEPT	ACCOUNT	BUDGET	
Equipment	lease	\$80,000.00	
	maint	\$22,000.00	
	purchase	\$60,000.00	
	rental	\$10,000.00	
	sets	\$15,000.00	
	tape	\$20,000.00	
Facilities	rent	\$48,000.00	
	supplies	\$5,500.00	
	utils	\$8,500.00	
Other	advert	\$60,000.00	
	musicfee	\$8,000.00	
	talent	\$33,000.00	
Staff	fulltime	\$295,000.00	
	parttime	\$100,000.00	
Travel	gas	\$2,000.00	
	leases	\$8,000.00	

Sample Program 2

```
proc report data=budget2 nowindows;
  column dept account budget;

  define dept / group;
  define account / group;
  define budget / analysis sum;

  title 'Groups Based on the Value of Multiple Variables';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display

Using Variable Values as Column Headers

Implementation

- Use the ACROSS option in one or more DEFINE statements to identify variables whose values will be column headers.
- If you want to see the value of an analysis variable in the columns created by the across variable, place that variable next to the across variable in the COLUMN statement. Separate the across variable from the analysis variable with a comma.
Note: If you want to specify multiple analysis variables, list their names in parentheses next to the across variable in the COLUMN statement. Separate the across variable and the adjacent parenthesis with a comma. Separate the analysis variables with blanks.
- Specify a statistic in the DEFINE statement of each analysis variable. The default is SUM.

If you place the across variable before the analysis variable, the name and values of the across variable are above the name of the analysis variable in the report (see Report 2). If you place the across variable after the analysis variable, the name and values of the across variable are below the name of the analysis variable (see Report 3). In either case, we say that these variables share the columns.

Columns created by an across variable contain statistics or computed values. If nothing is above or below an across variable, PROC REPORT displays the N statistic. This statistic is the number of observations in the input data set that belong to a cell of the report. A *cell* is a single unit of a report, formed by the intersection of a row and a column.

The examples in this chapter show how to display frequency counts (the N statistic) and analysis variables with an across variable. For information on placing computed variables above or below an across variable, see “Adding a Variable Computed from a Variable that Shares a Column with an Across Variable” in Chapter 5. For more information on sharing columns see “Interactions of Position and Usage” in the documentation for the DEFINE statement in Chapter 9.

Report 1: Showing Frequency Counts

In this report DATE is an across variable with nothing above or below it. The report shows how many observations the input data set contains for each department and date. For example, it contains six observations for the Equipment department on each date.

Showing Frequency Counts			1
	DATE		
DEPT	31MAR92	30JUN92	
Equipment	6	6	
Facilities	3	3	
Other	3	3	
Staff	2	2	
Travel	2	2	

Sample Program 1

```
proc report data=budget2 nowindows;
  column dept date;

  define dept / group;
  define date / across order=internal;

  title 'Showing Frequency Counts';
run;
```

Program Notes

- By default, PROC REPORT orders the columns created by an across variable according to the formatted values of that variable. For information on using the ORDER= option in the DEFINE statement to alter the sort order for an across variable, see “Ordering Rows” earlier in this chapter.

Report 2: Sharing a Column with an Analysis Variable

In this report DATE is an across variable that shares its columns with BUDGET, an analysis variable used to calculate the SUM statistic. Each value in the column for a particular date is the sum of all budgets in one department for the quarter ending on that date. For example, the total budget for all accounts in the Equipment department for the first quarter was \$109,500; the budget for these accounts in the second quarter was \$97,500.

Sharing a Column with an Analysis Variable			1
DEPT	DATE		
	31MAR92	30JUN92	
	BUDGET	BUDGET	
Equipment	\$109,500.00	\$97,500.00	
Facilities	\$31,750.00	\$30,250.00	
Other	\$46,500.00	\$54,500.00	
Staff	\$170,000.00	\$225,000.00	
Travel	\$4,300.00	\$5,700.00	

Sample Program 2

```
proc report data=budget2 nowindows;
  column dept date, budget;

  define dept / group;
  define date / across order=internal;
  define budget / analysis sum;

  title 'Sharing a Column with an Analysis Variable';
run;
```

Report 3: Sharing a Column with Multiple Analysis Variables

In this report DATE is an across variable that shares columns with BUDGET and ACTUAL. Both BUDGET and ACTUAL are analysis variables used to calculate the SUM statistic. The values in the column for a particular date are the totals of all budgets and all expenditures in one department for the quarter ending on that date. For example, the total budget for all accounts in the Equipment department for the first quarter was \$109,500; the total spent was \$114,995.48. The budget for these accounts in the second quarter was \$97,500; the total spent was \$93,433.73.

DEPT	BUDGET		ACTUAL	
	DATE		DATE	
	31MAR92	30JUN92	31MAR92	30JUN92
Equipment	\$109,500.00	\$97,500.00	\$114,995.48	\$93,433.73
Facilities	\$31,750.00	\$30,250.00	\$30,439.84	\$30,187.29
Other	\$46,500.00	\$54,500.00	\$48,014.21	\$60,626.23
Staff	\$170,000.00	\$225,000.00	\$171,492.80	\$222,364.71
Travel	\$4,300.00	\$5,700.00	\$3,582.41	\$4,874.58

Sample Program 3

```
proc report data=budget2 nowindows;
  column dept (budget actual) date;

  define dept / group;
  define date / across order=internal width=11;
  define budget / analysis sum;
  define actual / analysis sum;

  title 'Sharing a Column with Multiple Analysis Variables';
run;
```

Program Notes

- If items that share a column have different widths, PROC REPORT uses the width of the item closest to the first row of the report. In this case, the width of DATE is not large enough to accommodate the formatted values of BUDGET. The WIDTH= option in the DEFINE statement for DATE extends the column width to 11 characters.

Building Blocks

- Specifying and Ordering Columns to Display
- Ordering Rows
- Consolidating Multiple Observations into One Row
- Specifying Column Widths in Chapter 3

Chapter 2 Reports with Statistics

Introduction 23

Statistics You Can Use with PROC REPORT 23

Placing Statistics above or below a Variable 25

Reporting Multiple Statistics for One Variable 27

Showing Frequency Counts 29

Introduction

In general, when you include a statistic in a report, you must associate the statistic with a variable. There are two ways to do so:

- Place the statistic above or below a display, analysis, or across variable.
- Specify the statistic as a usage option in the DEFINE statement for an analysis variable.

The N statistic is the only statistic that you do not need to associate with a variable. If you include a column for the N statistic, each value in that column represents the number of observations described in that cell of the report. A *cell* is a single unit of a report, formed by the intersection of a row and a column.

For information on calculating totals and subtotals for statistics, see “Summarizing Variables and Statistics for the Whole Report” in Chapter 7 and “Summarizing Variables and Statistics for Groups” in Chapter 6.

Statistics You Can Use with PROC REPORT

The REPORT procedure can calculate these statistics:

N	RANGE	T
NMISS	SUM	PRT
MEAN	USS	SUMWGT
STD	CSS	VAR
MIN	STDERR	
MAX	CV	

Definitions of the statistics follow. The following notations are used where summation is over all nonmissing values:

x_i	the i th nonmissing observation on the variable.
w_i	the weight associated with x_i if a WEIGHT statement is specified, and 1 otherwise.
n	the number of nonmissing observations.
\bar{x}	$\sum w_i x_i / \sum w_i$.

$$s^2 = \frac{\sum w_i (x_i - \bar{x})^2}{(n - 1)}$$

$$z_i = (x_i - \bar{x}) / s \text{ standardized variables.}$$

The statistics are

N the number of observations with no missing values for a group, order, or across variable; or the number of nonmissing values for an analysis variable.

If you use the MISSING option in the PROC REPORT statement, N includes observations with missing group, order, or across variables.

NMISS the number of missing values for an analysis variable.

MEAN \bar{x} , the arithmetic mean.

STD s , the standard deviation.

MIN the minimum value.

MAX the maximum value.

RANGE MAX – MIN, the range.

SUM $\sum w_i x_i$, the weighted sum.

USS $\sum w_i x_i^2$, the uncorrected sum of squares.

CSS $\sum w_i (x_i - \bar{x})^2$, the sum of squares corrected for the mean.

STDERR s / \sqrt{n} , the standard error of the mean.

SUMWGT $\sum w_i$, the sum of weights.

CV $100 s / \bar{x}$, the percent coefficient of variation.

T $t = \bar{x} - \sqrt{n} / s$, Student's t for H_0 : population mean=0.

PRT the two-tailed p -value for Student's t with $n - 1$ degrees of freedom, the probability under the null hypothesis of obtaining an absolute value of t greater than the t -value observed in this sample.

VAR s^2 , the variance.

Placing Statistics above or below a Variable

Implementation

- Place the statistic next to a display, analysis, or across variable in the COLUMN statement. Separate the statistic from the variable with a comma.

Note: If you want to specify multiple statistics, list their names in parentheses next to the appropriate variable in the COLUMN statement. Separate the variable and the adjacent parenthesis with a comma. Separate the statistics with blanks.

If no analysis variable appears above or below a statistic, the only appropriate statistic is N.

The comma tells PROC REPORT to stack the items it separates. If you place the variable before the statistics, the name of the variable appears once, above the statistic, in the report. If you place the variable after the statistics, the name of the variable appears below the name of each statistic (see the report in this section).

► **Caution** *DEFINE Statements for Statistics*

When a statistic appears in a COLUMN statement, use a DEFINE statement for that statistic only if you want to specify things like formats and column headers. If you use a DEFINE statement, do not specify a usage option. The name of the statistic determines its usage. ▲

Report: Placing Statistics above an Analysis Variable

In this report the MIN and MAX statistics are above BUDGET.

Placing Statistics above an Analysis Variable			1
DEPT	MIN BUDGET	MAX BUDGET	
Equipment	\$4,000.00	\$40,000.00	
Facilities	\$2,750.00	\$24,000.00	
Other	\$3,000.00	\$30,000.00	
Staff	\$40,000.00	\$165,000.00	
Travel	\$800.00	\$4,500.00	

Sample Program

```
proc report nowindows data=budget2;
  column dept (min max), budget;

  define dept / group;
  define budget / analysis sum;

  title 'Placing Statistics above an Analysis Variable';
run;
```

Program Notes

- BUDGET is an analysis variable used to calculate the SUM statistic. However, the SUM statistic does not appear in the report. If you place statistics above or below an analysis variable, the usage option in the DEFINE statement for the analysis variable does not affect the report. The statistics you place in the COLUMN statement are the ones that appear in the report.

Building Blocks

- *Specifying and Ordering Columns to Display* in Chapter 1
- *Consolidating Multiple Observations into One Row* in Chapter 1

Reporting Multiple Statistics for One Variable

Implementation

- Make an entry for each use of the variable in the COLUMN statement. Use an alias so that you can have a distinct DEFINE statement for each occurrence of the variable. An alias has the form

name=alias

where *name* is the name of the item and *alias* is the alias to use in the DEFINE statement for that column.

- Write a DEFINE statement for each alias. Specify a different statistic for each one.

Note: You can use aliases to create multiple definitions for the same variable, not just to show multiple statistics. For example, you may want to show the same variable with different formats in two different columns of the report (see “Presenting One Variable in Two Different Ways” in Chapter 3).

Report

This report uses BUDGET as an analysis variable three times: once to calculate the MIN statistic, once to calculate MAX, and once to calculate N.

In addition to using BUDGET three times, this report customizes the column header for each of the three columns. Otherwise each column would have the same header, BUDGET.

Reporting Multiple Statistics for One Variable				1
DEPT	Minimum Budget	Maximum Budget	N	
Equipment	\$4,000.00	\$40,000.00	12	
Facilities	\$2,750.00	\$24,000.00	6	
Other	\$3,000.00	\$30,000.00	6	
Staff	\$40,000.00	\$165,000.00	4	
Travel	\$800.00	\$4,500.00	4	

Sample Program

```
proc report data=budget2 nowindows;
  column dept budget=budmin budget=budmax budget=nobs;

  define dept / group;
  define budmin / analysis min 'Minimum Budget';
  define budmax / analysis max 'Maximum Budget';
  define nobs / analysis n format=2. 'N';

  title 'Reporting Multiple Statistics for One Variable';
run;
```

Program Notes

- The aliases in the COLUMN statement enable you to write a distinct DEFINE statement for each use of BUDGET.
- Each DEFINE statement specifies a statistic (MIN, MAX, or N) and a column header (in quotation marks).
- The DEFINE statement for the N statistic also specifies a format.

Building Blocks

- Consolidating Multiple Observations into One Row in Chapter 1
- Specifying Formats in Chapter 3
- Customizing the Text in Column Headers in Chapter 4

Showing Frequency Counts

Implementation

- Include the N statistic in the COLUMN statement. It can appear independently or it can share a column with a display, analysis, or across variable.
- Use N as the statistic in the DEFINE statement for an analysis variable.

The reports in this section show you several different report layouts that use the N statistic in the COLUMN statement. For an example that uses the N statistic in the DEFINE statement for an analysis variable, see “Reporting Multiple Statistics for One Variable” earlier in this chapter.

► **Caution** *DEFINE Statements for Statistics*

A statistic can appear in a COLUMN statement as a report item or in a DEFINE statement as a statistic for an analysis variable. If it appears in the COLUMN statement, use a DEFINE statement only if you want to specify things like formats and column headers. Do not specify a usage option in the DEFINE statement for a statistic. The name of the statistic tells PROC REPORT how to use it. ▲

Report 1: Counting All the Observations that a Row Represents

In this report, the N statistic is in a column by itself. The numbers in the N column tell how many observations each row of the report represents.

Counting All the Observations that a Row Represents			1
DEPT	BUDGET	N	
Equipment	\$207,000.00	12	
Facilities	\$62,000.00	6	
Other	\$101,000.00	6	
Staff	\$395,000.00	4	
Travel	\$10,000.00	4	

Sample Program

```
proc report nowindows data=budget2;
  column dept budget n;

  define dept / group;
  define budget / analysis sum;

  title 'Counting All the Observations that a Row Represents';
run;
```

Report 2: Counting the Observations for Each Value of an Across Variable

In this report, the N statistic and BUDGET are side-by-side under DATE. The numbers in the columns for N tell how many observations there are for each department on each date. This is the number of observations that were used to calculate BUDGET for that date.

Counting the Observations for Each Value of an Across Variable					1
DEPT	DATE		N	BUDGET	N
	31MAR92	30JUN92			
Equipment	\$109,500.00		6	\$97,500.00	6
Facilities	\$31,750.00		3	\$30,250.00	3
Other	\$46,500.00		3	\$54,500.00	3
Staff	\$170,000.00		2	\$225,000.00	2
Travel	\$4,300.00		2	\$5,700.00	2

Sample Program 2

```
proc report nowindows data=budget2;
  column dept date, (budget n);

  define dept / group;
  define date / across order=internal;
  define budget / analysis sum;

  title 'Counting the Observations for Each Value';
  title2 'of an Across Variable';
run;
```

Report 3: Combining Report 1 and Report 2

This report combines the usages of the N statistic in the two previous reports.

The first usage of N is side-by-side with BUDGET under each value of DATE. The numbers in these columns tell how many observations there are for each department on each date. This is the number of observations that were used to calculate BUDGET for that date.

The second usage of N is as the last column of the report. The numbers in this column for N tell how many observations each row of the report represents.

Combining Report 1 and Report 2						1
DEPT	DATE					
	31MAR92		30JUN92			
	BUDGET	N	BUDGET	N	N	
Equipment	\$109,500.00	6	\$97,500.00	6	12	
Facilities	\$31,750.00	3	\$30,250.00	3	6	
Other	\$46,500.00	3	\$54,500.00	3	6	
Staff	\$170,000.00	2	\$225,000.00	2	4	
Travel	\$4,300.00	2	\$5,700.00	2	4	

Sample Program 3

```
proc report nowindows data=budget2;
  column dept date, (budget n) n;

  define dept / group;
  define date / across order=internal '_DATE_';
  define budget / analysis sum;

  title 'Combining Report 1 and Report 2';
run;
```

Program Notes

- To clarify the report layout, this program extends the column header for DATE. See “Extending a Column Header to Fill a Column” in Chapter 4.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Using Variable Values as Column Headers in Chapter 1
- Extending a Column Header to Fill a Column in Chapter 4

Chapter 3 Basic Report Enhancements

<i>Introduction</i>	33
<i>Adding Titles</i>	34
<i>Adding Footnotes</i>	36
<i>Changing the Linesize</i>	38
<i>Changing the Pagesize</i>	40
<i>Selecting Observations to Process</i>	42
<i>Controlling Space between Columns</i>	44
<i>Specifying Formats</i>	46
<i>Specifying Column Widths</i>	48
<i>Suppressing the Display of an Item</i>	50
<i>Wrapping the Value of a Character Variable across Multiple Rows</i>	52
<i>Presenting One Variable in Two Different Ways</i>	54
<i>Storing a Report Definition</i>	56
<i>Using a Stored Report Definition</i>	59

Introduction

This chapter shows you how to enhance your report in simple but effective ways. Some of the tasks use parts of the SAS System that you may already be familiar with; others use features specific to the REPORT procedure.

Adding Titles

Implementation

- Use TITLE statements.

TITLE statements work with the REPORT procedure as they do with the rest of the SAS System. TITLE statements are global. You can use them anywhere in your SAS program. See Chapter 9, “SAS Language Statements,” in *SAS Language: Reference*.

Report

Financial Information Grouped by Department				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  title 'Financial Information Grouped';
  title2 'by Department';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Adding Footnotes

Implementation

- Use FOOTNOTE statements.

FOOTNOTE statements work with the REPORT procedure as they do with the rest of the SAS System. FOOTNOTE statements are global. You can use them anywhere in your SAS program.

Report

Financial Information Grouped by Department				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

All dollar amounts are in Canadian dollars.

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  title 'Financial Information Grouped';
  title2 'by Department';
  footnote 'All dollar amounts are in Canadian dollars.';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Changing the Linesize

Implementation

- Use the LS= option on the PROC REPORT statement.

The LS= option overrides the value of the SAS system option LINESIZE= for the duration of the PROC REPORT step. The value for LS= can be any integer from 64 to 256.

Note: Not all devices support 256 columns.

Report

This report uses a linesize of 138 so that all the information for each date can fit on one line. The linesize allows room for the width of each column and the space between columns.

Changing the Linesize											1
DATE	Equipment		Facilities		DEPT Other		Staff		Travel		
	BUDGET	ACTUAL	BUDGET	ACTUAL	BUDGET	ACTUAL	BUDGET	ACTUAL	BUDGET	ACTUAL	
31MAR92	\$109,500.00	\$114,995.48	\$31,750.00	\$30,439.84	\$46,500.00	\$48,014.21	\$170,000.00	\$171,492.80	\$4,300.00	\$3,582.41	
30JUN92	\$97,500.00	\$93,433.73	\$30,250.00	\$30,187.29	\$54,500.00	\$60,626.23	\$225,000.00	\$222,364.71	\$5,700.00	\$4,874.58	

Sample Program

```
proc report data=budget2 nowindows ls=138;  
  column date dept,(budget actual);  
  
  define date / group order=internal;  
  define dept / across;  
  define budget / analysis sum;  
  define actual / analysis sum;  
  
  title 'Changing the Linesize';  
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Using Variable Values as Column Headers in Chapter 1

Changing the Pagesize

Implementation

- Use the PS= option on the PROC REPORT Statement.

The PS= option overrides the value of the SAS system option PAGESIZE= for the duration of the PROC REPORT step. The value for PS= can be any integer from 15 to 32,767.

Report

This report uses a pagesize of 20.

Changing the Pagesize					1
DATE	DEPT	ACCOUNT	BUDGET	ACTUAL	
31MAR92	Staff	fulltime	\$130,000.00	\$127,642.68	
30JUN92	Staff	fulltime	\$165,000.00	\$166,345.75	
31MAR92	Staff	parttime	\$40,000.00	\$43,850.12	
30JUN92	Staff	parttime	\$60,000.00	\$56,018.96	
31MAR92	Equipment	lease	\$40,000.00	\$40,000.00	
30JUN92	Equipment	lease	\$40,000.00	\$40,000.00	
31MAR92	Equipment	purchase	\$40,000.00	\$48,282.38	
30JUN92	Equipment	purchase	\$20,000.00	\$17,769.15	
31MAR92	Equipment	tape	\$8,000.00	\$6,829.42	
30JUN92	Equipment	tape	\$12,000.00	\$11,426.73	
31MAR92	Equipment	sets	\$7,500.00	\$8,342.68	
30JUN92	Equipment	sets	\$7,500.00	\$8,079.62	
31MAR92	Equipment	maint	\$10,000.00	\$7,542.13	
30JUN92	Equipment	maint	\$12,000.00	\$10,675.29	
31MAR92	Equipment	rental	\$4,000.00	\$3,998.87	
30JUN92	Equipment	rental	\$6,000.00	\$5,482.94	
31MAR92	Facilities	rent	\$24,000.00	\$24,000.00	

Changing the Pagesize					2
DATE	DEPT	ACCOUNT	BUDGET	ACTUAL	
30JUN92	Facilities	rent	\$24,000.00	\$24,000.00	
31MAR92	Facilities	utils	\$5,000.00	\$4,223.29	
30JUN92	Facilities	utils	\$3,500.00	\$3,444.81	
31MAR92	Facilities	supplies	\$2,750.00	\$2,216.55	
30JUN92	Facilities	supplies	\$2,750.00	\$2,742.48	
31MAR92	Travel	leases	\$3,500.00	\$3,045.15	
30JUN92	Travel	leases	\$4,500.00	\$3,889.65	
31MAR92	Travel	gas	\$800.00	\$537.26	
30JUN92	Travel	gas	\$1,200.00	\$984.93	
31MAR92	Other	advert	\$30,000.00	\$32,476.98	
30JUN92	Other	advert	\$30,000.00	\$37,325.64	
31MAR92	Other	talent	\$13,500.00	\$12,986.73	
30JUN92	Other	talent	\$19,500.00	\$18,424.64	
31MAR92	Other	musicfee	\$3,000.00	\$2,550.50	
30JUN92	Other	musicfee	\$5,000.00	\$4,875.95	

Sample Program

```
proc report data=budget2 nowindows ps=20;  
  title 'Changing the Pagesize';  
run;
```

Building Blocks

- Listing All Observations and All Variables in Chapter 1

Selecting Observations to Process

Implementation

- Use the WHERE statement.
- Use the WHERE= data set option.

The WHERE statement and the WHERE= data set option work with the REPORT procedure as they do with the rest of the SAS System. See Chapter 9, “SAS Language Statements,” in *SAS Language: Reference*.

Report

This report includes only those observations in which an account overspent its budget.

Selecting Observations to Process					1
DATE	DEPT	ACCOUNT	BUDGET	ACTUAL	
30JUN92	Staff	fulltime	\$165,000.00	\$166,345.75	
31MAR92	Staff	parttime	\$40,000.00	\$43,850.12	
31MAR92	Equipment	purchase	\$40,000.00	\$48,282.38	
31MAR92	Equipment	sets	\$7,500.00	\$8,342.68	
30JUN92	Equipment	sets	\$7,500.00	\$8,079.62	
31MAR92	Other	advert	\$30,000.00	\$32,476.98	
30JUN92	Other	advert	\$30,000.00	\$37,325.64	

Sample Programs

These two PROC REPORT steps produce the same report.

```
proc report data=budget2 nowindows;  
  where budget<actual;
```

```
  title 'Selecting Observations to Process';  
run;
```

```
proc report data=budget2((where=(budget<actual))) nowindows;  
  title 'Selecting Observations to Process';  
run;
```

Building Blocks

- Listing All Observations and All Variables in Chapter 1

Controlling Space between Columns

Implementation

- Use the SPACING= option in the PROC REPORT statement to specify the default number of blank characters between all columns.
- Use the SPACING= option in a DEFINE statement to override the default and to specify the number of blank characters to the left of a particular column.

The minimum value for SPACING= is 0; the default is 2. The maximum depends on the report: for each column in the report, the sum of its width and the blank characters to its left cannot exceed the linesize.

Report 1: Specifying Spacing for All Columns

This report has ten blank characters between columns.

Controlling Space between Columns				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
Staff	talent	\$33,000.00	\$31,411.37	
	fulltime	\$295,000.00	\$293,988.43	
Travel	parttime	\$100,000.00	\$99,869.08	
	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program1

```
proc report data=budget2 nowindows spacing=10;
  column dept account budget actual;

  define dept / group;
  define account / group;
  define budget / analysis sum;
  define actual / analysis sum;

  title 'Controlling Space between Columns';
run;
```

Report 2: Specifying Spacing for One Column

This report uses ten blank characters between all columns except the columns containing BUDGET and ACTUAL. These columns are separated by two blank characters.

Controlling Space between Columns				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program 2

```
proc report data=budget2 nowindows spacing=10;
  column dept account budget actual;

  define dept / group;
  define account / group;
  define budget / analysis sum;
  define actual / analysis sum spacing=2;

  title 'Controlling Space between Columns';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Specifying Formats

Implementation

- Use the FORMAT statement.
Note: This method works only for data set variables.
- Use the FORMAT= option in the DEFINE statement.
Note: You must use this method for computed variables and statistics.

FORMAT statements work with the REPORT procedure as they do with the rest of the SAS System. See Chapter 9, “SAS Language Statements,” in *SAS Language: Reference*.

The FORMAT= option assigns a SAS or user-defined format to the item.

When formatting an item, PROC REPORT searches for the format to use in these places and in this order:

1. in the FORMAT= option in the DEFINE statement
2. in the FORMAT statement
3. in the data set.

PROC REPORT uses the first format that it finds. If it finds no format, the procedure uses the BEST9. format for numeric variables and the \$w. format for character variables.

Report

This report uses the FORMAT statement to display BUDGET with the DOLLAR11. format instead of the DOLLAR11.2 format that is stored with the variable in the data set. It displays ACTUAL with the DOLLAR11.2 format, which is stored with it in the data set.

Specifying Formats				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000	\$80,000.00	
	maint	\$22,000	\$18,217.42	
	purchase	\$60,000	\$66,051.53	
	rental	\$10,000	\$9,481.81	
	sets	\$15,000	\$16,422.30	
	tape	\$20,000	\$18,256.15	
Facilities	rent	\$48,000	\$48,000.00	
	supplies	\$5,500	\$4,959.03	
	utils	\$8,500	\$7,668.10	
Other	advert	\$60,000	\$69,802.62	
	musicfee	\$8,000	\$7,426.45	
	talent	\$33,000	\$31,411.37	
Staff	fulltime	\$295,000	\$293,988.43	
	parttime	\$100,000	\$99,869.08	
Travel	gas	\$2,000	\$1,522.19	
	leases	\$8,000	\$6,934.80	

Sample Programs

These two PROC REPORT steps produce the same report.

```
proc report data=budget2 nowindows;
  format budget dollar11.;

  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  title 'Specifying Formats';
run;
```

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum format=dollar11.;
  define actual  / analysis sum;

  title 'Specifying Formats';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Specifying Column Widths

Implementation

- Use the WIDTH= option in one or more DEFINE statements.

Note: The WIDTH= option controls the width of a column. Formats control the display of values within columns. If you specify a column width that is not large enough for the report item's format, PROC REPORT writes a note to the SAS log and formats the value as closely as it can to the specified format (see "SAS Formats" in Chapter 3 of *SAS Language: Reference*).

Report

This report uses column widths large enough to accommodate each column header on one line.

Specifying Column Widths				1
Department Name	Account	Amount in Budget	Amount Spent	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchas	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplie	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfe	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltim	\$295,000.00	\$293,988.43	
	parttim	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group 'Department Name' width=15;
  define account / group 'Account' width=7;
  define budget  / analysis sum 'Amount in Budget' width=16;
  define actual  / analysis sum 'Amount Spent' width=12;

  title 'Specifying Column Widths';
run;
```

Program Notes

- The quoted strings in the DEFINE statements specify column headers.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Customizing the Text in Column Headers in Chapter 4

Suppressing the Display of an Item

Implementation

- Include the item in the COLUMN statement.
- Use the NOPRINT option in the DEFINE statement.

You may want to display a column that is computed from a report item that does not appear in the report. Placing an item in the COLUMN statement makes it available to you during the PROC REPORT step. When you specify the NOPRINT option, PROC REPORT uses the item as you define it in the DEFINE statement, but does not display it in the report. (Many options for the DEFINE statement, like the FORMAT= option, affect the appearance of the item in the report. You do not need them when you use the NOPRINT option.)

Report

This report uses BUDGET and ACTUAL to calculate BALANCE, but neither BUDGET nor ACTUAL appears in the report.

Suppressing the Display of an Item			1
DEPT	ACCOUNT	BALANCE	
Equipment	lease	\$0.00	
	maint	\$3,782.58	
	purchase	\$-6,051.53	
	rental	\$518.19	
	sets	\$-1,422.30	
	tape	\$1,743.85	
Facilities	rent	\$0.00	
	supplies	\$540.97	
	utils	\$831.90	
Other	advert	\$-9,802.62	
	musicfee	\$573.55	
	talent	\$1,588.63	
Staff	fulltime	\$1,011.57	
	parttime	\$130.92	
Travel	gas	\$477.81	
	leases	\$1,065.20	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum noprnt;
  define actual  / analysis sum noprnt;
  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
  endcomp;

  title 'Suppressing the Display of an Item';
run;
```

Program Notes

- BUDGET and ACTUAL are analysis variables used to calculate the SUM statistic. BALANCE is computed from the values of these statistics. Therefore, the assignment statement that calculates BALANCE uses the compound names BUDGET.SUM and ACTUAL.SUM. For information on adding variables that are not in the data set to the report, see Chapter 5.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Adding a Variable Computed from a Statistic in Chapter 5
- Specifying Formats

Wrapping the Value of a Character Variable across Multiple Rows

Implementation

- Use the FLOW option in the DEFINE statement.

If you do not use the FLOW option in the DEFINE statement for a character variable, PROC REPORT displays only as much of the value as fits in one line of the column for that variable. (Control the width of the column with the WIDTH= option in the DEFINE statement.) If you use the FLOW option, PROC REPORT uses as many lines as necessary to display the entire value within the appropriate column.

Report

This report wraps the value of COMMENT over as many lines as necessary. The value of COMMENT is either a blank or a 41-character string. The width of the column for COMMENT is 20.

Note: If you do not use the FLOW option, only the first 20 characters of COMMENT appear in the report.

Wrapping the Value of a Character Variable across Multiple Rows				1
DEPT	BUDGET	ACTUAL	COMMENT	
Equipment	\$207,000.00	\$208,429.21	This department has overdrawn its budget.	
Facilities	\$62,000.00	\$60,627.13		
Other	\$101,000.00	\$108,640.44	This department has overdrawn its budget.	
Staff	\$395,000.00	\$393,857.51		
Travel	\$10,000.00	\$8,456.99		

Sample Program

```
proc report data=budget2 nowindows;
  column dept budget actual comment;

  define dept    / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define comment / computed width=20 flow;

  compute comment / char length=41;
    if budget.sum-actual.sum < 0 then
      comment='This department has overdrawn its budget.';
    else comment=' ';
  endcomp;

  title 'Wrapping the Value of a Character Variable';
  title2 'across Multiple Rows';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Specifying Column Widths
- Adding a Character Variable in Chapter 5

Presenting One Variable in Two Different Ways

Implementation

- Make an entry for each use of the variable in the COLUMN statement. Use an alias so that you can have a distinct DEFINE statement for each occurrence of the variable. An alias has the form

name=alias

where *name* is the name of the item and *alias* is the alias to use in the DEFINE statement for that column.

- Write a DEFINE statement for each alias. Specify a different definition for each one.

Report

This report has two columns for the group variable DEPT. The definition for the first column uses

- the \$DEPT. format
- a customized column header, CODE.

The definition for the second column uses the default format and column header.

Presenting One Variable in Two Different Ways			1
CODE	DEPT		BUDGET
1000	Equipment		\$207,000.00
2000	Facilities		\$62,000.00
3000	Staff		\$395,000.00
4000	Travel		\$10,000.00
9999	Other		\$101,000.00

Sample Program

```
proc format;
  value $dept 'Equipment'='1000'
             'Facilities'='2000'
             'Other'='9999'
             'Staff'='3000'
             'Travel'='4000';
run;

proc report data=budget2 nowindows;
  column dept=deptcode dept budget;

  define deptcode / group format=$dept. 'CODE';
  define dept / group;
  define budget / analysis sum;

  title 'Presenting One Variable in Two Different Ways';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Specifying Formats
- Customizing the Text in Column Headers in Chapter 4

Storing a Report Definition

Implementation

- Use the OUTREPT= option in the PROC REPORT statement to save the report definition in a catalog entry.
- Use the LIST option in the PROC REPORT statement to write the report definition to the SAS log. Since you cannot read a catalog entry directly, this is the only way you can see the report definition.

The OUTREPT= option stores a report definition in the catalog entry you specify. Specify a three-level name (libref, catalog name, and entry name, separated by periods) as an argument to the option. If the catalog or entry does not exist, PROC REPORT creates it. The libref you use must already be defined. For information on defining a libref see the documentation for the LIBNAME statement in Chapter 9 “SAS Language Statements” in *SAS Language: Reference*.

The stored report definition may differ in these ways from the statements that you submit:

- It shows some defaults that you may not have specified.
- It omits some statements that are not specific to the REPORT procedure, whether you submit them with the PROC REPORT step or whether they are already in effect when you submit the step. These statements include
 - BY
 - FOOTNOTE
 - FREQ
 - TITLE
 - WEIGHT
 - WHERE
- It does not include these PROC REPORT statement options:
 - DATA=
 - LIST
 - OUTREPT=
 - PROFILE=
 - REPORT=
 - WINDOWS|NOWINDOWS
- It does not include SAS system options.

For information on using a report definition after you have stored it, see “Using a Stored Report Definition” later in this chapter.

Report and SAS Log

To create this report, PROC REPORT processes the TITLE, WHERE, LABEL, and FORMAT statements in the PROC REPORT step. (Compare this report to the one in “Using a Stored Report Definition” later in this chapter. In that report, the WHERE and TITLE statements have no effect.)

The SAS log shows the results of storing a report definition while using the LIST option.

Storing a Report Definition			
Department	Account	Amount Budgeted	Amount Spent

Equipment	lease	80,000.00	80,000.00
	maint	22,000.00	18,217.42
	purchase	60,000.00	66,051.53
	rental	10,000.00	9,481.81
	sets	15,000.00	16,422.30
	tape	20,000.00	18,256.15
Facilities	rent	48,000.00	48,000.00
	supplies	5,500.00	4,959.03
	utils	8,500.00	7,668.10

1

```

PROC REPORT LS=72 PS=60 SPLIT="/" HEADLINE HEADSKIP CENTER ;
COLUMN ( DEPT ACCOUNT BUDGET ACTUAL );

DEFINE DEPT / GROUP FORMAT= $10. WIDTH=10 SPACING=2 LEFT
"Department" ;
DEFINE ACCOUNT / GROUP FORMAT= $8. WIDTH=8 SPACING=2 LEFT
"Account" ;
DEFINE BUDGET / SUM FORMAT= COMMA11.2 WIDTH=11 SPACING=2 RIGHT
"Amount Budgeted" ;
DEFINE ACTUAL / SUM FORMAT= COMMA11.2 WIDTH=11 SPACING=2 RIGHT
"Amount Spent" ;

WARNING: A new catalog SASUSER.REPORTS has been created.
NOTE: Definition stored in SASUSER.REPORTS.OUTREPT.
7          The SAS System

NOTE: The PROCEDURE REPORT printed page 1.
NOTE: The PROCEDURE REPORT used 0:00:01.0 real 0:00:00.0 cpu.

```

Sample Program

```
proc report data=budget2 nowindows headline headskip
  outrept=sasuser.reports.outrept list;

  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  title 'Storing a Report Definition';
  where dept='Equipment' or dept='Facilities';
  label dept='Department'
        account='Account'
        budget='Amount Budgeted'
        actual='Amount Spent';
  format budget actual comma11.2;
run;
```

Program Notes

- The HEADLINE and HEADSKIP options in the PROC REPORT statement underline and place a blank line beneath the column headers.

Building Blocks

- Consolidating Multiple Observations into One Row in Chapter 1

Using a Stored Report Definition

Implementation

- Specify the report definition to use as an argument to the REPORT= option in the PROC REPORT statement.

Specify the three-level name of the report definition you want to use as an argument to the option. A three-level name consists of a libref, a catalog name, and an entry name, separated by periods. For information on defining a libref see the documentation for the LIBNAME statement in Chapter 9 “SAS Language Statements” in *SAS Language: Reference*.

You can apply a report definition to any SAS data set that contains the variables used in the report definition. You can use report definitions created in the windowing environment in the nonwindowing environment, and you can use report definitions created in the nonwindowing environment in the windowing environment.

Note: If you use the REPORT= option, you cannot use the COLUMN statement.

Report

This report is created from the report definition SASUSER.REPORTS.OUTREPT. This definition is created in “Storing a Report Definition” earlier in this chapter.

Using a Stored Report Definition				1
These TITLE Statements Are Not Part of the Report Definition				
Department	Account	Amount Budgeted	Amount Spent	
Equipment	lease	80,000.00	80,000.00	
	maint	22,000.00	18,217.42	
	purchase	60,000.00	66,051.53	
	rental	10,000.00	9,481.81	
	sets	15,000.00	16,422.30	
	tape	20,000.00	18,256.15	
Facilities	rent	48,000.00	48,000.00	
	supplies	5,500.00	4,959.03	
	utils	8,500.00	7,668.10	
Other	advert	60,000.00	69,802.62	
	musicfee	8,000.00	7,426.45	
	talent	33,000.00	31,411.37	
Staff	fulltime	295,000.00	293,988.43	
	parttime	100,000.00	99,869.08	
Travel	gas	2,000.00	1,522.19	
	leases	8,000.00	6,934.80	

Sample Program

```
proc report data=budget2 report=sasuser.reports.outrept nowindows;
  title 'Using a Stored Report Definition';
  title3 'These TITLE Statements Are Not Part of the Report Definition';
run;
```

Building Blocks

- Consolidating Multiple Observations into One Row in Chapter 1
- Storing a Report Definition

Chapter 4 Customizing Column Headers

Introduction 61

Customizing the Text in Column Headers 62

Controlling Line Breaks in Column Headers 64

Extending a Column Header to Fill a Column 66

Adding a Blank Line beneath Column Headers 70

Underlining Column Headers 71

Spanning Multiple Columns with One Customized Column Header 73

Introduction

By default, PROC REPORT uses a variable's label as its column header and the name of the statistic as the column header for a statistic. If the label does not fit on one line, PROC REPORT uses multiple lines. By default, the procedure splits the text at a blank character, if possible.

If a variable has no label, PROC REPORT uses the variable's name. If you want to use names when labels exist, submit the following SAS statement before invoking the REPORT procedure:

```
options nolabel;
```

This chapter explains how to customize column headers.

Customizing the Text in Column Headers

Implementation

- Place the text between single or double quotation marks in the DEFINE statement for the item in that column.

You can also use the LABEL statement to specify column headers for data set variables. For information on the LABEL statement see Chapter 9, “SAS Language Statements” in *SAS Language: Reference*.

Report

In this report, the line breaks in the column headers are produced by default. For information on controlling the placement of these line breaks see “Controlling Line Breaks in Column Headers” later in this chapter.

Customizing the Text in Column Headers				1
Department	Account	Amount in Budget	Amount Spent	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
Facilities	tape	\$20,000.00	\$18,256.15	
	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
Other	utils	\$8,500.00	\$7,668.10	
	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
Staff	talent	\$33,000.00	\$31,411.37	
	fulltime	\$295,000.00	\$293,988.43	
Travel	parttime	\$100,000.00	\$99,869.08	
	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group 'Department';
  define account / group 'Account';
  define budget  / analysis sum 'Amount in Budget';
  define actual  / analysis sum 'Amount Spent';

  title 'Customizing the Text in Column Headers';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Controlling Line Breaks in Column Headers

Implementation

- Put each line of text in a separate set of quotation marks in the DEFINE statement for the item in that column.
- Use the split character.

If you use multiple sets of quotation marks, each set defines a separate line of the header.

If you use the split character, the REPORT procedure breaks the header when it reaches the split character and continues the header on the next line. The split character itself is not part of the column header. By default, PROC REPORT uses the slash (/) as the split character. Define an alternate split character with the SPLIT= option in the PROC REPORT statement.

Report 1: Using Quotation Marks

Using Quotation Marks			1
Department	Account	Amount in Budget	Amount Spent
Equipment	lease	\$80,000.00	\$80,000.00
	maint	\$22,000.00	\$18,217.42
	purchase	\$60,000.00	\$66,051.53
	rental	\$10,000.00	\$9,481.81
	sets	\$15,000.00	\$16,422.30
	tape	\$20,000.00	\$18,256.15
Facilities	rent	\$48,000.00	\$48,000.00
	supplies	\$5,500.00	\$4,959.03
	utils	\$8,500.00	\$7,668.10
Other	advert	\$60,000.00	\$69,802.62
	musicfee	\$8,000.00	\$7,426.45
	talent	\$33,000.00	\$31,411.37
Staff	fulltime	\$295,000.00	\$293,988.43
	parttime	\$100,000.00	\$99,869.08
Travel	gas	\$2,000.00	\$1,522.19
	leases	\$8,000.00	\$6,934.80

Sample Program 1

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group 'Department';
  define account / group 'Account';
  define budget  / analysis sum 'Amount' 'in' 'Budget';
  define actual  / analysis sum 'Amount Spent';

  title 'Using Quotation Marks';
run;
```

Program Notes

- The line break in the column header **Amount Spent** is produced by default.

Report 2: Using the Split Character

Using the Split Character				1
Department	Account	Amount in Budget	Amount Spent	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program 2

```
proc report data=budget2 nowindows split='*';
  column dept account budget actual;

  define dept    / group 'Department';
  define account / group 'Account';
  define budget  / analysis sum 'Amount*in*Budget';
  define actual  / analysis sum 'Amount Spent';

  title 'Using the Split Character';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Customizing the Text in Column Headers

Extending a Column Header to Fill a Column

Implementation

- Use special characters as the first and last characters of the header.

If the first and last characters of a header are one of these characters, PROC REPORT uses that character to expand the header to fill the space over the column:

- = _ . * +

If the first character of a header is < and the last character is >, or vice-versa, the REPORT procedure expands the header to fill the space over the column by repeating the first character before the text of the header and the last character after it (see report 3).

Note: When you use variable values as column headers, PROC REPORT creates a column for each formatted value of the variable. The procedure honors special characters in the first and last characters of the format (see report 2).

Report 1: Extending a Column Header with an Underscore Character

This report uses the underscore character in a column header in the DEFINE statement for the variable DATE.

		DATE	
		31MAR92	30JUN92
DEPT	ACCOUNT	BUDGET	BUDGET
Equipment	lease	\$40,000.00	\$40,000.00
	maint	\$10,000.00	\$12,000.00
	purchase	\$40,000.00	\$20,000.00
	rental	\$4,000.00	\$6,000.00
	sets	\$7,500.00	\$7,500.00
	tape	\$8,000.00	\$12,000.00
Facilities	rent	\$24,000.00	\$24,000.00
	supplies	\$2,750.00	\$2,750.00
	utils	\$5,000.00	\$3,500.00
Other	advert	\$30,000.00	\$30,000.00
	musicfee	\$3,000.00	\$5,000.00
	talent	\$13,500.00	\$19,500.00
Staff	fulltime	\$130,000.00	\$165,000.00
	parttime	\$40,000.00	\$60,000.00
Travel	gas	\$800.00	\$1,200.00
	leases	\$3,500.00	\$4,500.00

Sample Program 1

```

proc report data=budget2 nowindows;
  column dept account date,budget;

  define dept    / group;
  define account / group;
  define date    / across order=internal ' _DATE_ ';

  title 'Extending a Column Header with an Underscore Character';
run;

```

Report 2: Extending a Column Header with Special Characters in a Format

This report uses hyphens in the format assigned to DATE.

Extending a Column Header with Special Characters in a Format		DATE		1
DEPT	ACCOUNT	--31MAR92-- BUDGET	--31JUN92-- BUDGET	
Equipment	lease	\$40,000.00	\$40,000.00	
	maint	\$10,000.00	\$12,000.00	
	purchase	\$40,000.00	\$20,000.00	
	rental	\$4,000.00	\$6,000.00	
	sets	\$7,500.00	\$7,500.00	
	tape	\$8,000.00	\$12,000.00	
Facilities	rent	\$24,000.00	\$24,000.00	
	supplies	\$2,750.00	\$2,750.00	
	utils	\$5,000.00	\$3,500.00	
Other	advert	\$30,000.00	\$30,000.00	
	musicfee	\$3,000.00	\$5,000.00	
	talent	\$13,500.00	\$19,500.00	
Staff	fulltime	\$130,000.00	\$165,000.00	
	parttime	\$40,000.00	\$60,000.00	
Travel	gas	\$800.00	\$1,200.00	
	leases	\$3,500.00	\$4,500.00	

Sample Program 2

```

proc format;
  value dateval '31mar92'd='--31MAR92--';
  '30jun92'd='--31JUN92--';
run;

proc report data=budget2 nowindows;
  column dept account date,budget;

  define dept    / group;
  define account / group;
  define date    / across order=internal format=dateval.;

```

Sample Program 3

```
proc report data=budget2 nowindows;
  column dept account date,budget;

  define dept    / group;
  define account / group;
  define date    / across order=internal '>DATE<';

  title 'Extending a Column Header with Inequality Signs';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Using Variable Values as Column Headers in Chapter 1
- Specifying Formats in Chapter 3

Adding a Blank Line beneath Column Headers

Implementation

- Use the HEADSKIP option in the PROC REPORT statement.

Report

Adding a Blank Line beneath Column Headers				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
Facilities	tape	\$20,000.00	\$18,256.15	
	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
Other	utils	\$8,500.00	\$7,668.10	
	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows headskip;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  title 'Adding a Blank Line beneath Column Headers';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Underlining Column Headers

Implementation

- Use the HEADLINE option in the PROC REPORT statement.
- Customize the text of the headers to include underlining characters.

The HEADLINE option underlines all column headers and the spaces between columns. Using underlining characters in the individual column headers underlines the headers without underlining the spaces between columns.

Report 1: Underlining Column Headers and Spaces between Columns

Underlining Column Headers and Spaces between Columns				1
DEPT	ACCOUNT	BUDGET	ACTUAL	

Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
Facilities	tape	\$20,000.00	\$18,256.15	
	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
Other	utils	\$8,500.00	\$7,668.10	
	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
Staff	talent	\$33,000.00	\$31,411.37	
	fulltime	\$295,000.00	\$293,988.43	
Travel	parttime	\$100,000.00	\$99,869.08	
	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program 1

```
proc report data=budget2 nowindows headline;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  title 'Underlining Column Headers and Spaces between Columns';
run;
```

Report 2: Underlining Column Headers without Underlining the Spaces between Columns

Underlining Column Headers without Underlining the Spaces between Columns				1
Department	Account	Budget	Actual	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program 2

```
proc report data=budget2 nowindows;
  column dept account budget actual;
  define dept / group 'Department' -----;
  define account / group 'Account' -----;
  define budget / analysis sum 'Budget' -----;
  define actual / analysis sum 'Actual' -----;

  title 'Underlining Column Headers without Underlining';
  title2 'the Spaces between Columns';
run;
```

Note: You need to use only two underlining characters to underline the entire column header. For an explanation see “Extending a Column Header to Fill a Column” earlier in this Chapter.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Customizing the Text in Column Headers
- Controlling Line Breaks in Column Headers
- Extending a Column Header to Fill a Column

Spanning Multiple Columns with One Customized Column Header

Implementation

- Place, inside parentheses, in the COLUMN statement
 - the column header (in single or double quotation marks)
 - a list of items to span. Separate the items with blanks.

Report

Spanning Multiple Columns with One Customized Column Header				1
		CANADIAN DOLLARS		
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```

proc report data=budget2 nowindows;
  column dept account ('CANADIAN DOLLARS' budget actual);

  define dept / group;
  define account / group;
  define budget / analysis sum;
  define actual / analysis sum;

  title 'Spanning Multiple Columns with One Customized Column Header';
run;

```

Building Blocks

- *Specifying and Ordering Columns to Display in Chapter 1*
- *Consolidating Multiple Observations into One Row in Chapter 1*

Chapter 5 Adding Variables that Are Not in the Input Data Set

Introduction 75

Features that You Can Use in a Compute Block 76

How to Reference Report Variables in a Compute Block 76

Adding a Variable Computed from a Group, Order, Computed, or Display Variable 78

Adding a Variable Computed from a Statistic 80

Adding a Variable Computed from a Variable that Shares a Column with an Across Variable 82

Adding a Character Variable 85

Introduction

A report can include variables that are not in the input data set. You compute values for such variables with statements and functions inside a *compute block*. A compute block for a computed variable contains (in this order)

1. a COMPUTE statement
2. selected SAS language features, CALL DEFINE statements, or both,
3. an ENDCOMP statement.

A list of SAS language features that you can use between the COMPUTE and ENDCOMP statements appears later in this section.

A COMPUTE statement for a computed variable must include (in this order)

1. the keyword COMPUTE
2. the name of the computed variable.

If the computed variable is a character variable, the COMPUTE statement also includes one or both of these options:

The option ...	specifies ...
CHARACTER	the variable is a character variable. If you specify LENGTH=, you do not need to specify CHARACTER.
LENGTH= <i>length</i>	the length of the character variable. <i>length</i> is an integer from 1 to 200. If you specify CHARACTER but do not specify a length, the default is 8.

Features that You Can Use in a Compute Block

Compute blocks support these SAS language features:

- DM statement
- %INCLUDE statement
- selected DATA step statements:

assignment	LENGTH
CALL	LINK
DO (all forms)	RETURN
END	SELECT
GO TO	sum
IF-THEN/ELSE	
- comments
- null statements
- macro variables
- all DATA step functions.

Compute blocks for a computed variable also support the CALL DEFINE statement. For information on the CALL DEFINE statement see Chapter 9 “The REPORT Language.”

For more information on compute blocks see the documentation for the COMPUTE statement in Chapter 9.

How to Reference Report Variables in a Compute Block

Reference report variables in a compute block in one of three ways:

- by name.
- by a compound name that identifies both the variable and the name of the statistic you calculate with it. A compound name has this form:

variable-name.statistic

- by column number, in the form

Cn

where *n* is the number of the column (from left to right) in the report. When you determine the column number, count columns that are not displayed (see “Suppressing the Display of an Item” in Chapter 3).

The following table shows when to use each type of reference in a compute block.

If the variable you reference is this type ...	then refer to it by ...	For example ...
group	name	DEPT
order	name	DEPT
computed	name	DEPT
display	name	DEPT
display sharing a column with a statistic	a compound name	BUDGET.SUM
analysis	a compound name	BUDGET.MEAN
any type sharing a column with an across variable	column number	_C3_

► **Caution** *Position of Computed Variables*

A computed variable must be calculated from report items that are to its left in the COLUMN statement. Otherwise, the value of the computed variable is always missing. (For an explanation of these missing values see Chapter 10, “How PROC REPORT Builds a Report.”) ▲

► **Caution** *Aliases Are Not Valid in Compute Blocks*

Aliases that you create in a COLUMN statement are valid only in DEFINE statements. You cannot use them in compute blocks. ▲

The reports in this chapter illustrate the different ways of referencing data set variables. They also illustrate the differences between adding a character and numeric variable to a report.

Adding a Variable Computed from a Group, Order, Computed, or Display Variable

Implementation

- Include the new variable in the COLUMN statement.
- Define the variable's usage as COMPUTED in a DEFINE statement.
- Compute the value of the variable in a compute block.

When you base a computed variable on a group, order, or computed variable, or on a display variable with no statistics above or below it, refer to variables that you use in your calculations by their names. For information on using statistics with display variables, see “Placing Statistics above or below a Variable” in Chapter 2.

Report

This report is based on the data set EMPINFO. This data set contains the first and last names of employees as well as information about their salaries, date of hire, and so forth.

In this report FULLNAME is a computed variable. It is computed from the order variables LNAME and FNAME, which do not appear in the report.

Adding a Variable Based on a Group, Order, Computed, or Display Variable					1
FULLNAME	ID	HIRED	SALARY	PHONE	
Helga Jensen	477562122	01FEB81	47400	286-2816	
Marie Kulenic	894724859	24JUN83	41400	493-1472	
Jay Sayre	324987451	15NOV84	44800	933-2998	
Andrew Tolson	596771321	18MAR88	41200	929-4800	
Anna Zweerink	988427431	07JUL85	43700	929-3885	

Note: The values of the order variables LNAME and FNAME determine the order of the rows in the report even though they are not displayed.

Sample Program

This program includes the DATA step that creates EMPINFO.

```

data empinfo;
  input id $ 1-9 lname $ 11-20 fname $ 22-31 @33 hired date7. salary phone $;
  format hired date7.;
  cards;
324987451 Sayre      Jay      15nov84 44800 933-2998
596771321 Tolson    Andrew  18mar88 41200 929-4800
477562122 Jensen    Helga   01feb81 47400 286-2816
894724859 Kulenic    Marie   24jun83 41400 493-1472
988427431 Zweerink  Anna    07jul85 43700 929-3885
;

proc report data=empinfo nowindows;
  column lname fname fullname id hired salary phone;

  define lname      / order noprint;
  define fname      / order noprint;
  define fullname   / computed;
  define id         / display;
  define hired      / display;
  define salary     / analysis sum;
  define phone      / display;

  compute fullname / char length=20;
  fullname=trim(fname)||' '||lname;
endcomp;

  title 'Adding a Variable Based on a Group, Order, ';
  title2 'Computed, or Display Variable';
run;

```

Program Notes

- The assignment statement that computes FULLNAME uses the variables' names.
- Both LNAME and FNAME are in the COLUMN statement so that their values are available for computation.
- The NOPRINT option in the DEFINE statements for LNAME and FNAME suppresses their appearance in the report (see “Suppressing the Display of an Item” in Chapter 3).

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Ordering Rows in Chapter 1
- Suppressing the Display of an Item in Chapter 3

Adding a Variable Computed from a Statistic

You add a computed variable based on a statistic when the computation of the variable uses either

- an analysis variable
- a display variable with statistics above or below it.

Every analysis variable is associated with a statistic. You make this association in one of two ways:

- with an option in the DEFINE statement for the analysis variable
- by placing a statistic above or below the analysis variable.

For information on using analysis variables, see “Consolidating Multiple Observations into One Row” in Chapter 1. For information on using statistics above or below other variables, see “Placing Statistics above or below a Variable” in Chapter 2.

Implementation

- Include the new variable in the COLUMN statement.
- Define the variable as COMPUTED in a DEFINE statement.
- Compute the value of the variable in a compute block.

When you base a computed variable on a statistic, refer to the statistic by a compound name (see the introduction to this chapter).

Report

In this report BALANCE is a computed variable. It is computed from the analysis variables BUDGET and ACTUAL.

Adding a Variable Based on a Statistic					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
  endcomp;

  title 'Adding a Variable Based on a Statistic';
run;
```

Program Notes

- The assignment statement that computes BALANCE uses the compound names BUDGET.SUM and ACTUAL.SUM.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Specifying Formats in Chapter 3

Adding a Variable Computed from a Variable that Shares a Column with an Across Variable

Implementation

- Include the new variable in the COLUMN statement.
- Define the variable as COMPUTED in a DEFINE statement.
- Compute the value of the variable in a compute block.

When you base a computed variable on variables that share a column with an across variable, refer to variables that you use in your computations by column number (see the introduction to this chapter). If the computed variable also shares a column with an across variable, refer to the computed variable by column number.

Report 1: When the Computed Variable Does Not Share a Column with an Across Variable

In this report RATIO is a computed variable. It is the ratio of the budget for the first quarter to the budget for the second quarter. It is computed for each account from the the values of BUDGET for the first and second quarters. BUDGET shares a column with DATE.

When the Computed Variable Does Not Share the Column					1
		DATE			
DEPT	ACCOUNT	31MAR92 BUDGET	30JUN92 BUDGET	RATIO	
Equipment	lease	\$40,000.00	\$40,000.00	1.00	
	maint	\$10,000.00	\$12,000.00	0.83	
	purchase	\$40,000.00	\$20,000.00	2.00	
	rental	\$4,000.00	\$6,000.00	0.67	
	sets	\$7,500.00	\$7,500.00	1.00	
	tape	\$8,000.00	\$12,000.00	0.67	
Facilities	rent	\$24,000.00	\$24,000.00	1.00	
	supplies	\$2,750.00	\$2,750.00	1.00	
	utils	\$5,000.00	\$3,500.00	1.43	
Other	advert	\$30,000.00	\$30,000.00	1.00	
	musicfee	\$3,000.00	\$5,000.00	0.60	
	talent	\$13,500.00	\$19,500.00	0.69	
Staff	fulltime	\$130,000.00	\$165,000.00	0.79	
	parttime	\$40,000.00	\$60,000.00	0.67	
Travel	gas	\$800.00	\$1,200.00	0.67	
	leases	\$3,500.00	\$4,500.00	0.78	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account date,budget ratio;

  define dept    / group;
  define account / group;
  define date    / across order=internal;
  define budget  / analysis sum;
  define ratio   / computed format=5.2;

  compute ratio;
    ratio=_c3_/_c4_;
  endcomp;

  title 'When the Computed Variable Does Not Share the Column';
run;
```

Program Notes

- Because BUDGET shares a column with the across variable DATE, the program references BUDGET by column number. The value of BUDGET for the first quarter is in column 3; the value for the second quarter is in column 4.
- RATIO does not share a column with an across variable. The program references RATIO by name.

Report 2: When the Computed Variable Does Share a Column

In this report QTRBAL is a computed variable. It is computed from the analysis variables BUDGET and ACTUAL. The variables BUDGET, ACTUAL, and QTRBAL all share a column with DATE.

When the Computed Variable Does Share a Column								1
		DATE						
		31MAR92			30JUN92			
DEPT	ACCOUNT	BUDGET	ACTUAL	QTRBAL	BUDGET	ACTUAL	QTRBAL	
Equipment	lease	\$40,000.00	\$40,000.00	\$0.00	\$40,000.00	\$40,000.00	\$0.00	
	maint	\$10,000.00	\$7,542.13	\$2,457.87	\$12,000.00	\$10,675.29	\$1,324.71	
	purchase	\$40,000.00	\$48,282.38	\$-8,282.38	\$20,000.00	\$17,769.15	\$2,230.85	
	rental	\$4,000.00	\$3,998.87	\$1.13	\$6,000.00	\$5,482.94	\$517.06	
	sets	\$7,500.00	\$8,342.68	\$-842.68	\$7,500.00	\$8,079.62	\$-579.62	
Facilities	tape	\$8,000.00	\$6,829.42	\$1,170.58	\$12,000.00	\$11,426.73	\$573.27	
	rent	\$24,000.00	\$24,000.00	\$0.00	\$24,000.00	\$24,000.00	\$0.00	
	supplies	\$2,750.00	\$2,216.55	\$533.45	\$2,750.00	\$2,742.48	\$7.52	
Other	utils	\$5,000.00	\$4,223.29	\$776.71	\$3,500.00	\$3,444.81	\$55.19	
	advert	\$30,000.00	\$32,476.98	\$-2,476.98	\$30,000.00	\$37,325.64	\$-7,325.64	
	musicfee	\$3,000.00	\$2,550.50	\$449.50	\$5,000.00	\$4,875.95	\$124.05	
Staff	talent	\$13,500.00	\$12,986.73	\$513.27	\$19,500.00	\$18,424.64	\$1,075.36	
	fulltime	\$130,000.00	\$127,642.68	\$2,357.32	\$165,000.00	\$166,345.75	\$-1,345.75	
Travel	parttime	\$40,000.00	\$43,850.12	\$-3,850.12	\$60,000.00	\$56,018.96	\$3,981.04	
	gas	\$800.00	\$537.26	\$262.74	\$1,200.00	\$984.93	\$215.07	
	leases	\$3,500.00	\$3,045.15	\$454.85	\$4,500.00	\$3,889.65	\$610.35	

Sample Program

```
proc report data=budget2 nowindows ls=100;
  column dept account date, (budget actual qtrbal);

  define dept      / group;
  define account  / group;
  define date      / across order=internal;
  define budget   / analysis sum;
  define actual   / analysis sum;
  define qtrbal   / computed format=dollar11.2 width=11;

  compute qtrbal;
    _c5 = _c3 - _c4;
    _c8 = _c6 - _c7;
  endcomp;

  title 'When the Computed Variable Does Share a Column';
run;
```

Program Notes

- Because BUDGET and ACTUAL share a column with the across variable DATE, the program references them by column number. The value of BUDGET for the first quarter is in column 3; the value for the second quarter is in column 6. The value of ACTUAL for the first quarter is in column 4; the value for the second quarter is in column 7.
- QTRBAL also shares the column with DATE. Therefore, the program references QTRBAL by column number. The value of QTRBAL for the first quarter is in column 5; the value for the second quarter is in column 8.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Ordering Rows in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Using Variable Values as Column Headers in Chapter 1
- Changing the Linesize in Chapter 3
- Specifying Formats in Chapter 3

Adding a Character Variable

Implementation

- Include the new variable in the COLUMN statement.
- Define the variable as COMPUTED in a DEFINE statement.
- Compute the value of the variable in a compute block.
- Include the CHARACTER option, the LENGTH= option, or both in the COMPUTE statement (see the introduction to this Chapter).

Report

In this report OVER is a computed character variable that flags accounts that have overspent their budgets. The value of OVER depends on the the difference between the values of the analysis variables BUDGET and ACTUAL.

Adding a Character Variable					1
DEPT	ACCOUNT	BUDGET	ACTUAL	OVER	
Equipment	lease	\$80,000.00	\$80,000.00		
	maint	\$22,000.00	\$18,217.42		
	purchase	\$60,000.00	\$66,051.53	*	
	rental	\$10,000.00	\$9,481.81		
	sets	\$15,000.00	\$16,422.30	*	
Facilities	tape	\$20,000.00	\$18,256.15		
	rent	\$48,000.00	\$48,000.00		
	supplies	\$5,500.00	\$4,959.03		
Other	utils	\$8,500.00	\$7,668.10		
	advert	\$60,000.00	\$69,802.62	*	
	musicfee	\$8,000.00	\$7,426.45		
	talent	\$33,000.00	\$31,411.37		
Staff	fulltime	\$295,000.00	\$293,988.43		
	parttime	\$100,000.00	\$99,869.08		
Travel	gas	\$2,000.00	\$1,522.19		
	leases	\$8,000.00	\$6,934.80		

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual over;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define over    / computed format=$1. width=4;

  compute over / character length=1;
    if budget.sum-actual.sum<0 then over='*';
    else over=' ';
  endcomp;

  title 'Adding a Character Variable';
run;
```

Program Notes

- The assignment statement that computes OVER uses the compound names BUDGET.SUM and ACTUAL.SUM.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Adding a Variable Computed from a Statistic
- Specifying Column Width in Chapter 3

Chapter 6 Working with Groups of Observations

Introduction	87
<i>Variable Values at Breaks between Groups</i>	88
<i>Working with Groups</i>	88
<i>Default Group Summaries</i>	89
<i>Customized Group Summaries</i>	90
<i>Features that You Can Use in a Compute Block</i>	91
<i>How to Reference Report Variables in a Compute Block</i>	92
Producing a Separate Page for Each Group	93
Inserting a Blank Line between Groups	97
Inserting Multiple Blank Lines between Groups	99
Summarizing Variables and Statistics for Groups	101
Overlining Summary Information	103
Underlining Summary Information	105
Suppressing the Value of the Group Variable in the Summary Information	107
Changing the Value of a Character Variable in the Summary Information	109
Specifying Text and Variable Values between Groups	111
Controlling the Amount of Space Used to Write a Character Variable	114
Controlling the Amount of Space Used to Write a Numeric Variable	117
Controlling the Horizontal Position of Text between Groups	120
Initializing Variables for Each Group	122
Placing Conditional Text between Groups	125
Controlling Whether Statements in a Compute Block Execute between Groups	128

Introduction

For some reports, you may want to list all observations in the input data set. For others, you may prefer to consolidate information about a group of observations. You can summarize information about each group of observations. To do so, you create a *break* in the report before or after each group.

Break lines are lines of text (including blanks) that appear at a break. There are two ways to display break lines. The first way is simpler. It produces default group summaries. The second way is more flexible. It produces customized group summaries and provides a way to slightly modify the default group summaries.

A report can contain

- one or more default group summaries
- one or more customized group summaries
- a mixture of default and customized summaries.

With one exception (the example of BY processing in “Producing a Separate Page for

Each Group”), the examples in this chapter all use breaks and groups.

Note: You can also create breaks (and therefore summaries)

- whenever the value of an order variable changes. For information on using order variables see “Ordering Rows” in Chapter 1 and the discussion of the ORDER option in the documentation for the DEFINE statement in Chapter 9.
- at the beginning or end of a report (see Chapter 7).

Variable Values at Breaks between Groups

In a break for a group of observations,

- character variables to the right of the break variable have missing values
- statistics and analysis variables are calculated for the group
- computed variables are calculated by executing the statements in the COMPUTE block attached to that variable, just as they are for any other row of the report (see Chapter 5).

Working with Groups

A group is a set of observations that has a unique combination of values for all group variables. When you group observations

- PROC REPORT tries to consolidate each group into one row of the report. If PROC REPORT cannot create groups, it returns a message and displays group variables as it displays order variables. For more information on groups, see the discussion of the GROUP option and “Interactions of Position and Usage” in the documentation for the DEFINE statement in Chapter 9.
- PROC REPORT, by default, orders the rows of a report according to the formatted values of group variables. You can control the order with the ORDER= and DESCENDING options on the DEFINE statement (see “Ordering Rows” in Chapter 1).
- the values displayed for analysis variables are values of a statistic calculated for all observations in the group. The default statistic for an analysis variable is the SUM statistic. You specify an alternate statistic by naming the statistic you want as an option in the DEFINE statement. For information on using statistics in a report see Chapter 2.

To create groups

- Use the GROUP option in one or more DEFINE statements to designate group variables.
- Be sure that the report does not contain
 - order variables
 - display variables that do not share a column with a statistic (see “Placing Statistics above or below a Variable” in Chapter 2).

For information on order and display variables see the introduction to Chapter 1 and the

documentation for the DEFINE statement in Chapter 9.

Note: Other base SAS procedures implement the concept of groups with the CLASS statement.

Default Group Summaries

Default group summaries can

- visually separate groups
- summarize statistics and calculate computed variables for each group.

The default group summaries in this report summarize the analysis variables BUDGET and ACTUAL. Two rows of hyphens and a blank line separate the summaries from the rest of the report.

DEPT	ACCOUNT	BUDGET	ACTUAL
Equipment	lease	\$80,000.00	\$80,000.00
	maint	\$22,000.00	\$18,217.42
	purchase	\$60,000.00	\$66,051.53
	rental	\$10,000.00	\$9,481.81
	sets	\$15,000.00	\$16,422.30
	tape	\$20,000.00	\$18,256.15

Equipment		\$207,000.00	\$208,429.21

Facilities	rent	\$48,000.00	\$48,000.00
	supplies	\$5,500.00	\$4,959.03
	utils	\$8,500.00	\$7,668.10

Facilities		\$62,000.00	\$60,627.13

Other	advert	\$60,000.00	\$69,802.62
	musicfee	\$8,000.00	\$7,426.45
	talent	\$33,000.00	\$31,411.37

Other		\$101,000.00	\$108,640.44

Staff	fulltime	\$295,000.00	\$293,988.43
	parttime	\$100,000.00	\$99,869.08

Staff		\$395,000.00	\$393,857.51

Travel	gas	\$2,000.00	\$1,522.19
	leases	\$8,000.00	\$6,934.80

Travel		\$10,000.00	\$8,456.99

To create default group summaries, write a BREAK statement. A BREAK statement must include (in this order)

1. the keyword BREAK
2. the BEFORE or AFTER argument
3. the name of a group variable. This variable is the *break variable*. PROC REPORT

creates a break each time the value of this variable changes.

If you want the summaries to appear before the first row of each group, use the BEFORE argument. If you want the summaries to appear after the last row of each group, use the AFTER argument.

The BREAK statement supports options that control the appearance of the default group summaries. You can use any combination of options in the BREAK statement. The order in which you use the options does not affect the report. This chapter illustrates the use of these options.

Customized Group Summaries

When you create customized summaries, you execute SAS language statements (and one statement unique to PROC REPORT, the LINE statement) at the break. These statements display text, calculate values for a group, or both.

The customized group summaries in this report summarize the analysis variables BUDGET and ACTUAL, just as the default report did. However, the customized summaries include additional text.

				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	

The budget for this department was \$207,000.00.				
The department spent \$208,429.21.				

Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	

The budget for this department was \$62,000.00.				
The department spent \$60,627.13.				

Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	

The budget for this department was \$101,000.00.				
The department spent \$108,640.44.				

Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	

The budget for this department was \$395,000.00.				
The department spent \$393,857.51.				

Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

The budget for this department was \$10,000.00.				
The department spent \$8,456.99.				

The statements that create customized group summaries must be in a *compute block*. A compute block for a customized group summary contains (in this order)

1. a COMPUTE statement
2. selected SAS language features, LINE statements, or both,
3. an ENDCOMP statement.

A list of features that you can use between the COMPUTE and ENDCOMP statements appears later in this section.

A COMPUTE statement for a group summary must include (in this order)

1. the keyword COMPUTE
2. the BEFORE or AFTER argument
3. the name of a group variable. This variable is the *break variable*. PROC REPORT creates a break each time the value of this variable changes.

If you want the summaries to appear before the first row of each group, use the BEFORE argument. If you want the summaries to appear after the last row of each group, use the AFTER argument.

Features that You Can Use in a Compute Block

Compute blocks support these SAS language features:

- DM statement
- %INCLUDE statement
- selected DATA step statements:

assignment	LENGTH
CALL	LINK
DO (all forms)	RETURN
END	SELECT
GO TO	sum
IF-THEN/ELSE	

- comments
- null statements
- macro variables
- all DATA step functions.

Compute blocks for a customized summary also support the LINE statement. For information on the LINE statement see Chapter 9, “The REPORT Language.”

For more information on compute blocks see “COMPUTE Statement” in Chapter 9.

How to Reference Report Variables in a Compute Block

Reference report variables in a compute block in one of three ways:

- by name.
- by a compound name that identifies both the variable and the name of the statistic you calculate with it. A compound name has this form:

variable-name.statistic

- by column number, in the form

Cn

where *n* is the number of the column (from left to right) in the report. When you determine the column number, count columns that are not displayed (see “Suppressing the Display of an Item” in Chapter 3).

The following table shows when to use each type of reference in a customized group summary.

If the variable you reference is this type ...	then refer to it by ...	For example ...
group	name	DEPT
computed	name	DEPT
display sharing a column with a statistic	a compound name	BUDGET.SUM
analysis	a compound name	BUDGET.MEAN
any type sharing a column with an across variable	column number	_C3_

Note: You cannot create groups if the report contains any order variables or any display variables without statistics above or below them.

► **Caution Aliases Are Not Valid in Compute Blocks**

Aliases that you create in a COLUMN statement are valid only in DEFINE statements. You cannot use them in compute blocks. ▲

Compute blocks can produce an endless variety of customized summaries. They can also slightly modify a default summary. This chapter gets you started. For more information see “BREAK Statement” and “LINE Statement” in Chapter 9, “The REPORT Language.”

Producing a Separate Page for Each Group

Implementation

You can create a separate page of the report for each group in either of these ways:

- Use the PAGE option in the BREAK statement.
- Use a BY statement in which the BY variable is the group variable for which you want to create separate pages. To use a BY statement, you must first sort or index the data set.

These methods produce nearly identical reports. However, if you use a BY statement, the report contains a BY line at the top of each group. The BY line displays the BY variable and its current value. You can mimic the BY line or create your own version of it by creating a customized summary before the group variable instead of using the BY statement (see “Specifying Text and Variable Values between Groups” later in this chapter).

Report 1: Inserting a Page Break between Groups

In this report DEPT is a group variable. It is also the break variable. PROC REPORT produces a separate page of the report for each value of DEPT.

Inserting a Page Break between Groups				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	

Inserting a Page Break between Groups				2
DEPT	ACCOUNT	BUDGET	ACTUAL	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	

Inserting a Page Break between Groups				3
DEPT	ACCOUNT	BUDGET	ACTUAL	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	

Inserting a Page Break between Groups				4
DEPT	ACCOUNT	BUDGET	ACTUAL	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	

Inserting a Page Break between Groups				5
DEPT	ACCOUNT	BUDGET	ACTUAL	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program 1

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept / group;
  define account / group;
  define budget / analysis sum;
  define actual / analysis sum;

  break after dept / page;

  title 'Inserting a Page Break between Groups';
run;
```

Report 2: Using a Group Variable as a BY Variable

In this report DEPT is a group variable. It is also the BY variable.

Using a Group Variable as a BY Variable				1
----- DEPT=Equipment -----				
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	

Using a Group Variable as a BY Variable				2
----- DEPT=Facilities -----				
DEPT	ACCOUNT	BUDGET	ACTUAL	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	

Using a Group Variable as a BY Variable				3
----- DEPT=Other -----				
DEPT	ACCOUNT	BUDGET	ACTUAL	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	

Using a Group Variable as a BY Variable				4
----- DEPT=Staff -----				
DEPT	ACCOUNT	BUDGET	ACTUAL	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	

Using a Group Variable as a BY Variable				5
----- DEPT=Travel -----				
DEPT	ACCOUNT	BUDGET	ACTUAL	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

In this report the value of DEPT appears in the first column of the report and in the BY line. To eliminate this redundancy, you can either

- remove the DEFINE statement for DEPT and remove DEPT from the COLUMN statement.
- suppress the BY line by executing the following OPTIONS statement before you submit the program:

```
options nobyline;
```

Suppressing the BY line produces a report identical to the report produced by inserting a page break between groups.

Sample Program 2

```
options byline;

proc sort data=budget2 out=sorted;
  by dept;
run;

proc report data=sorted nowindows;
  column dept account budget actual;

  define dept / group;
  define account / group;
  define budget / analysis sum;
  define actual / analysis sum;

  by dept;
  title 'Using a Group Variable as a BY Variable';
run;
```

Note: PROC REPORT fully supports the #BYVAR, #BYVAL, and #BYLINE specifications. These specifications insert BY-Group information into titles. For further information see Chapter 15, “Background for Procedures,” in SAS Technical Report P-222: *Changes and Enhancements to Base SAS Software*.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Inserting a Blank Line between Groups

Implementation

- Use the SKIP option in the BREAK statement.

Report

Inserting a Blank Line between Groups				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  break after dept / skip;

  title 'Inserting a Blank Line between Groups';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Inserting Multiple Blank Lines between Groups

Implementation

Use customized group summaries to insert multiple blank lines between groups.

- Create a compute block for the break variable.
- Use a LINE statement with a blank for an argument for each blank line you want.

The LINE statement writes break lines containing text, values calculated for a group, or both. The LINE statement must be between a COMPUTE and an ENDCOMP statement.

Report

This report has three blank lines after each group.

Inserting Multiple Blank Lines between Groups				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  compute after dept;
    line ' ';
    line ' ';
    line ' ';
  endcomp;

  title 'Inserting Multiple Blank Lines between Groups';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Summarizing Variables and Statistics for Groups

Implementation

- Use the SUMMARIZE option in the BREAK statement.

The SUMMARIZE option writes default *summary lines*. Each default summary line summarizes statistics and calculates computed variables for multiple observations, in this case a group. By default, group summary lines also include the break variable.

Report

In this report BUDGET and ACTUAL are analysis variables used to calculate the SUM statistic. Therefore, the values for BUDGET and ACTUAL in the summary lines are sums for all rows in the group. If you defined these variables as calculating a different statistic (MEAN, for example), the values in the summary line would be the value of that statistic for all rows in the group.

Summarizing Variables and Statistics for Groups				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Equipment		\$207,000.00	\$208,429.21	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Facilities		\$62,000.00	\$60,627.13	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Other		\$101,000.00	\$108,640.44	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Staff		\$395,000.00	\$393,857.51	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
Travel		\$10,000.00	\$8,456.99	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  break after dept / skip summarize;

  title 'Summarizing Variables and Statistics for Groups';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Inserting a Blank Line between Groups

Overlining Summary Information

Implementation

- Use the OL (overline) or DOL (double overline) option in the BREAK statement.

The OL option writes a line of hyphens (-) above each value that appears in the summary lines or that would appear in the summary lines if you specified the SUMMARIZE option.

The DOL option writes a line of equals signs (=) across the report above each value that appears in the summary lines or that would appear in the summary lines if you specified the SUMMARIZE option.

In both cases, the overlining extends for the width of the column even if the value does not fill the column.

Note: If you use both options, PROC REPORT honors only OL.

Report

Overlining Summary Information				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
-----		-----	-----	
Equipment		\$207,000.00	\$208,429.21	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
-----		-----	-----	
Facilities		\$62,000.00	\$60,627.13	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
-----		-----	-----	
Other		\$101,000.00	\$108,640.44	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
-----		-----	-----	
Staff		\$395,000.00	\$393,857.51	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
-----		-----	-----	
Travel		\$10,000.00	\$8,456.99	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  break after dept / skip summarize 01;

  title 'Overlining Summary Information';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Summarizing Variables and Statistics for Groups
- Inserting a Blank Line between Groups of Observations

Underlining Summary Information

Implementation

- Use the UL (underline) or DUL (double underline) option in the BREAK statement.

The UL option writes a line of hyphens (-) below each value that appears in the summary lines or that would appear in the summary lines if you specified the SUMMARIZE option.

The DUL option writes a line of equals signs (=) below each value that appears in the summary lines or that would appear in the summary lines if you specified the SUMMARIZE option.

In both cases, the underlining extends for the width of the column even if the value does not fill the column.

Note: If you use both options, PROC REPORT honors only UL.

Report

Underlining Summary Information				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Equipment		\$207,000.00	\$208,429.21	
-----		-----	-----	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Facilities		\$62,000.00	\$60,627.13	
-----		-----	-----	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Other		\$101,000.00	\$108,640.44	
-----		-----	-----	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Staff		\$395,000.00	\$393,857.51	
-----		-----	-----	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
Travel		\$10,000.00	\$8,456.99	
-----		-----	-----	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  break after dept / skip summarize u;

  title 'Underlining Summary Information';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Summarizing Variables and Statistics for Groups
- Inserting a Blank Line between Groups

Suppressing the Value of the Group Variable in the Summary Information

Implementation

- Use the SUPPRESS option in the BREAK statement.

Report

In this report the value of the break variable, DEPT, is not in the group summary lines. (Compare this report to the one in “Summarizing Variables and Statistics for Groups” earlier in this chapter.)

Suppressing the Value of the Group Variable in the Summary Information				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
			\$207,000.00	\$208,429.21
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
			\$62,000.00	\$60,627.13
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
			\$101,000.00	\$108,640.44
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
			\$395,000.00	\$393,857.51
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
			\$10,000.00	\$8,456.99

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  break after dept / skip summarize suppress;

  title 'Suppressing the Value of the Group Variable ';
  title2 'in the Summary Information';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Summarizing Variables and Statistics for Groups
- Inserting a Blank Line between Groups

Changing the Value of a Character Variable in the Summary Information

Implementation

You may want to put text in the default summary line for emphasis.

- Create a compute block for the break variable.
- Use an assignment statement in that compute block to assign the text to a character variable.

Typically, you assign the text to the break variable or to a variable that does not normally appear in the summary lines.

Report

In this report the text “Subtotal:” replaces the value of the group variable DEPT in the summary lines.

Changing the Value of a Character Variable in the Summary Information				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Subtotal:		\$207,000.00	\$208,429.21	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Subtotal:		\$62,000.00	\$60,627.13	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Subtotal:		\$101,000.00	\$108,640.44	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Subtotal:		\$395,000.00	\$393,857.51	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
Subtotal:		\$10,000.00	\$8,456.99	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  break after dept / skip summarize;

  compute after dept;
  dept='Subtotal:.';
endcomp;

title 'Changing the Value of a Character Variable';
title2 'in the Summary Information';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Inserting a Blank Line between Groups
- Summarizing Variables and Statistics for Groups

Specifying Text and Variable Values between Groups

Implementation

Use customized group summaries to write text and variable values between groups.

- Create a compute block for the break variable.
- Use LINE statements in the compute block to write text and variable values at the breaks.
 - To display text, enclose the text in quotation marks.
 - To display the value of a variable, specify the variable's name followed by the format that you want PROC REPORT to use.

► **Caution** *Formatting Variables in LINE Statements*

If you do not specify a format for a variable in a LINE statement, PROC REPORT does not display the value. ▲

By default, if the SAS system option CENTER is in effect, PROC REPORT centers summary text on the page. For information on controlling the position of the text, see “Controlling the Horizontal Position of Text between Groups” later in this chapter, the discussion of the CENTER option in “PROC REPORT Statement” in Chapter 9, and the discussion of pointer control in “LINE Statement” in Chapter 9.

Report

The customized group summaries in this report include text and the values of DEPT and BUDGET.

The format you use with a variable determines how much space PROC REPORT uses to write its value. Because DEPT is a character variable, the procedure uses trailing blanks to fill extra space. Because BUDGET is a numeric variable, the procedure uses leading blanks to fill extra space. For information on removing the extra blanks in customized summaries see “Controlling the Amount of Space Used to Write a Character Variable” and “Controlling the Amount of Space Used to Write a Numeric Variable” later in this chapter.

Specifying Text and Variables between Groups			1
DEPT	ACCOUNT	BUDGET	
Equipment	lease	\$80,000.00	
	maint	\$22,000.00	
	purchase	\$60,000.00	
	rental	\$10,000.00	
	sets	\$15,000.00	
	tape	\$20,000.00	
The budget for the Equipment department is: \$207,000.			
Facilities	rent	\$48,000.00	
	supplies	\$5,500.00	
	utils	\$8,500.00	
The budget for the Facilities department is: \$62,000.			

Other	advert	\$60,000.00
	musicfee	\$8,000.00
	talent	\$33,000.00
The budget for the Other department is: \$101,000.		
Staff	fulltime	\$295,000.00
	parttime	\$100,000.00
The budget for the Staff department is: \$395,000.		
Travel	gas	\$2,000.00
	leases	\$8,000.00
The budget for the Travel department is: \$10,000.		

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;

  compute after dept;
    line ' ';
    line 'The budget for the ' dept $10. ' department is:'
      budget.sum dollar9. ' .';
    line ' ';
  endcomp;

  title 'Specifying Text and Variables between Groups';
run;

```

Program Notes

- The LINE statements that have a quoted blank as their arguments write blank lines before and after the other customized break lines.
- The longest LINE statement prints three pieces of text and two variables. It prints
 - the text between quotation marks just as it appears in the program
 - the value of DEPT using the \$10. format
 - the value of BUDGET.SUM using the DOLLAR9. format.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Controlling the Amount of Space Used to Write a Character Variable

Implementation

If you write a variable with a typical SAS format, PROC REPORT always uses the same amount of space for that variable even if the values have different lengths. Values for character variables are left-justified within the space. You can eliminate extra space with DATA step code in the compute block that writes the text.

- Use the LENGTH function to determine the length of the current value of the variable. (The LENGTH function ignores trailing blanks.)
- Use an assignment statement to assign the length to a variable that does not appear in the report.
- Use this variable with the \$VARYING. format to print each value of the variable in the appropriate number of horizontal spaces.

Report

In this report each value of DEPT occupies only as many horizontal positions as necessary. For example, **Equipment** occupies 9 spaces, and **Other** occupies 5.

As a consequence of using the \$VARYING. format, the customized summaries have different lengths. By default, if the SAS system option NOCENTER is in effect, PROC REPORT centers customized summary lines. For information on aligning the summary lines, see “Controlling the Horizontal Position of Text between Groups” later in this chapter, the discussion of the CENTER option in “PROC REPORT Statement” in Chapter 9, and the discussion of pointer control in “LINE Statement” in Chapter 9.

Controlling the Amount of Space Used to Write a Character Variable			1
DEPT	ACCOUNT	BUDGET	
Equipment	lease	\$80,000.00	
	maint	\$22,000.00	
	purchase	\$60,000.00	
	rental	\$10,000.00	
	sets	\$15,000.00	
	tape	\$20,000.00	
The Equipment budget is: \$207,000.			
Facilities	rent	\$48,000.00	
	supplies	\$5,500.00	
	utils	\$8,500.00	
The Facilities budget is: \$62,000.			
Other	advert	\$60,000.00	
	musicfee	\$8,000.00	
	talent	\$33,000.00	
The Other budget is: \$101,000.			
Staff	fulltime	\$295,000.00	
	parttime	\$100,000.00	
The Staff budget is: \$395,000.			
Travel	gas	\$2,000.00	
	leases	\$8,000.00	
The Travel budget is: \$10,000.			

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;

  compute after dept;
    line ' ';
    len=length(dept);
    line 'The ' dept $varying. len ' budget is: '
      budget.sum dollar9. '.';
    line ' ';
  endcomp;

  title 'Controlling the Amount of Space Used to Write';
  title2 'a Character Variable';
run;

```

Program Notes

- The LINE statements that have a quoted blank as their arguments write blank lines before and after the other customized break lines.
- The assignment statement creates a new variable, LEN. The value of LEN is the length of the current value of DEPT. If DEPT is **Equipment**, LEN is 9; if DEPT is **Other**, LEN is 5. You calculate this variable to use with the \$VARYING. format. LEN does not appear in the report.
- The longest LINE statement prints three pieces of text and two variables. It prints
 - the text between quotation marks just as it appears in the program.
 - the value of DEPT using the \$VARYING. format. The number of horizontal positions that this format uses depends on the value of LEN. By using the \$VARYING. format, you can print the value of DEPT with the appropriate number of horizontal positions.
 - the value of BUDGET.SUM with the DOLLAR9. format.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Specifying Text and Variables between Groups

Controlling the Amount of Space Used to Write a Numeric Variable

Implementation

If you write a variable with a typical SAS format, PROC REPORT always uses the same amount of space for that variable even if the values have different lengths. Values for numeric variables are right-justified within the space. You can eliminate extra space with DATA step code in the compute block that writes the text.

- Use the PUT function to convert the formatted value of the numeric variable to a character string.
- Use the LEFT function to left-align the character string.
- Use the LENGTH function to determine the length of the string, which differs for different values of the numeric variable. (The LENGTH function ignores trailing blanks.)
- Use an assignment statement to assign the length to a variable that does not appear in the report.
- Use this variable with the \$VARYING. format to print each value of the variable in the appropriate number of horizontal spaces.

Report

In this report each value of BUDGET occupies only as many horizontal positions as necessary. For example, **\$207,000** occupies 8 spaces, and **\$10,000** occupies 7.

As a consequence of using the \$VARYING. format, the customized summaries have different lengths. By default, if the SAS system option CENTER is in effect, PROC REPORT centers customized summary lines on the page. For information on aligning summary lines, see “Controlling the Horizontal Position of Text between Groups” later in this chapter, the discussion of the CENTER option in “PROC REPORT Statement” in Chapter 9, and the discussion of pointer control in “LINE Statement” in Chapter 9.

Controlling the Amount of Space Used to Write a Numeric Variable			1
DEPT	ACCOUNT	BUDGET	
Equipment	lease	\$80,000.00	
	maint	\$22,000.00	
	purchase	\$60,000.00	
	rental	\$10,000.00	
	sets	\$15,000.00	
	tape	\$20,000.00	
The Equipment budget is: \$207,000.			
Facilities	rent	\$48,000.00	
	supplies	\$5,500.00	
	utils	\$8,500.00	
The Facilities budget is: \$62,000.			
Other	advert	\$60,000.00	
	musicfee	\$8,000.00	
	talent	\$33,000.00	
The Other budget is: \$101,000.			
Staff	fulltime	\$295,000.00	
	parttime	\$100,000.00	
The Staff budget is: \$395,000.			
Travel	gas	\$2,000.00	
	leases	\$8,000.00	
The Travel budget is: \$10,000.			

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;

  compute after dept;
    line ' ';
    len1=length(dept);
    makechar=left(put(budget.sum,dollar9.));
    len2=length(makechar);
    line 'The ' dept $varying. len1 ' budget is: '
        makechar $varying. len2 ' .';
    line ' ';
  endcomp;

  title 'Controlling the Amount of Space Used to Write';
  title2 'a Numeric Variable';
run;

```

Program Notes

- The assignment statement for MAKECHAR converts the formatted value of BUDGET to a character variable and left-aligns the result.
- The assignment statement for LEN2 creates a new variable, LEN2. The value of LEN2 is the length of the current value of BUDGET. If BUDGET is **\$207,000**, LEN2 is 8; if BUDGET is **\$10,000**, LEN2 is 7. You calculate this variable to use with the \$VARYING. format. LEN2 does not appear in the report.
- The longest LINE statement prints three pieces of text and two variables. It prints
 - the text between quotation marks just as it appears in the program.
 - the values of DEPT and BUDGET.SUM using the \$VARYING. format.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Inserting Multiple Blank Lines between Groups
- Specifying Text and Variable Values between Groups
- Controlling the Amount of Space Used to Write a Character Variable

Controlling the Horizontal Position of Text between Groups

Implementation

- Use the pointer control “@” to specify the number of the column in which you want to begin writing.

By default, when the SAS system option CENTER is in effect, PROC REPORT centers text in customized summaries. If you use pointer control in a compute block, centering is turned off for that compute block.

Report

In this report, each customized group summary begins in column 12.

Controlling the Horizontal Position of Text between Groups			1
DEPT	ACCOUNT	BUDGET	
Equipment	lease	\$80,000.00	
	maint	\$22,000.00	
	purchase	\$60,000.00	
	rental	\$10,000.00	
	sets	\$15,000.00	
	tape	\$20,000.00	
The Equipment budget is: \$207,000.			
Facilities	rent	\$48,000.00	
	supplies	\$5,500.00	
	utils	\$8,500.00	
The Facilities budget is: \$62,000.			
Other	advert	\$60,000.00	
	musicfee	\$8,000.00	
	talent	\$33,000.00	
The Other budget is: \$101,000.			
Staff	fulltime	\$295,000.00	
	parttime	\$100,000.00	
The Staff budget is: \$395,000.			
Travel	gas	\$2,000.00	
	leases	\$8,000.00	
The Travel budget is: \$10,000.			

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;

  compute after dept;
    line ' ';
    len=length(dept);
    line @12 'The ' dept $varying. len ' budget is: '
          budget.sum dollar9. '.';
    line ' ';
  endcomp;

  title 'Controlling the Horizontal Position of Text between Groups';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Inserting Multiple Blank Lines between Groups
- Specifying Text and Variable Values between Groups
- Controlling the Amount of Space Used to Write a Character Variable

Initializing Variables for Each Group

Implementation

For some reports, you may want to do calculations using variables that do not appear in the report (*DATA step variables*). The key difference between DATA step variables and report variables (variables that appear in the report) is in how PROC REPORT handles their values from one row of the report to the next. PROC REPORT initializes report variables to missing at the beginning of each row. It retains the values of DATA step variables unless you explicitly reset them. (For more information see Chapter 10, “How PROC REPORT Builds a Report.”)

For example, you might want to count how many accounts in each department have spent more than their budgets. For this task, you must initialize the value of the DATA step variable that counts the accounts before you begin to process a new group.

To initialize variables for each group

- create a compute block before the group variable
- use assignment statements to initialize DATA step variables.

Report

In this report, the customized group summaries tell how many accounts in each department overspent their budgets. You must reset the counter to 0 before each group.

Initializing Variables for Each Group					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	
	2 accounts exceeded their budgets.				
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
0 accounts exceeded their budgets.					
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
1 accounts exceeded their budgets.					
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
0 accounts exceeded their budgets.					
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	
0 accounts exceeded their budgets.					

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define balance / computed format=dollar11.2;

  compute before dept;
    count=0;
  endcomp;

  compute balance;
    balance=budget.sum-actual.sum;
    if balance<0 and account ne ' ' then count+1;
  endcomp;

  compute after dept;
    line ' ';
    line count 1. ' accounts exceeded their budgets.';
    line ' ';
  endcomp;

  title 'Initializing Variables for Each Group';
run;

```

Note: When you create a compute block, PROC REPORT calculates the corresponding default summary line and processes it as part of the report even if the summary line does not appear in the report. However, you may not want to execute the compute block at a summary line. In this case, for example, summary lines do not represent individual accounts. Therefore, on summary lines where BALANCE is less than 0, you do not want to increment COUNT. The following shaded condition in the IF statement suppresses execution of the statement on the summary lines because only at the summary lines does ACCOUNT have a missing value:

```

if balance<0 and account ne ' ' then count+1;

```

For information on controlling whether statements execute at a break, see “Controlling Whether Statements in a Compute Block Execute between Groups” later in this chapter.

Building Blocks

- *Specifying and Ordering Columns to Display in Chapter 1*
- *Consolidating Multiple Observations into One Row in Chapter 1*
- *Adding a Variable Computed from a Statistic in Chapter 5*
- *Inserting Multiple Blank Lines between Groups*
- *Specifying Text and Variable Values between Groups*
- *Controlling Whether Statements in a Compute Block Execute between Groups*

Placing Conditional Text between Groups

Implementation

- Use conditional processing (IF-THEN, IF-THEN/ELSE, or SELECT statements) to store the appropriate version of the text in a character variable.
- Use a LINE statement to write that variable in your customized summary lines.

Note: You cannot use the LINE statement in conditional statements because it does not take effect until PROC REPORT has executed all other statements in the compute block.

A character variable takes its length from the first value assigned to it or from a LENGTH statement if the LENGTH statement precedes the first use of the variable. Use a LENGTH statement in the compute block to assign the necessary length if the following are both true:

- The different versions of text are not the same length.
- The longest version is not the first version.

Report

In this report, two different pieces of text appear in the group summaries.

- If no accounts in the department overspent their budgets, the text says, “All accounts in this department were within their budgets.”
- If one or more accounts in the department overspent their budgets, the text says, “At least one account in this department overspent its budget.”

Placing Conditional Text between Groups					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	
At least one account in this department overspent its budget					
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
All accounts in this department were within their budgets.					
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
At least one account in this department overspent its budget					
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
All accounts in this department were within their budgets.					
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	
All accounts in this department were within their budgets.					

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept      / group;
  define account  / group;
  define budget   / analysis sum;
  define actual   / analysis sum;
  define balance  / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
    if balance<0 then over='yes';
  endcomp;

  compute before dept;
    over='no';
  endcomp;

  compute after dept;
    length text1 $ 60;
    line ' ';
    if over='yes' then
      text1='At least one account in this department overspent its budget.';
    else text1='All accounts in this department were within their budgets.';
    line text1 $60.;
    line ' ';

```

```
endcomp;  
  
title 'Placing Conditional Text between Groups';  
run;
```

Program Notes

- The LENGTH statement assigns a length of 60 to TEXT1.
- The conditional logic in the IF-THEN/ELSE statements assigns the appropriate value to TEXT1.
- The LINE statement prints the current version of TEXT1 with a \$60. format.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Inserting Multiple Blank Lines between Groups
- Specifying Text and Variable Values between Groups
- Initializing Variables for Each Group

Controlling Whether Statements in a Compute Block Execute between Groups

Implementation

When you create a compute block for a break, PROC REPORT calculates the corresponding default summary line and processes it as part of the report even if the summary line does not appear in the report. However, for some reports you will want to execute code in a compute block for detail rows of the report but not for summary lines, or for summary lines but not for detail rows. To do so,

- Use conditional statements: IF-THEN, IF-THEN/ELSE, or SELECT
- Include a condition that is true for detail rows but not for summary lines, or true for summary lines but not detail rows.

In a break between groups, all character variables to the right of the break variable have missing values. By default, the break variable has a value, but if you use the SUPPRESS option on the BREAK statement, the break variable has a missing value. Use this information to restrict processing.

► **Caution** *Missing Values in Your Data Set*

If the variable whose value you use to restrict processing is a group, order, or across variable and you use the MISSING option in the PROC REPORT statement or if the variable is a display variable, missing values in the input data set for that variable affect whether the conditional code executes. To avoid undesirable results in these circumstances, either use a condition based on a variable that has no missing values in the input data set or use a DATA step to reassign the missing value to a nonmissing value before you invoke PROC REPORT. ▲

Report

In this report, the customized group summaries tell how many accounts in each department overspent their budgets.

Controlling Whether Statements in a Compute Block Execute between Groups					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	
2 accounts exceeded their budgets.					
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
0 accounts exceeded their budgets.					
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
1 accounts exceeded their budgets.					
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
0 accounts exceeded their budgets.					
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	
0 accounts exceeded their budgets.					

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define balance / computed format=dollar11.2;

  compute before dept;
    count=0;
  endcomp;

  compute balance;
    balance=budget.sum-actual.sum;
    if balance<0 and account ne '' then count+1;
  endcomp;

  compute after dept;
    line ' ';
    line count 1. ' accounts exceeded their budgets.';
    line ' ';
  endcomp;

  title 'Controlling Whether Statements in a Compute Block Execute';

```

```
title2 'between Groups';  
run;
```

Program Notes

- Without the shaded condition in the IF statement, PROC REPORT incorrectly increments COUNT at the breaks between groups if the value of BALANCE for the group is less than 0. Because ACCOUNT is missing in the summary lines, the shaded condition ensures that PROC REPORT does not increment COUNT on summary lines.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Inserting Multiple Blank Lines between Groups
- Initializing Variables for Each Group
- Specifying Text and Variable Values between Groups

Chapter 7 Working with the Report as a Whole

Introduction	131
<i>Variable Values at a Break at the Beginning or End of a Report</i>	132
<i>Default Report Summaries</i>	132
<i>Customized Report Summaries</i>	133
<i>Features that You Can Use in a Compute Block</i>	134
<i>How to Reference Report Variables in a Compute Block</i>	134
Summarizing Variables and Statistics for the Whole Report	136
Overlining Summary Information	138
Underlining Summary Information	140
Inserting a Blank Line after Summary Information at the Beginning of a Report	142
Inserting Multiple Blank Lines after Summary Information at the Beginning of a Report	144
Changing the Value of a Character Variable in the Summary Information	146
Specifying Text and Variable Values at the Beginning or End of a Report	148
Controlling the Horizontal Position of Text at the Beginning or End of a Report	151
Initializing Variables at the Beginning of a Report	153
Placing Conditional Text at the Beginning or End of a Report	156
Controlling Whether Statements in a Compute Block Execute at the Beginning and End of a Report	159

Introduction

You can summarize information about the whole report at the beginning or end of the report. To do so, you create a *break* at the beginning or end of the report.

Break lines are lines of text (including blanks) that appear at a break. There are two ways to display break lines. The first way is simpler. It produces a default report summary. The second way is more flexible. It produces a customized report summary and provides a way of slightly modifying the default report summary.

You can use both default and customized report summaries in the same report.

You can also create breaks (and therefore summaries) whenever the value of a group or order variable changes. For information on using breaks and groups together, see Chapter 6. For information on using order variables see “Ordering Rows” in Chapter 1 and the discussion of the ORDER option in the documentation for the DEFINE statement in Chapter 9.

Variable Values at a Break at the Beginning or End of a Report

In a break at the beginning or end of a report

- character variables have missing values.
- statistics and analysis variables are calculated for the entire report.
- computed variables are calculated by executing the statements in the COMPUTE block attached to that variable, just as they are for any other row of the report (see Chapter 5).

Default Report Summaries

Default report summaries can

- visually separate summary information from the rest of the report
- summarize statistics and calculate computed variables for the whole report.

The default report summary in this report sums the analysis variables BUDGET and ACTUAL for the whole report. (BUDGET and ACTUAL are analysis variables used to calculate the SUM statistic.) A row of hyphens separates the summary from the rest of the report.

				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
		-----	-----	
		\$775,000.00	\$780,011.28	

To create a default report summary, write an RBREAK (report break) statement. An RBREAK statement must include (in this order)

1. the keyword RBREAK
2. the BEFORE or AFTER argument.

If you want the summary to appear before the first row of the report, use the BEFORE argument. If you want the summary to appear after the last row of the report, use the AFTER

argument.

The RBREAK statement supports options that control the appearance of a default report summary. This chapter illustrates the use of most of these options. You can use any combination of options in the RBREAK statement. The order in which you use the options does not affect the report.

Customized Report Summaries

When you create a customized report summary, you execute DATA step statements (and one statement unique to PROC REPORT, the LINE statement) at the break. These statements display text, calculate values for the whole report, or both.

The customized report summary in this report summarizes the analysis variables BUDGET and ACTUAL, just as the default report did. However, the customized summary includes additional text and a box around the summary information.

DEPT	ACCOUNT	BUDGET	ACTUAL
Equipment	lease	\$80,000.00	\$80,000.00
	maint	\$22,000.00	\$18,217.42
	purchase	\$60,000.00	\$66,051.53
	rental	\$10,000.00	\$9,481.81
	sets	\$15,000.00	\$16,422.30
	tape	\$20,000.00	\$18,256.15
Facilities	rent	\$48,000.00	\$48,000.00
	supplies	\$5,500.00	\$4,959.03
	utils	\$8,500.00	\$7,668.10
Other	advert	\$60,000.00	\$69,802.62
	musicfee	\$8,000.00	\$7,426.45
	talent	\$33,000.00	\$31,411.37
Staff	fulltime	\$295,000.00	\$293,988.43
	parttime	\$100,000.00	\$99,869.08
Travel	gas	\$2,000.00	\$1,522.19
	leases	\$8,000.00	\$6,934.80

1

```

*****
*
* The budget for the entire company was $775,000.00.
* The amount spent by the entire company was $780,011.28.
*
*****

```

The statements that create a customized report summary must be in a *compute block*. A compute block for a customized report summary contains (in this order)

1. a COMPUTE statement
2. selected SAS language features, LINE statements, or both,
3. an ENDCOMP statement.

A list of features that you can use between the COMPUTE and ENDCOMP statements appears later in this section.

A COMPUTE statement for a report summary must include (in this order)

1. the keyword COMPUTE
2. the BEFORE or AFTER argument.

If you want the summary to appear before the first row of the report, use the BEFORE argument. If you want the summary to appear after the last row of the report, use the AFTER argument.

Features that You Can Use in a Compute Block

Compute blocks support these SAS language features:

- DM statement
- %INCLUDE statement
- selected DATA step statements:

assignment	LENGTH
CALL	LINK
DO (all forms)	RETURN
END	SELECT
GO TO	sum
IF-THEN/ELSE	

- comments
- null statements
- macro variables
- all DATA step functions.

Compute blocks for a customized summary also support the LINE statement. For information on the LINE statement see Chapter 9, “The REPORT Language.”

For more information on compute blocks see “COMPUTE Statement” in Chapter 9.

How to Reference Report Variables in a Compute Block

Reference report variables in a compute block in one of three ways:

- by name.
- by a compound name that identifies both the variable and the name of the statistic you calculate with it. A compound name has this form:

variable-name.statistic

- by column number, in the form

Cn

where *n* is the number of the column (from left to right) in the report. When you determine the column number, count columns that are not displayed (see “Suppressing the Display of an Item” in Chapter 3).

The following table shows when to use each type of reference in a customized report summary.

If the variable you reference is this type ...	then refer to it by ...	For example ...
group	name	DEPT
order	name	DEPT
computed	name	DEPT
display	name	DEPT
display sharing a column with a statistic	a compound name	BUDGET.SUM
analysis	a compound name	BUDGET.MEAN
any type sharing a column with an across variable	column number	_C3_

► **Caution Aliases Are Not Valid in Compute Blocks**

Aliases that you create in a COLUMN statement are valid only in DEFINE statements. You cannot use them in compute blocks. ▲

Compute blocks can produce an endless variety of customized summaries. They can also slightly modify a default summary. This chapter gets you started. For more information see “RBREAK Statement” and “LINE Statement” in Chapter 9, “The REPORT Language.”

Summarizing Variables and Statistics for the Whole Report

Implementation

- Use the SUMMARIZE option in the RBREAK statement.

The SUMMARIZE option writes a default *summary line*. A default summary line summarizes statistics and calculates computed variables for multiple observations, in this case for the whole report.

Report

In this report BUDGET and ACTUAL are analysis variables used to calculate the SUM statistic. Therefore, the values for BUDGET and ACTUAL in the summary line are sums for all rows of the report. If you defined these variables as calculating a different statistic (MEAN, for example), the values in the summary line would be the value of that statistic for all rows in the report.

Summarizing Variables and Statistics for the Whole Report				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
		\$775,000.00	\$780,011.28	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  rbreak after / summarize;

  title 'Summarizing Variables and Statistics for the Whole Report';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1

Overlining Summary Information

Implementation

- Use the OL (overline) or DOL (double overline) option in the RBREAK statement.

The OL option writes a line of hyphens (-) above each value that appears in the summary line or that would appear in the summary line if you specified the SUMMARIZE option.

The DOL option writes a line of equals signs (=) above each value that appears in the summary line or that would appear in the summary line if you specified the SUMMARIZE option.

In both cases, the overlining extends for the width of the column even if the value does not fill the column.

Note: If you use both options, PROC REPORT honors only OL.

Report

Overlining Summary Information				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
		-----	-----	
		\$775,000.00	\$780,011.28	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  rbreak after / summarize o1;

  title 'Overlining Summary Information';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Summarizing Variables and Statistics for the Whole Report

Underlining Summary Information

Implementation

- Use the UL (underline) or DUL (double underline) option in the RBREAK statement.

The UL option writes a line of hyphens (-) below each value that appears in the summary line or that would appear in the summary line if you specified the SUMMARIZE option.

The DUL option writes a line of equals signs (=) below each value that appears in the summary line or that would appear in the summary line if you specified the SUMMARIZE option.

In both cases, the underlining extends for the width of the column even if the value does not fill the column.

Note: If you use both options, PROC REPORT honors only UL.

Report

Underlining Summary Information				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
		\$775,000.00	\$780,011.28	
		-----	-----	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  rbreak before / summarize u1;

  title 'Underlining Summary Information';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Summarizing Variables and Statistics for the Whole Report

Inserting a Blank Line after Summary Information at the Beginning of a Report

Implementation

- Use the SKIP option in the RBREAK statement.

Report

Inserting a Blank Line after Summary Information at the Beginning of a Report				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
		\$775,000.00	\$780,011.28	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept / group;
  define account / group;
  define budget / analysis sum;
  define actual / analysis sum;

  rbreak before / summarize skip;

  title 'Inserting a Blank Line after Summary Information';
  title2 'at the Beginning of a Report';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Summarizing Variables and Statistics for the Whole Report

Inserting Multiple Blank Lines after Summary Information at the Beginning of a Report

Implementation

Use a customized report summary to insert multiple blank lines after the summary information.

- Create a compute block for the beginning of the report.
- Use a LINE statement with a blank for an argument for each blank line you want.

The LINE statement writes break lines containing text, values calculated for a group of observations (in this case for the whole report), or both. The LINE statement must be between a COMPUTE and an ENDCOMP statement.

Report

This report has two blank lines after the summary information at the beginning of the report.

Inserting Multiple Blank Lines after Summary Information				1
at the Beginning of a Report				
DEPT	ACCOUNT	BUDGET	ACTUAL	
		\$775,000.00	\$780,011.28	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  rbreak before / summarize;

  compute before;
    line ' ';
    line ' ';
  endcomp;

  title 'Inserting Multiple Blank Lines after Summary Information';
  title2 'at the Beginning of a Report';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Summarizing Variables and Statistics for the Whole Report

Changing the Value of a Character Variable in the Summary Information

Implementation

You may want to put text in the default summary line for emphasis.

- Create a compute block for the beginning or end of the report, depending on where the summary information appears.
- Use an assignment statement in that compute block to assign the text to a character variable.

Report

In this report the text “Total:” replaces the value of DEPT in the summary line. DEPT is a character variable. By default, it has a missing value in the summary line.

Changing the Value of a Character Variable in the Summary Information				1
DEPT	ACCOUNT	BUDGET	ACTUAL	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
Facilities	tape	\$20,000.00	\$18,256.15	
	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
Other	utils	\$8,500.00	\$7,668.10	
	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
Staff	talent	\$33,000.00	\$31,411.37	
	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	
Total:		\$775,000.00	\$780,011.28	

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget actual;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;

  rbreak after / summarize;

  compute after;
    dept='Total:.';
  endcomp;

  title 'Changing the Value of a Character Variable';
  title2 'in the Summary Information';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Summarizing Variables and Statistics for the Whole Report

Specifying Text and Variable Values at the Beginning or End of a Report

Implementation

Use a customized report summary to write text and variable values at the beginning or end of the report.

- Create a compute block for the beginning or end of the report.
- Use LINE statements in the compute block to write text and variable values at the break.
 - To display text, enclose the text in quotation marks.
 - To display the value of a variable, specify the variable's name, followed by the format you want PROC REPORT to use.

► **Caution** *Formatting Variables in LINE Statements*

If you do not specify a format for a variable in a LINE statement, PROC REPORT does not display the value. ▲

By default, if the SAS system option CENTER is in effect, PROC REPORT centers summary text on the page. For information on controlling the position of the text, see “Controlling the Horizontal Position of Text at the Beginning or End of a Report” later in this chapter, the discussion of the CENTER option in “PROC REPORT Statement” in Chapter 9, and the discussion of pointer control in “LINE Statement” in Chapter 9.

Report

The customized report summary in this report includes text and the value of BUDGET for the entire company.

The format you use with a variable determines how much space PROC REPORT uses to write its value. Because BUDGET is a numeric variable, the procedure uses leading blanks to fill extra space. For information on removing the extra blanks in customized summaries see “Controlling the Amount of Space Used to Write a Character Variable” and “Controlling the Amount of Space Used to Write a Numeric Variable” in Chapter 6.

Specifying Text and Variable Values at the Beginning or End of a Report			1
DEPT	ACCOUNT	BUDGET	
Equipment	lease	\$80,000.00	
	maint	\$22,000.00	
	purchase	\$60,000.00	
	rental	\$10,000.00	
	sets	\$15,000.00	
	tape	\$20,000.00	
Facilities	rent	\$48,000.00	
	supplies	\$5,500.00	
	utils	\$8,500.00	
Other	advert	\$60,000.00	
	musicfee	\$8,000.00	
	talent	\$33,000.00	
Staff	fulltime	\$295,000.00	
	parttime	\$100,000.00	
Travel	gas	\$2,000.00	
	leases	\$8,000.00	
The budget for the entire company is			\$775,000.

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;

  compute after;
  line ' ';
  line 'The budget for the entire company is '
  budget.sum dollar9. '.';
  line ' ';
  endcomp;

  title 'Specifying Text and Variable Values at the Beginning or End';
  title2 'of a Report';
run;
```

Program Notes

- The LINE statements that have a quoted blank as their arguments write blank lines before and after the other customized break lines.
- The longest LINE statement prints two pieces of text and one variable. It prints
 - the text between quotation marks just as it appears in the program.
 - the value of BUDGET.SUM using the DOLLAR9. format.

Building Blocks

- *Specifying and Ordering Columns to Display in Chapter 1*
- *Consolidating Multiple Observations into One Row in Chapter 1*

Controlling the Horizontal Position of Text at the Beginning or End of a Report

Implementation

- Use the pointer control “@” to specify the number of the column in which you want to begin writing.

By default, if the SAS system option CENTER is in effect, PROC REPORT centers text in customized summaries. If you use pointer control in a compute block, centering is turned off for that compute block.

Report

In this report, the customized report summary begins in column 12.

Controlling the Horizontal Position of Text at the Beginning or End of a Report			1
DEPT	ACCOUNT	BUDGET	
Equipment	lease	\$80,000.00	
	maint	\$22,000.00	
	purchase	\$60,000.00	
	rental	\$10,000.00	
	sets	\$15,000.00	
	tape	\$20,000.00	
Facilities	rent	\$48,000.00	
	supplies	\$5,500.00	
	utils	\$8,500.00	
Other	advert	\$60,000.00	
	musicfee	\$8,000.00	
	talent	\$33,000.00	
Staff	fulltime	\$295,000.00	
	parttime	\$100,000.00	
Travel	gas	\$2,000.00	
	leases	\$8,000.00	
The budget for the entire company is \$775,000.			

Sample Program

```
proc report data=budget2 nowindows;
  column dept account budget;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;

  compute after;
    line ' ';
    line @12 'The budget for the entire company is '
          budget.sum dollar8. '.';
    line ' ';
  endcomp;

  title 'Controlling the Horizontal Position of Text';
  title2 'at the Beginning or End of a Report';
run;
```

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Specifying Text and Variable Values at the Beginning or End of a Report

Initializing Variables at the Beginning of a Report

Implementation

For some reports, you may want to do calculations using variables that do not appear in the report (*DATA step variables*). The key difference between DATA step variable and report variables (variables that appear in the report) is in how PROC REPORT handles their values from one row of the report to the next. PROC REPORT initializes report variables to missing at the beginning of each row. It retains the values of DATA step variables unless you explicitly reset them. (For more information see Chapter 10, “How PROC REPORT Builds a Report.”)

For example, you might want to count how many accounts in the company have spent more than their budgets. For a task like this, you must initialize the value of the DATA step variable that counts the accounts at the beginning of a report.

To initialize variables at the beginning of a report

- create a compute block at the beginning of the report
- use assignment statements to initialize DATA step variables.

Report

This customized report summary tells how many accounts in the company overspent their budgets. The counter is set to 0 at the beginning of the report.

Initializing Variables at the Beginning of a Report					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	
3 accounts exceeded their budgets.					

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define balance / computed format=dollar11.2;

  compute before;
    count=0;
  endcomp;

  compute balance;
    balance=budget.sum-actual.sum;
    if balance<0 and account ne ' ' then count+1;
  endcomp;

  compute after;
    line ' ';
    line count 1. ' accounts exceeded their budgets.';
    line ' ';
  endcomp;

  title 'Initializing Variables at the Beginning of a Report';
run;

```

Note: When you create a compute block, PROC REPORT calculates the corresponding default summary line and processes it as part of the report even if the summary line does not appear in the report. However, you may not want to execute the compute block at a summary line. In this case, for example, the summary line does not represent individual accounts. Therefore, if BALANCE is less than 0 on the summary line, you do not want to increment COUNT. The following shaded condition in the IF statement suppresses execution of the statement on the summary line because only at the summary lines does ACCOUNT have a missing value:

```

if balance<0 and account ne ' ' then count+1;

```

For information on controlling whether statements execute at a break, see “Controlling Whether Statements in a Compute Block Execute at the Beginning and End of a Report” later in this chapter.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Adding a Variable Computed from a Statistic in Chapter 5
- Specifying Text and Variable Values at the Beginning or End of a Report
- Controlling Whether Statements in a Compute Block Execute at the Beginning and End of a Report

Placing Conditional Text at the Beginning or End of a Report

Implementation

- Use conditional processing (IF-THEN, IF-THEN/ELSE, or SELECT statements) to store the appropriate version of the text in a character variable.
- Use a LINE statement to write that variable in your customized summary lines.

Note: You cannot use the LINE statement in conditional statements because it does not take effect until PROC REPORT has executed all other statements in the compute block.

A character variable takes its length from the first value assigned to it or from a LENGTH statement if the LENGTH statement precedes the first use of the variable. Use a LENGTH statement in the compute block to assign the necessary length if the following are both true:

- The different versions of text are not the same length.
- The longest version is not the first version.

Report

In these reports, the text that appears in the customized summary depends on whether or not any departments overspent their budgets. The second report includes only the Travel and Facilities departments, neither of which overspent its budget.

- If no accounts overspent their budgets, the text says, “All accounts were within their budgets.”
- If one or more accounts overspent their budgets, the text says, “At least one account overspent its budget.”

Placing Conditional Text at the Beginning or End of a Report					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
Facilities	tape	\$20,000.00	\$18,256.15	\$1,743.85	
	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	

At least one account overspent its budget.

Placing Conditional Text at the Beginning or End of a Report					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	
All accounts were within their budgets.					

Sample Program

Only the code for the second report appears here. The code for the first report is the same except that it does not include the WHERE clause.

```
proc report data=budget2 nowindows;
  where dept='Facilities' or dept='Travel';

  column dept account budget actual balance;

  define dept / group;
  define account / group;
  define budget / analysis sum;
  define actual / analysis sum;
  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
    if balance<0 then over='yes';
  endcomp;

  compute after;
    line ' ';
    length text1 $42.;
    if over='yes' then
      text1='At least one account overspent its budget.';
    else text1='All accounts were within their budgets.';
    line text1 $42.;
  endcomp;

  title 'Placing Conditional Text at the Beginning or End';
  title2 'of a Report';
run;
```

Program Notes

- The LENGTH statement assigns a length of 42 to TEXT1.
- The conditional logic in the IF/THEN else statements assigns the appropriate value to TEXT1.
- The ELSE statement assigns an alternative value to TEXT1.
- The LINE statement prints the current version of TEXT1 with a \$42. format.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Specifying Text and Variable Values at the Beginning or End of a Report

Controlling Whether Statements in a Compute Block Execute at the Beginning and End of a Report

Implementation

When you create a compute block for a break, PROC REPORT calculates the corresponding default summary line and processes it as part of the report even if the summary line does not appear in the report. However, for some reports you will want to execute code in a compute block for detail rows of the report but not for summary lines, or for summary lines but not for detail rows. To do so,

- Use conditional statements: IF-THEN, IF-THEN/ELSE, or SELECT
- Include a condition that is true for detail rows but not for summary lines, or true for summary lines but not detail rows.

In a report break, character variables have missing values. Use this information to restrict processing.

► **Caution** *Missing Values in Your Data Set*

If the variable whose value you use to restrict processing is a group, order, or across variable and you use the MISSING option in the PROC REPORT statement or if the variable is a display variable, missing values in the input data set for that variable affect whether the conditional code executes. To avoid undesirable results in these circumstances, either use a condition based on a variable that has no missing values in the input data set or use a DATA step to reassign the missing value to a nonmissing value before you invoke PROC REPORT. ▲

Report

In this report, the report summary tells how many accounts in the company overspent their budgets.

Controlling Whether Statements in a Compute Block Execute at the Beginning and End of a Report					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	

3 accounts exceeded their budgets.

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define balance / computed format=dollar11.2;

  compute before;
    count=0;
  endcomp;

  compute balance;
    balance=budget.sum-actual.sum;
    if balance<0 and account ne ' ' then count+1;
  endcomp;

  compute after;
    line ' ';
    line count 1. ' accounts exceeded their budgets.';
    line ' ';
  endcomp;

  title 'Controlling Whether Statements in a Compute Block';
  title2 'Execute at the Beginning and End of a Report';
run;

```

Without the shaded condition in the IF statement, PROC REPORT would incorrectly increment the value of COUNT on the summary line if the value of BALANCE is less than 0 (which it is in this case). Because ACCOUNT is missing on the summary lines, PROC REPORT increments COUNT only on detail rows where BALANCE is less than 0.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Initializing Variables at the Beginning of a Report
- Specifying Text and Variable Values at the Beginning or End of a Report

Chapter 8 Calculating Percentages

Introduction 163

Building a Report 163

Compound Names 164

Initializing Variables 164

Calculating Percentages within Groups 166

Calculating Percentages over All Groups 169

Calculating Cumulative Percentages within Groups 172

Calculating Cumulative Percentages for the Whole Report 175

Introduction

Before you can calculate percentages, you need to understand

- how to use the same compound name to mean different things in different parts of the report
- how to initialize DATA step variables.

You also need some understanding of how PROC REPORT builds a report. This chapter provides enough information about this process for you to calculate percentages and cumulative percentages within groups and for a whole report. For more information on how PROC REPORT constructs a report, see Chapter 10.

Building a Report

When PROC REPORT builds a report it

1. consolidates the data by group or order variables
2. calculates (and stores in a temporary file) all statistics you use in the report, those for detail rows as well as those for default summary lines (see the introductions to Chapter 6 and Chapter 7)
3. begins constructing the report row by row.

Consequently,

- you can use group statistics in compute blocks for a break before, as well as after, the group variable.
- you can use statistics for the whole report in a compute block at the beginning, as well as at the end, of the report.

You reference these statistics with the appropriate compound name.

Compound Names

When you use statistics in a report, you refer to them in compute blocks by a compound name like BUDGET.SUM. However, in different parts of the report, that same name takes on different meanings. Understanding these different meanings is crucial to understanding how to calculate percentages. Consider this report:

DEPT	ACCOUNT	BUDGET
Equipment	lease	\$80,000.00
	maint	\$22,000.00
	purchase	\$60,000.00
	rental	\$10,000.00
	sets	\$15,000.00
	tape	\$20,000.00
-----		-----
Equipment		\$207,000.00
Facilities	rent	\$48,000.00
	supplies	\$5,500.00
	utils	\$8,500.00
-----		-----
Facilities		\$62,000.00
Other	advert	\$60,000.00
	musicfee	\$8,000.00
	talent	\$33,000.00
-----		-----
Other		\$101,000.00
Staff	fulltime	\$295,000.00
	parttime	\$100,000.00
-----		-----
Staff		\$395,000.00
Travel	gas	\$2,000.00
	leases	\$8,000.00
-----		-----
Travel		\$10,000.00
=====		=====
Total:		\$775,000.00
=====		=====

Here BUDGET.SUM takes on three different meanings:

- In detail rows, the value is the budget for one account in a department for two quarters of the year.
- In the group summary lines, the value is the budget for one department.
- In the report summary line, the value is the budget for the entire company.

Initializing Variables

To calculate percentages you need to initialize variables in compute blocks in your report.

Variables that appear in one or more columns of the report are *report variables*. At the beginning of each row PROC REPORT initializes all report variables to missing. It then fills in the columns from left to right.

Variables that you use in a compute block but do not display in the report are *DATA step variables*. At the beginning of the report, PROC REPORT initializes all numeric DATA step variables to 0 and all character DATA step variables to a blank. The value of a DATA

step variable does not change until you specifically assign a value to it. PROC REPORT retains the value of a DATA step variable from the execution of one compute block to another.

Because all compute blocks share the current values of all variables, you can initialize DATA step variables at a break at the beginning of the report or at a break before a variable. All the reports in this chapter initialize variables in one or more compute blocks. Other examples of initialization are in “Initializing Variables for Each Group” in Chapter 6 and “Initializing Variables at the Beginning of a Report” in Chapter 7.

Calculating Percentages within Groups

Implementation

- Define a group variable.
- Define the variable that you want to compute percentages for as an analysis variable used to calculate the SUM statistic (the default for numeric variables).
- Include both the group variable and the analysis variable in the COLUMN statement.
- Create a compute block before the group variable. In this compute block, store the SUM statistic for the analysis variable in a DATA step variable.
- Create a report variable for the percentages that you are calculating.
 - Define this variable as a computed variable with the appropriate format.
 - Include this variable in the COLUMN statement.
 - Create a compute block for this variable. In the assignment statement that calculates the percentage variable, use the compound name of the analysis variable as the numerator and the DATA step variable that stores the SUM statistic as the denominator.

Report

In this report the computed variable PCTDEPT is the percentage of the departmental budget that is assigned to the current row. On detail rows PCTDEPT is the percentage assigned to individual accounts. On summary rows PCTDEPT is the percentage assigned to the whole department (100%).

Calculating Percentages within Groups				1
DEPT	ACCOUNT	BUDGET	PCTDEPT	
Equipment	lease	\$80,000.00	38.6%	
	maint	\$22,000.00	10.6%	
	purchase	\$60,000.00	29.0%	
	rental	\$10,000.00	4.8%	
	sets	\$15,000.00	7.2%	
	tape	\$20,000.00	9.7%	
Equipment		\$207,000.00	100.0%	
Facilities	rent	\$48,000.00	77.4%	
	supplies	\$5,500.00	8.9%	
	utils	\$8,500.00	13.7%	
Facilities		\$62,000.00	100.0%	
Other	advert	\$60,000.00	59.4%	
	musicfee	\$8,000.00	7.9%	
	talent	\$33,000.00	32.7%	
Other		\$101,000.00	100.0%	
Staff	fulltime	\$295,000.00	74.7%	
	parttime	\$100,000.00	25.3%	
Staff		\$395,000.00	100.0%	
Travel	gas	\$2,000.00	20.0%	
	leases	\$8,000.00	80.0%	
Travel		\$10,000.00	100.0%	

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget pctdept;
  define dept / group;
  define account / group;
  define budget / analysis sum;
  define pctdept / computed format=percent8.1;

  compute before dept;
    deptbud=budget.sum;
  endcomp;

  compute pctdept;
    pctdept=budget.sum/deptbud;
  endcomp;

  break after dept / skip summarize;

  title 'Calculating Percentages within Groups';
run;

```

Program Notes

- The DEFINE statement for DEPT defines it as a group variable.
- The DEFINE statement for BUDGET defines it as an analysis variable used to calculate the SUM statistic.
- The DEFINE statement for PCTDEPT defines it as a computed variable and assigns it the PERCENT8.1 format.
- The COLUMN statement includes the group variable DEPT, the analysis variable BUDGET, and the computed variable PCTDEPT.
- The assignment statement in the compute block before DEPT stores the value of BUDGET.SUM in DEPTBUD. This assignment statement executes only once for each group, just before PROC REPORT constructs the first row of the group. Here, BUDGET.SUM is the budget for a whole department.
- The assignment statement in the compute block for PCTDEPT calculates the percentage of the department's budget assigned to each account. This assignment statement executes once for each row of the report. In detail rows, BUDGET.SUM is the budget for one account. In summary lines, BUDGET.SUM is the budget for one department.

You do not need to multiply PCTDEPT by 100 if you use the PERCENT. format to write the percentage.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Adding a Variable Computed from a Statistic in Chapter 5
- Specifying Formats in Chapter 3
- Inserting a Blank Line between Groups in Chapter 6
- Summarizing Variables and Statistics for Groups in Chapter 6

Calculating Percentages over All Groups

Implementation

- Define the variable that you want to compute percentages for as an analysis variable used to calculate the sum statistic (the default for numeric variables).
- Include the analysis variable in the COLUMN statement.
- Create a compute block for the beginning of the report. In this compute block, store the SUM statistic for the analysis variable in a DATA step variable.
- Create a report variable for the percentages that you are calculating.
 - Define this variable as a computed variable with the appropriate format.
 - Include this variable in the COLUMN statement.
 - Create a compute block for this variable. In the assignment statement that calculates the percentage variable, use the compound name of the analysis variable as the numerator and the DATA step variable that stores the SUM statistic as the denominator.

Report

In this report the computed variable PCTCOMP is the percentage of the company's budget that is assigned to the current row of the report.

Calculating Percentages over All Groups				1
DEPT	ACCOUNT	BUDGET	PCTCOMP	
Equipment	lease	\$80,000.00	10.3%	
	maint	\$22,000.00	2.8%	
	purchase	\$60,000.00	7.7%	
	rental	\$10,000.00	1.3%	
	sets	\$15,000.00	1.9%	
	tape	\$20,000.00	2.6%	
Equipment		\$207,000.00	26.7%	
Facilities	rent	\$48,000.00	6.2%	
	supplies	\$5,500.00	0.7%	
	utils	\$8,500.00	1.1%	
Facilities		\$62,000.00	8.0%	
Other	advert	\$60,000.00	7.7%	
	musicfee	\$8,000.00	1.0%	
	talent	\$33,000.00	4.3%	
Other		\$101,000.00	13.0%	
Staff	fulltime	\$295,000.00	38.1%	
	parttime	\$100,000.00	12.9%	
Staff		\$395,000.00	51.0%	
Travel	gas	\$2,000.00	0.3%	
	leases	\$8,000.00	1.0%	
Travel		\$10,000.00	1.3%	
		\$775,000.00	100.0%	

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget pctcomp;

  define dept      / group;
  define account  / group;
  define budget    / analysis sum;
  define pctcomp   / computed format=percent8.1;

  compute before;
    compbud=budget.sum;
  endcomp;

  compute pctcomp;
    pctcomp=budget.sum/compbud;
  endcomp;

  break after dept / skip summarize;

  rbreak after / summarize;

  title 'Calculating Percentages over All Groups';
run;

```

Program Notes

- The DEFINE statement for BUDGET defines it as an analysis variable used to calculate the SUM statistic.
- The DEFINE statement for PCTCOMP defines it as a computed variable and assigns it the PERCENT8.1 format.
- The COLUMN statement includes the analysis variable BUDGET and the computed variable PCTCOMP.
- The assignment statement in the compute block at the beginning of the report stores the value of BUDGET.SUM in COMPBUD. This assignment statement executes only once for the entire report, just before PROC REPORT constructs the first row of the report. Here, BUDGET.SUM is the budget for the whole company.
- The assignment statement in the compute block for PCTCOMP calculates the percentage of the company's budget assigned to the current row. This assignment statement executes for each row of the report. In detail rows, PCTCOMP is the percentage assigned to an individual account. In summary rows PCTCOMP is the percentage assigned to a whole department.

You do not need to multiply PCTCOMP by 100 if you use the PERCENT. format to write the percentage.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Adding a Variable Computed from a Statistic in Chapter 5
- Specifying Formats in Chapter 3
- Inserting a Blank Line between Groups in Chapter 6
- Summarizing Variables and Statistics for Groups in Chapter 6
- Summarizing Variables and Statistics for the Whole Report in Chapter 7

Calculating Cumulative Percentages within Groups

Implementation

- Calculate percentages within the group.
- Define a variable for the cumulative percentages that you are calculating. Include this variable in the COLUMN statement.
- Use DATA step programming to accumulate the percentages in a group (see the sample code).

Report

In this report the computed variable CUMPCT is the cumulative percentage of the departmental budget for each account.

Calculating Cumulative Percentages within Groups					1
DEPT	ACCOUNT	BUDGET	PCTDEPT	CUMPCT	
Equipment	lease	\$80,000.00	38.6%	38.6%	
	maint	\$22,000.00	10.6%	49.3%	
	purchase	\$60,000.00	29.0%	78.3%	
	rental	\$10,000.00	4.8%	83.1%	
	sets	\$15,000.00	7.2%	90.3%	
	tape	\$20,000.00	9.7%	100.0%	
Facilities	rent	\$48,000.00	77.4%	77.4%	
	supplies	\$5,500.00	8.9%	86.3%	
	utils	\$8,500.00	13.7%	100.0%	
Other	advert	\$60,000.00	59.4%	59.4%	
	musicfee	\$8,000.00	7.9%	67.3%	
	talent	\$33,000.00	32.7%	100.0%	
Staff	fulltime	\$295,000.00	74.7%	74.7%	
	parttime	\$100,000.00	25.3%	100.0%	
Travel	gas	\$2,000.00	20.0%	20.0%	
	leases	\$8,000.00	80.0%	100.0%	

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget pctdept cumpct;

  define dept      / group;
  define account  / group;
  define pctdept  / computed format=percent8.1;
  define cumpct   / computed format=percent8.1;

  compute before dept;
    deptbud=budget.sum;
    cume=0;
  endcomp;

  compute pctdept;
    pctdept=budget.sum/deptbud;
    if account ne ' ' then cume+pctdept;
  endcomp;

  compute cumpct;
    cumpct=cume;
  endcomp;

  break after dept / skip;

  title 'Calculating Cumulative Percentages within Groups';
run;

```

Program Notes

- The DEFINE statement for CUMPCT defines it as a computed variable and assigns it the PERCENT8.1 format. You do not need to multiply CUMPCT by 100 in you use the PERCENT. format to write the percentage.
- The assignment statement in the compute block before DEPT assigns CUME a value of 0. This assignment statement executes only once for each group, just before PROC REPORT starts to write the rows for a group. CUME is a DATA step variable. In the compute blocks for PCTDEPT and CUMPCT, it is used to calculate cumulative percentages.
- The assignment statement in the compute block for PCTDEPT calculates the percentage of the department's budget assigned to each account. This statement executes for each row of the report. The sum statement in the compute block for PCTDEPT adds this amount to CUME to calculate the cumulative percentage. This statement executes only for detail rows of the report. (In detail rows ACCOUNT does not have a missing value.)
 When you create a compute block at a break (COMPUTE BEFORE DEPT), PROC REPORT calculates the corresponding default summary line and processes it as part of the report even if the summary line does not appear in the report. Therefore, you must restrict execution of the sum statement in the compute block for PCTDEPT to the detail rows of the report. If you do not, PROC REPORT adds the percentages on the summary lines to the cumulative percentages.

- The assignment statement in the compute block for CUMPCT assigns the current value of CUME to CUMPCT.

Note: You cannot calculate CUMPCT directly as the sum of PCTDEPT and CUME because CUMPCT is a report variable. PROC REPORT initializes report variables to 0 at the beginning of each row of the report. To retain the cumulative values, you must use a DATA step variable.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Adding a Variable Computed from a Statistic in Chapter 5
- Specifying Formats in Chapter 3
- Inserting a Blank Line between Groups in Chapter 6
- Initializing Variables for Each Group in Chapter 6
- Calculating Percentages within Groups

Calculating Cumulative Percentages for the Whole Report

Implementation

- Calculate percentages over all groups.
- Create a report variable for the cumulative percentages that you are calculating. Include this variable in the COLUMN statement.
- Use DATA step programming to accumulate the percentages as you build the report (see the sample code).

Report

In this report the computed variable CUMPCT is the cumulative percentage of the company budget for each account.

Calculating Cumulative Percentages for the Whole Report					1
DEPT	ACCOUNT	BUDGET	PCTCOMP	CUMPCT	
Equipment	lease	\$80,000.00	10.3%	10.3%	
	maint	\$22,000.00	2.8%	13.2%	
	purchase	\$60,000.00	7.7%	20.9%	
	rental	\$10,000.00	1.3%	22.2%	
	sets	\$15,000.00	1.9%	24.1%	
	tape	\$20,000.00	2.6%	26.7%	
Facilities	rent	\$48,000.00	6.2%	32.9%	
	supplies	\$5,500.00	0.7%	33.6%	
	utils	\$8,500.00	1.1%	34.7%	
Other	advert	\$60,000.00	7.7%	42.5%	
	musicfee	\$8,000.00	1.0%	43.5%	
	talent	\$33,000.00	4.3%	47.7%	
Staff	fulltime	\$295,000.00	38.1%	85.8%	
	parttime	\$100,000.00	12.9%	98.7%	
Travel	gas	\$2,000.00	0.3%	99.0%	
	leases	\$8,000.00	1.0%	100.0%	

Sample Program

```

proc report data=budget2 nowindows;
  column dept account budget pctcomp cumpct;

  define dept      / group;
  define account   / group;
  define budget    / analysis sum;
  define pctcomp   / computed format=percent8.1;
  define cumpct    / computed format=percent8.1;

  compute before;
    compbud=budget.sum;
    cume=0;
  endcomp;

  compute pctcomp;
    pctcomp=budget.sum/compbud;
    if account ne ' ' then cume+pctcomp;
  endcomp;

  compute cumpct;
    cumpct=cume;
  endcomp;

  break after dept / skip;

  title 'Calculating Cumulative Percentages for the Whole Report';
run;

```

Program Notes

- The DEFINE statement for CUMPCT defines it as a computed variable and assigns it the PERCENT8.1 format.
- The assignment statement in the compute block for the beginning of the report assigns CUME a value of 0. This assignment statement executes only once for the whole report, just before PROC REPORT starts to write the first row of the report. CUME is a DATA step variable. In the compute blocks for PCTCOMP and CUMPCT, it is used to calculate cumulative percentages.
- The assignment statement in the compute block for PCTCOMP calculates the percentage of the company's budget assigned to each account. This statement executes for every row of the report. The sum statement adds this amount to CUME to calculate the cumulative percentage. This statement executes only on detail rows of the report. (In detail rows ACCOUNT does not have a missing value.)

When you create a compute block at a break (COMPUTE BEFORE DEPT), PROC REPORT calculates the corresponding default summary line and processes it as part of the report even if the summary line does not appear in the report. Therefore, you must restrict execution of the sum statement in the compute block for PCTCOMP to the detail rows of the report. If you do not, PROC REPORT adds the percentage on the summary lines to the cumulative percentage.

- The assignment statement in the compute block for CUMPCT assigns the current value of CUME to CUMPCT.

Note: You cannot calculate CUMPCT directly as the sum of PCTCOMP and CUME because CUMPCT is a report variable. PROC REPORT initializes report variables to 0 at the beginning of each row of the report. To retain the cumulative values, you must use a DATA step variable.

Building Blocks

- Specifying and Ordering Columns to Display in Chapter 1
- Consolidating Multiple Observations into One Row in Chapter 1
- Adding a Variable Computed from a Statistic in Chapter 5
- Specifying Formats in Chapter 3
- Inserting a Blank Line between Groups in Chapter 6
- Initializing Variables for Each Group in Chapter 6
- Controlling Whether Statements in a Compute Block Execute at the Beginning and End of a Report in Chapter 7
- Calculating Percentages over All Groups

Chapter 9 The REPORT Language

<i>Overview</i>	180
<i>REPORT Procedure Syntax</i>	180
<i>PROC REPORT Statement</i>	181
<i>BREAK Statement</i>	188
<i>Order of Break Lines</i>	191
<i>BY Statement</i>	191
<i>Requirements for BY Processing</i>	192
<i>CALL DEFINE Statement</i>	192
<i>COLUMN Statement</i>	193
<i>COMPUTE Statement</i>	198
<i>How to Reference Data Set Variables in a Compute Block</i>	199
<i>Compute Block Processing</i>	200
<i>DEFINE Statement</i>	201
<i>Interactions of Position and Usage</i>	211
<i>ENDCOMP Statement</i>	214
<i>FREQ Statement</i>	214
<i>LINE Statement</i>	215
<i>Differences between the LINE and PUT Statements</i>	216
<i>RBREAK Statement</i>	217
<i>Order of Break Lines</i>	219
<i>WEIGHT Statement</i>	220
<i>Examples</i>	220
<i>Specifying and Ordering Columns to Display with the COLUMN Statement</i>	221
<i>Ordering Rows with the ORDER and ORDER= Options</i>	222
<i>Grouping Observations with the GROUP Option and an Analysis Variable</i>	223
<i>Creating Column Headers from Variable Values with the ACROSS Option</i>	224
<i>Printing All Values for a Long Row on Consecutive Lines with the WRAP Option</i>	225
<i>Creating a Group for Missing Values with the MISSING Option</i>	226
<i>Customizing Column Headers</i>	228
<i>Making Multipanel Reports with the PANELS= Option</i>	229
<i>Adding Variables to a Report with a COMPUTE Block</i>	230
<i>Creating Default Group Summaries with the BREAK Statement</i>	231
<i>Creating a Default Report Summary with the RBREAK Statement</i>	233
<i>Creating Customized Group Summaries with the COMPUTE Statement</i>	234
<i>Wrapping the Value of a Character Variable across Multiple Rows</i>	236
<i>Suppressing the Display of an Item with the NOPRINT Option</i>	237
<i>Writing a Report for Use in a Windowing Environment</i>	238

Overview

Earlier chapters in this book are task-oriented. They explain how to use PROC REPORT to accomplish specific tasks like underlining column headers or creating customized summaries. This chapter is software-oriented. It provides detailed information about the individual pieces of the report language.

The first part of this chapter presents the high-level syntax for the REPORT procedure. An individual section on each statement describes the statement in detail, explaining its purpose and how to use it. Finally, the examples show how to use many of the features of the REPORT procedure.

REPORT Procedure Syntax

The REPORT procedure uses the following statements:

```

PROC REPORT <report-option(s)>;
BREAK location break-variable </ break-option(s)>;
COLUMN column-specification(s) ;
COMPUTE location < break-variable >;
    LINE specification(s);
    ENDCOMP;
COMPUTE item </ type-specification >;
    CALL DEFINE (column-id, 'attribute-name', value);
    ENDCOMP;
DEFINE item /<usage >
    < attribute(s) >
    < option(s) >
    < justification >
    <'column-header-1' < . . . 'column-header-n'>;
RBREAK location </ break-option(s)>;

BY <DESCENDING> variable-1 < . . . <DESCENDING> variable-n <NOTSORTED>;
FREQ variable;
WEIGHT variable;

```

- The PROC REPORT statement is required.
- The following statements are optional:

BREAK	BY
COLUMN	FREQ
DEFINE	WEIGHT
RBREAK	

- The following statements are optional, but each time you use one, you must use the other:

```

COMPUTE
ENDCOMP

```

The COMPUTE statement must precede the ENDCOMP statement. For

information on the statements and functions you can use between a COMPUTE and an ENDCOMP statement, see “COMPUTE Statement” later in this chapter.

- The LINE statement is optional. Use it only after a COMPUTE statement that includes the *location* argument.
- The CALL DEFINE statement is optional. Use it only after a COMPUTE statement that includes the *item* argument.

PROC REPORT Statement

The PROC REPORT statement invokes the REPORT procedure. Optionally, the statement specifies the SAS data set and report definition to use for the report. The statement can include a number of options that control the layout of the report and the behavior of the REPORT procedure.

Syntax

The general form of the PROC REPORT statement is

PROC REPORT <*report-option(s)*> ;

- *Report-option(s)* can be one or more of the following:

CENTER NOCENTER	PANELS= <i>number-of-panels</i>
COLWIDTH= <i>column-width</i>	PROFILE= <i>libref.catalog</i>
DATA= <i>SAS-data-set</i>	PS= <i>page-size</i>
HEADLINE	PSPACE= <i>space-between-panels</i>
HEADSKIP	REPORT= <i>libref.catalog.entry</i>
LIST	SHOWALL
LS= <i>line-size</i>	SPACING= <i>space-between-columns</i>
MISSING	SPLIT= <i>'character'</i>
NAMED	WINDOWS NOWINDOWS
NOHEADER	WRAP
OUTREPT= <i>libref.catalog.entry</i>	

Options

The PROC REPORT statement supports the following options:

CENTER|NOCENTER

specifies whether or not to center the report and summary text (customized break lines) between the left and right margins. Customized break lines can appear between groups, at the beginning of a report, or at the end of a report (see “LINE Statement” later in this chapter).

You can control the centering of a report from the PROC REPORT statement, from a stored report definition, or from the setting of the SAS system option CENTER. The REPORT procedure first honors the setting in the PROC REPORT statement. If you do not specify how to center the report there, PROC REPORT honors the setting of the CENTER option in a report definition loaded with the REPORT= option in the PROC

REPORT statement. If you use neither a procedure statement option nor a stored report definition, PROC REPORT honors the setting of the CENTER system option.

Note: When the CENTER option is in effect, PROC REPORT ignores spacing that precedes the leftmost variable in the report. When the NOCENTER option is in effect, the procedure honors spacing that precedes the leftmost variable.

COLWIDTH=*column-width*

specifies the default number of horizontal positions for columns containing computed variables or numeric data set variables. *column-width* can range from 1 to the line size. The default is COLWIDTH=9.

When setting the width for a column, PROC REPORT first looks at the WIDTH= option in the DEFINE statement for that column. If the WIDTH= option is not present, PROC REPORT uses a column width large enough to accommodate the format for the item. (For information on formats see the discussion of the FORMAT= option in “DEFINE Statement” later in this chapter.)

If no format is associated with the item, the column width depends on variable type:

If the variable is a ...	then the column width is the ...
character variable in the input data set	length of the variable
numeric variable in the input data set	the value of the COLWIDTH= option
computed variable (numeric or character)	the value of the COLWIDTH= option

DATA=*SAS-data-set*

specifies the input data set. If you omit the DATA= option, PROC REPORT uses the data set whose name is stored in the automatic variable _LAST_. This data set is usually the data set most recently created within your SAS session, although you can set the value of _LAST_ yourself with the SAS system option _LAST_=.

HEADLINE

underlines all column headers and the spaces between them at the top of each page of the report.

The HEADLINE option underlines with the second character in the string defined by the SAS system option FORMCHAR=. By default, the second character is a hyphen (-). Use the FORMCHAR= system option to change it to another character.

HEADSKIP

writes a blank line beneath all column headers (or beneath the underlining that the HEADLINE option writes) at the top of each page of the report.

LIST

writes to the SAS log a listing of the REPORT definition created by the PROC REPORT step you submit. The listing may differ in these ways from the statements that you submit:

- It shows some defaults that you may not have specified.
- It omits some statements that are not specific to the REPORT procedure, whether you submit them with the PROC REPORT step or whether they are already in

effect when you submit the step. These statements include

- BY
- FOOTNOTE
- FREQ
- TITLE
- WEIGHT
- WHERE

- It does not include these PROC REPORT statement options:
 - DATA=
 - LIST
 - OUTREPT=
 - PROFILE=
 - REPORT=
 - WINDOWS|NOWINDOWS

- It does not include SAS system options.

LS=*line-size*

specifies the length of a line of the report. *line-size* is an integer from 64 to 256. You can control the line size of a report from the PROC REPORT statement, from a stored report definition, or from the SAS system option LINESIZE=. The REPORT procedure first honors the setting in the PROC REPORT statement. If you do not specify a line size for the report there, PROC REPORT honors the linesize stored in a report definition that you load with the REPORT= option in the PROC REPORT statement. If you use neither a procedure statement option nor a stored report definition, PROC REPORT honors the setting of the LINESIZE= system option.

MISSING

considers missing values as valid values for group, order, or across variables. Special missing values used to represent numeric values (the letters A through Z and the underscore (_) character) are each considered as a different value. A group for each missing value appears in the report. If you omit the MISSING option, PROC REPORT does not include observations with a missing value for any group, order, or across variables in the report. For an example of the MISSING option, see “Creating a Group for Missing Values with the MISSING Option” later in this chapter.

NAMED

writes *name=* in front of each value in the report, where *name* is the column header for the value. You may find it useful to specify the NAMED option in conjunction with the WRAP option to produce a report that wraps all columns for a single row of the report onto consecutive lines rather than placing columns of a wide report on separate pages. When you use the NAMED option, PROC REPORT automatically uses the NOHEADER option.

For an example of a report that uses the NAMED option, see “Printing All Values for a Long Row on Consecutive Lines with the WRAP Option” later in this chapter.

NOHEADER

suppresses column headers, including those that span multiple columns.

OUTREPT=*libref.catalog.entry*

stores in the specified entry the REPORT definition defined by the PROC REPORT step you submit. The stored report definition may differ in these ways from the statements that you submit:

- It shows some defaults that you may not have specified.
- It omits some statements that are not specific to the REPORT procedure, whether you submit them with the PROC REPORT step or whether they are already in effect when you submit the step. These statements include
 - BY
 - FOOTNOTE
 - FREQ
 - TITLE
 - WEIGHT
 - WHERE
- It does not include these PROC REPORT statement options:
 - DATA=
 - LIST
 - OUTREPT=
 - PROFILE=
 - REPORT=
 - WINDOWS|NOWINDOWS
- It does not include SAS system options.

When you use the OUTREPT= option, specify the following:

- | | |
|----------------|--|
| <i>libref</i> | points to a SAS data library. Define the libref with the LIBNAME statement before invoking the REPORT procedure. For details on the LIBNAME statement, see Chapter 9, “SAS Language Statements,” in <i>SAS Language: Reference</i> . |
| <i>catalog</i> | specifies or creates a SAS catalog in <i>libref</i> . |
| <i>entry</i> | specifies the entry to which to write the report definition. |

The SAS System assigns an entry type of REPT to the entry.

PANELS=*number-of-panels*

specifies the number of panels on each page of the report. If the width of a report is less than half of the line size, you can display the data in multiple sets of columns so that rows that would otherwise appear on multiple pages appear on the same page. Each set of columns is a *panel*. A familiar example of this kind of report is a telephone book,

which contains multiple panels of names and telephone numbers on a single page.

When writing a report with multiple panels, PROC REPORT fills one panel before beginning the next.

By default, the REPORT procedure creates a report with one panel per page. If *number-of-panels* is larger than the number of panels that can fit on the page, PROC REPORT creates as many panels as it can. The number of panels that fits on a page depends on the

- width of the panel
- space between panels
- linesize.

For information on the space between panels and the linesize, see the discussions of the PSPACE= and LS= options in this section.

For an example of a report that uses multiple panels, see “Making Multipanel Reports with the PANELS= Option” later in this chapter.

PROFILE=*libref.catalog*

locates your REPORT profile. In a profile, you can specify whether you want the windowing or the nonwindowing environment as your default environment. When you use the PROFILE= option, specify the following:

- libref* points to a SAS data library. Define the libref with the LIBNAME statement before invoking the REPORT procedure. For details on the LIBNAME statement, see *SAS Language: Reference*.
- catalog* specifies the catalog that contains your profile.

PROC REPORT uses the entry REPORT.PROFILE in the catalog you specify as your profile. If no such entry exists, or if you do not specify a profile, PROC REPORT uses the profile in SASUSER.PROFILE. If you have no profile and you specify neither WINDOWS nor NOWINDOWS in the PROC REPORT statement, PROC REPORT looks at the setting of the SAS system option DMS. If DMS is ON, PROC REPORT uses the windowing environment; if DMS is OFF, it uses the nonwindowing environment.

You create a profile from the PROFILE window while using PROC REPORT in a windowing environment. To create a profile, follow these steps:

1. Invoke PROC REPORT with the WINDOWS option.
2. Select Profile from the Locals pull-down menu.
3. Set the WINDOWS option appropriately.
4. Select OK to exit the PROFILE window. When you exit the window, PROC REPORT stores the profile in SASUSER.PROFILE.REPORT.PROFILE. Use the CATALOG procedure or the CATALOG window if you want to copy the profile to another location.

Note: If you open the PROFILE window and decide not to create a profile, select CANCEL to close the window.

PS=*page-size*

specifies the number of lines in a page of the report. *pagesize* is an integer from 15 to 32,767. You can control the page size of a report from the PROC REPORT statement,

from a stored report definition, or from the setting of the SAS system option PAGESIZE=. The REPORT procedure first honors the setting in the PROC REPORT statement. If you do not specify a page size for the report there, PROC REPORT honors the pagesize stored in a report definition that you load with the REPORT= option in the PROC REPORT statement. If you use neither a procedure statement option nor a stored report definition, PROC REPORT honors the PAGESIZE= system option.

PSPACE=*space-between-panels*

specifies the number of blank characters between panels. The default is PSPACE=4.

The REPORT procedure separates all panels in the report by the same number of blank characters. For each panel, the sum of its width and the number of blank characters separating it from the panel to its left cannot exceed the line size.

REPORT=*libref.catalog.entry*

specifies the report definition to use. The REPORT procedure stores all report definitions as entries of type REPT in a SAS catalog. When you use the REPORT= option, specify the following:

libref points to a SAS data library. Define the libref with the LIBNAME statement before invoking the REPORT procedure. For details on the LIBNAME statement, see *SAS Language: Reference*.

catalog specifies a SAS catalog in *libref*.

entry specifies the entry that contains the report definition you want.

Note: If you use the REPORT= option, you cannot use the COLUMN statement.

For information on storing report definitions see the discussion of the OUTREPT= option earlier in this section.

SHOWALL

overrides options in the DEFINE statement that suppress the display of a column. See the discussions of the NOPRINT and NOZERO options in “DEFINE Statement” later in this chapter.

SPACING=*space-between-columns*

specifies the number of blank characters between columns. The default is SPACING=2.

The REPORT procedure separates all columns in the report by the number of blank characters specified by the SPACING= option in the PROC REPORT statement unless you use the SPACING= option in the DEFINE statement to change the spacing to the left of a specific item. For each column, the sum of its width and the blank characters between it and the column to its left cannot exceed the line size.

Note: When the CENTER option is in effect, PROC REPORT ignores spacing that precedes the leftmost variable in the report. When the NOCENTER option is in effect, the procedure honors spacing that precedes the leftmost variable.

SPLIT=*'character'*

specifies the split character. If you use the split character in a column header, the REPORT procedure breaks the header when it reaches that character and continues the header on the next line. The split character itself is not part of the column header. The default is SPLIT='/

The FLOW option in the DEFINE statement honors the split character.

WINDOWS | NOWINDOWS
WD | NOWD

selects a windowing or nonwindowing environment.

When you use the **WINDOWS** option, the SAS System opens the **REPORT** window, which enables you to modify a report repeatedly and to see the modifications immediately. When you use the **NOWINDOWS** option, **PROC REPORT** runs without the **REPORT** window and sends its output to the SAS procedure output.

If you do not specify **WINDOWS** or **NOWINDOWS** in the **PROC REPORT** statement, the procedure honors the setting of the option in your report profile. If you do not have a report profile, **PROC REPORT** looks at the setting of the SAS system option **DMS**. If **DMS** is **ON**, **PROC REPORT** uses the windowing environment; if **DMS** is **OFF**, it uses the nonwindowing environment.

WRAP

displays one value from each column of the report, on consecutive lines if necessary, before displaying another value from the first column. By default, **PROC REPORT** displays values for only as many columns as it can fit on one page. It fills a page with values for these columns before starting to display values for the remaining columns on the next page.

For an example of a report that uses the **WRAP** and **NAMED** options, see “Printing All Values for a Long Row on Consecutive Lines with the **WRAP** Option” later in this chapter.

Note: Typically, you use the **WRAP** option in conjunction with the **NAMED** option to avoid wrapping column headers.

Information for the Windowing Environment

- If you use the **NOHEADER** option, users in a windowing environment cannot select any report items.
- The **LIST** option has no effect in the windowing environment. The **RSOURCE** command writes a listing of the current report definition to the **SOURCE** window and displays that window.
- When you use the **PROFILE** window to create a report profile, you can customize some aspects of the **REPORT** procedure that apply only in a windowing environment.
- The **PROC REPORT** statement supports these options in a windowing environment:
 - **COMMAND**
 - **HELP=**
 - **NORKEYS**
 - **PROMPT**

For more information, see *SAS Guide to the REPORT Procedure*.

BREAK Statement

The BREAK statement produces a default summary at a break (a change in the value of a group or order variable). The *break lines* that make up a default summary can

- visually separate observations
- summarize statistics and computed variables for multiple observations. A break line that summarizes information is a *summary line*.

The information in a summary applies to a set of observations with a unique combination of values for the break variable and all other group or order variables to the right of the break variable in the report.

The break variable must be a group or order variable. You cannot have both group and order variables in the same report. For an explanation of break variables see “Requirements” later in this section.

For illustrations of the BREAK statement and its options, see “Creating Default Group Summaries with the BREAK Statement” later in this chapter.

Syntax

The general form of the BREAK statement is

BREAK *location break-variable* < / *break-option(s)* >;

- *location* is one of the following:

AFTER

BEFORE

- *break-option(s)* can be one or more of the following:

DOL

DUL

OL

PAGE

SKIP

SUMMARIZE

SUPPRESS

UL

Requirements

You must specify the following arguments in the BREAK statement:

location

controls the placement of the break lines and is either

AFTER places the break lines after the last row of each set of rows that have the same value for the break variable.

BEFORE places the break lines before the first row of each set of rows that have the same value for the break variable.

break-variable

is a group or order variable. The REPORT procedure writes break lines each time the value of this variable changes.

Options

The BREAK statement supports the following options:

DOL

(for double overlining) uses the thirteenth character in the string defined by the SAS system option FORMCHAR= to overline each value

- that appears in the summary line
- that would appear in the summary line if you specified the SUMMARIZE option.

By default, the thirteenth character in the FORMCHAR= string is an equals sign (=). Use the FORMCHAR= system option to change it to another character.

If you specify both the OL and DOL options, PROC REPORT honors only OL.

DUL

(for double underlining) uses the thirteenth character in the string defined by the SAS system option FORMCHAR= to underline each value

- that appears in the summary line
- that would appear in the summary line if you specified the SUMMARIZE option.

By default, the thirteenth character in the FORMCHAR= string is an equals sign (=). Use the FORMCHAR= system option to change it to another character.

If you specify both the UL and DUL options, PROC REPORT honors only UL.

OL

(for overlining) uses the second character in the string defined by the SAS system option FORMCHAR= to overline each value

- that appears in the summary line
- that would appear in the summary line if you specified the SUMMARIZE option.

By default, the second character in the FORMCHAR= string is a hyphen (-). Use the FORMCHAR= system option to change it to another character.

If you specify both the OL and DOL options, PROC REPORT honors only OL.

PAGE

starts a new page after the last line of each group of break lines.

SKIP

writes a blank line after the last line of each group of break lines.

SUMMARIZE

writes a summary line in each group of break lines. A summary line for a set of observations contains values for

- the break variable (which you can suppress with the SUPPRESS option)
- group or order variables to the left of the break variable in the report
- statistics
- analysis variables
- computed variables.

The following table shows how PROC REPORT calculates the value for each kind of report item in a summary line created by the BREAK statement:

If the report item is ...	then its value is ...
the break variable	the current value of the variable (or a missing value if you use the SUPPRESS option)
group, order, or display variables to the left of the break variable	the current value of the variable
group, order, or display variables to the right of the break variable	missing
a statistic	the value of the statistic over all observations in the set
an analysis variable	the value of the statistic specified as the usage option in the DEFINE statement. PROC REPORT calculates the value of the statistic over all observations in the set. The default usage is SUM.
a computed variable	the results of the calculations based on the code in the corresponding compute block (see “COMPUTE Statement” later in this chapter.)

Note: You cannot use group and order variables in the same report. You cannot use group and display variables in the same report.

SUPPRESS

suppresses printing of

- the value of the break variable in the summary line
- any underlining and overlining in the break lines in the column containing the break variable.

Note: If you use the SUPPRESS option, the value of the break variable is unavailable for use in customized break lines unless you assign it a value in the COMPUTE block associated with the break (see “COMPUTE Statement” later in this chapter).

UL

(for underlining) uses the second character in the string defined by the SAS system option FORMCHAR= to underline each value

- that appears in the summary line
- that would appear in the summary line if you specified the SUMMARIZE option.

By default, the second character in the FORMCHAR= string is a hyphen -. Use the FORMCHAR= system option to change it to another character.

If you specify both the UL and DUL options, PROC REPORT honors only UL.

Order of Break Lines

When a default summary contains more than one break line, the order in which the break lines appear is

1. overlining or double overlining (OL or DOL)
2. summary line (SUMMARIZE)
3. underlining or double underlining (UL or DUL)
4. skipped line (SKIP)
5. page break (PAGE).

Note: If you define a customized summary for the break, customized break lines appear after underlining or double underlining. For more information on customized break lines, see “COMPUTE Statement” and “LINE Statement” later in this chapter.

Information for the Windowing Environment

The BREAK statement supports the COLOR= option in the windowing environment. For more information, see *SAS Guide to the REPORT Procedure*.

BY Statement

The BY statement creates a separate report for each BY group. A BY group is a set of observations with the same values for all BY variables.

Syntax

The general form of the BY statement is

```
BY <DESCENDING> variable-1 < . . . <DESCENDING> variable-n <NOTSORTED>;
```

Requirements for BY Processing

When you use a BY statement, the REPORT procedure expects the input data set to be sorted in order of the BY variables or to have an appropriate index. If your input data set is not sorted in ascending order, you can

- Use the SORT procedure with a similar BY statement to sort the data.
- If appropriate, use one of the BY statement options, NOTSORTED or DESCENDING.
- Create an index on the BY variables you want to use. For more information on creating indexes and using the BY statement with indexed data sets, see Chapter 17, “The DATASETS Procedure,” in the *SAS Procedures Guide*.

Note: Using the BY statement does not make the FIRST. and LAST. variables available in compute blocks.

Information for the Windowing Environment

You cannot use the BY statement in the windowing environment.

CALL DEFINE Statement

You can use the CALL DEFINE statement in a COMPUTE block to set the value of an attribute for a particular value in a column. Use this statement to write report definitions that other people will use in a windowing environment. None of the attributes has an affect in the nonwindowing environment. (See “Requirements” later in this section for a table describing available attributes.)

CALL DEFINE statements are valid only in compute blocks for which the COMPUTE statement contains the *item* argument.

Syntax

The general form of the CALL DEFINE statement is

```
CALL DEFINE (column-id, 'attribute-name', value);
```

Requirements

You must use the following arguments in the CALL DEFINE statement:

column-id

identifies the column in which to define the attribute. *column-id* specifies a column name or a column number. A column name can be

- a character literal (in quotation marks)
- a character expression.

A column number can be

- a numeric literal
- a numeric expression
- a name of the form `_Cn_`, where *n* is the column number.

You can use the automatic variable `_COL_` to identify the current column. The current column is the column containing the report item named in the COMPUTE statement.

attribute-name

specifies the attribute to define. For attribute names, refer to Table 9.1.

value

sets the value for the attribute. For values for each attribute, refer to Table 9.1.

Table 9.1 Attribute Descriptions

Attribute	Description	Values
blink	Controls blinking of current value.	1 turns blinking on; 0 turns it off.
color	Controls the color of the current value.	'blue', 'red', 'pink', 'green', 'cyan', 'yellow', 'white', 'orange', 'black', 'magenta', 'gray', 'brown'
command	Specifies that a series of commands follows	a quoted string of valid SAS commands to submit to the command line
highlight	Controls highlighting of current value.	1 turns highlighting on; 0 turns it off.
rsvsideo	Controls display of the current value.	1 turns reverse video on; 0 turns it off.

You can attach a compute block to any type of column. PROC REPORT executes the block once for each row of the report, as it fills in the value for that column. If you change a column attribute, the change applies only to the specified column on the current row. For an example of using the CALL DEFINE statement with the COMMAND attribute, see “Writing a Report for Use in a Windowing Environment” later in this chapter.

Note: The attributes BLINK, HIGHLIGHT, and RVSVIDEO do not work on all devices.

COLUMN Statement

The COLUMN statement describes the arrangement of columns and of headers that span more than one column in the report. You can use any number of COLUMN statements. If you use more than one COLUMN statement, the procedure simply concatenates all variables from all of the COLUMN statements. The following COLUMN statement includes three variables:

```
column dept account budget;
```

It has the same effect as the following COLUMN statements:

```
column dept;
column account;
column budget;
```

Note: You cannot use the COLUMN statement if you use the REPORT= option in the PROC REPORT statement.

Syntax

The general form of the COLUMN statement is

COLUMN *column-specification(s)* ;

- *column-specification(s)* is a concatenation of one or more of the following:
 - a single report item
 - two or more report items separated by commas
 - a list of two or more report items in parentheses
 - a single report item and an alias

Requirements

You define complex report layouts by concatenating individual column specifications. The different types of column specifications are described in this section.

A Single Report Item

A column specification can be the name of a single item: a data set variable, a computed variable, or a statistic. Available statistics are

N	RANGE	T
NMISS	SUM	PRT
MEAN	USS	SUMWGT
STD	CSS	VAR
MIN	STDERR	
MAX	CV	

For example, the following COLUMN statement creates a report with three columns: one for DEPT, one for BUDGET, and one for ACTUAL:

```
column dept budget actual;
```

Two or More Report Items Separated by Commas

A column specification can be two or more items separated by commas. The REPORT procedure stacks items separated by commas one above another in the report, with the leftmost item on top. For example, the following COLUMN statement creates a report with three columns: one for department, one with the MEAN statistic below BUDGET, and one with the MEAN statistic below ACTUAL. The DEFINE statement defines DEPT as a group variable so that the REPORT procedure can calculate the MEAN statistics. The results appear in Output 9.1.

```
proc report data=budget2 nowindows;
  column dept budget,mean actual,mean;

  define dept / group;
  define budget / analysis sum;
  define actual / analysis sum;
run;
```

Output 9.1
Stacking Items with
the COLUMN
Statement

	BUDGET	ACTUAL
DEPT	MEAN	MEAN
Equipment	\$17,250	\$17,369
Facilities	\$10,333	\$10,105
Other	\$16,833	\$18,107
Staff	\$98,750	\$98,464
Travel	\$2,500	\$2,114

Note: In this program BUDGET and ACTUAL are analysis variables used to calculate the SUM statistic. However, the SUM statistics do not appear in the report. If you place statistics above or below an analysis variable, the usage option in the DEFINE statement for the analysis variable does not affect the report. The statistics you specify in the COLUMN statement are the ones that appear in the report. For more information see “Placing Statistics above or below a Variable” in Chapter 2.

A List of Two or More Report Items in Parentheses

A column specification can be a list of two or more items in parentheses. The REPORT procedure applies any item separated from a parenthesized list by a comma to each item inside the parentheses. For example, the following COLUMN statements are equivalent:

```
column dept (budget actual),mean;
column dept budget,mean actual,mean;
```

You can also use parentheses to specify headers that span multiple columns. In such a case, one or more of the items in a list is a quoted string to use as the header:

```
('header-1' < . . . 'header-n' > item-list )
```

where *header* is a string of characters that spans one or more columns in the report. The list of items following the headers within the same pair of parentheses specifies the columns to

span. The REPORT procedure prints each header on a separate line.

You can use split characters in a header to split one header over multiple lines. See the discussion of the SPLIT= option in “PROC REPORT Statement” earlier in this chapter.

If the first and last characters of a header are one of the following characters, PROC REPORT uses that character to expand the header to fill the space over the column:

– = _ . * +

If the first character of a header is < and the last character is >, or vice-versa, the REPORT procedure expands the header to fill the space over the column by repeating the first character before the text of the header and the last character after it.

For example, the following statements enhance the report in Output 9.1 by adding a two-line header that includes the date of the report (accessed by the automatic macro variable SYSDATE) above the columns for BUDGET and ACTUAL. The enhanced report is in Output 9.2.

```
proc report data=budget2 nowindows;
  column dept
    ("Report as of" "&sysdate._" budget,mean actual,mean);
  define dept / group;
run;
```

Output 9.2
Creating a Header
That Spans Multiple
Columns

			1
			Report as of 24AUG93
	BUDGET MEAN	ACTUAL MEAN	
DEPT			
Equipment	\$17,250	\$17,369	
Facilities	\$10,333	\$10,105	
Other	\$16,833	\$18,107	
Staff	\$98,750	\$98,464	
Travel	\$2,500	\$2,114	

Note: When you use a macro variable in a header, enclose the header in double rather than single quotes so that the SAS System can resolve the macro variable. Furthermore, if you use the underscore to expand the header, place a period after the name of the macro variable so that the special character does not become part of the name of the macro variable. For more information on using the SAS macro facility, see the *SAS Guide to Macro Processing*.

For more examples of the use of special characters see “Customizing Column Headers” later in this chapter and “Extending a Column Header to Fill a Column” in Chapter 4.

A Single Report Item and an Alias

You can use the same data set variable, computed variable, or statistic more than once in the same COLUMN statement. However, you can use only one DEFINE statement for any given name. Therefore, you may need to create an alternative name (an *alias*) for a variable or statistic that you use multiple times in a report. You can then use one DEFINE statement for each alias. (The DEFINE statement designates characteristics such as the usage of the item in the report and what its format is. If you do not use a DEFINE statement for an item, the REPORT procedure uses default values for the characteristics.)

The COLUMN statement used to create the report in Output 9.2 includes two uses of

the MEAN statistic, one attached to the variable BUDGET and one attached to the variable ACTUAL:

```
column dept ("Report as of" "&sysdate._"
            budget,mean actual,mean);
```

In both cases, the default characteristics are acceptable, so you do not need DEFINE statements for the MEAN statistics.

But suppose that you want to assign different formats and headers to each occurrence of the MEAN statistic. You specify headers and formats with the DEFINE statement. Because you can only have one DEFINE statement for the name MEAN, you need to create an alias for at least one occurrence of MEAN so that you can use separate DEFINE statements to specify the formats and headers.

You assign an alias in the COLUMN statement by replacing the name of an item with the following:

name =alias

where *name* is the name of the item, and *alias* is the alias to use in the DEFINE statement. For example, the following statement provides aliases for both occurrences of the MEAN statistic:

```
column dept budget,mean=budmean actual,mean=actmean;
```

Now, you can use BUDMEAN and ACTMEAN in separate DEFINE statements to assign them different characteristics. For example, the following SAS statements create a report that uses different formats and headers for the two occurrences of the MEAN statistic. The results appear in Output 9.3.

```
proc report data=budget2 nowindows;
  column dept budget,mean=budmean actual,mean=actmean;

  define dept / group;
  define budmean / format=dollar8. 'Average/Budgeted';
  define actmean / format=dollar11.2 'Average/Spent';
run;
```

Output 9.3
Using Aliases in the
COLUMN Statement

	BUDGET	ACTUAL
DEPT	Average Budgeted	Average Spent
Equipment	\$17,250	\$17,369.10
Facilities	\$10,333	\$10,104.52
Other	\$16,833	\$18,106.74
Staff	\$98,750	\$98,464.38
Travel	\$2,500	\$2,114.25

Note: Assigning an alias in the COLUMN statement does not by itself alter the report. It enables you to use separate DEFINE statements for each occurrence of a variable or statistic. Aliases that you create in a COLUMN statement are valid only in DEFINE statements. You cannot use them in compute blocks.

COMPUTE Statement

The COMPUTE statement marks the beginning of a section of programming statements that PROC REPORT executes either at a specified location or on every row of the report. An ENDCOMP statement marks the end of the group of statements. This group of statements is a *compute block*. A PROC REPORT step can contain multiple compute blocks. A compute block can

- define computed variables
- perform calculations
- create customized break lines.

Compute blocks support these SAS language features:

- DM statement
- %INCLUDE statement
- selected DATA step statements:

assignment	LENGTH
CALL	LINK
DO (all forms)	RETURN
END	SELECT
GO TO	sum
IF-THEN/ELSE	

- comments
- null statements
- macro variables
- all DATA step functions.

For information on SAS language features see *SAS Language: Reference*. Compute blocks also support these PROC REPORT features:

- Compute blocks for a customized summary support the LINE statement.
- Compute blocks for a report item support the CALL DEFINE statement.

For information on these features see “LINE Statement” later in this chapter and “CALL DEFINE Statement” earlier in this chapter.

Syntax

The general form of the COMPUTE statement is

```
COMPUTE location<break-variable>;
COMPUTE item <l type-specification>;
```

- *location* is one of the following:

AFTER
BEFORE

- *type-specification*, which indicates that the computed variable is a character variable, is either or both of the following:

CHARACTER
LENGTH=*length*

How to Reference Data Set Variables in a Compute Block

You reference data set variables in a compute block in one of three ways:

- by name.
- by a compound name that identifies both the variable and the name of the statistic you calculate with it. A compound name has this form:

variable-name.statistic

- by column number, in the form

_C_n_

where *n* is the number of the column (from left to right) in the report.

The following table shows how to use each type of reference in a compute block.

If the variable you reference is this type ...	then refer to it by ...	For example ...
group	name	DEPT
order	name	DEPT
computed	name	DEPT
display	name	DEPT
display sharing a column with a statistic	a compound name	BUDGET.SUM
analysis	a compound name	BUDGET.MEAN
any type sharing a column with an across variable	column number	_C3_

Compute Block Processing

PROC REPORT processes compute blocks in two different ways.

- If the COMPUTE statement contains a *location* argument, PROC REPORT executes the compute block only at that location.
- If the COMPUTE block contains an *item* argument, PROC REPORT executes the compute block on every row of the report when it comes to the column for that item.

Requirements

An ENDCOMP statement must follow the last programming statement.

You must specify either a location or a computed variable in the COMPUTE statement.

location

determines where PROC REPORT executes the compute block and is one of the following:

AFTER executes the compute block at a break in one of the following places:

- after the last row of a set of rows that have the same value for *break-variable*.
- at the end of the report if you do not specify a break variable.

BEFORE executes the compute block at a break in one of the following places:

- before the first row of a set of rows that have the same value for *break-variable*.
- at the beginning of the report if you do not specify a break variable.

item

is a data set variable, a computed variable, or a statistic. You must include *item* in the COLUMN statement. If *item* is a computed variable, you must include a DEFINE statement for it.

For examples of customized breaks before and after a break variable, see “Creating Customized Group Summaries with the COMPUTE Statement” later in this chapter. For examples of computing a variable see “Adding Variables to a Report with a COMPUTE Block” later in this chapter.

Note: When you use the COMPUTE statement, you do not have to use a corresponding BREAK or RBREAK statement. Use these statements only when you want to implement one or more BREAK statement or RBREAK statement options.

Options

The COMPUTE statement supports the following options:

break-variable

is a group or order variable. If you specify a location (BEFORE or AFTER) in the COMPUTE statement, you can also specify a *break-variable*. When you specify a break variable, the REPORT procedure executes the statements between the COMPUTE and ENDCOMP statements each time the value of the break variable changes.

CHARACTER

CHAR

specifies that the computed variable is a character variable. The default is numeric.

LENGTH=*length*

specifies the length of the character variable. *length* can range from 1 to 200. If you specify CHARACTER but do not specify a length, LENGTH=8.

DEFINE Statement

The DEFINE statement designates the characteristics of the specified item. These characteristics include how PROC REPORT uses the item in the report, formats the item, and chooses the column header.

Syntax

The general form of the DEFINE statement is

```
DEFINE item /<usage>
      <attribute(s)>
      <option(s)>
      <justification>
      <'column-header-1' <. . .> 'column-header-n'>;
```

□ *usage* is one of the following:

ACROSS	DISPLAY
ANALYSIS	GROUP
COMPUTED	ORDER

□ *attribute(s)* can be one or more of the following:

```
FORMAT=format
ORDER=DATA|FORMATTED|FREQ|INTERNAL
SPACING=horizontal-positions
statistic
WIDTH=column-width
```

□ *option(s)* can be one or more of the following:

DESCENDING
FLOW
NOPRINT
NOZERO
PAGE

□ *justification* is one of the following:

CENTER
LEFT
RIGHT

Requirements

You must use the following argument in the DEFINE statement:

item

a data set variable, a computed variable, a statistic, or an alias from a COLUMN statement.

Note: Do not specify a usage option in the DEFINE statement for a statistic. The name of the statistic tells PROC REPORT how to use it.

Options

The DEFINE statement supports the following options:

ACROSS

creates a column for each value of *item*, which must be a data set variable or an alias. Columns created by across variables contain statistics or computed values. If nothing is above or below an across variable, PROC REPORT displays the N statistic (the number of observations in the input data set that belong to that cell of the report).

For example, in Output 9.4, DATE is a group variable, and DEPT is an across variable. The report shows the number of observations in the data set for each department in each quarter.

Output 9.4
*An Across Variable
with No Items Above
or Below It*

DATE	Department					1
	Equipment	Facilities	Other	Staff	Travel	
31MAR92	6	3	3	2	2	
30JUN92	6	3	3	2	2	

Output 9.5 shows the same report with BUDGET added below DEPT and defined as an analysis variable used to calculate the SUM statistic. In this report the values in each column created for DEPT represent the sum of the budgets allocated to all accounts within that department for one quarter.

Output 9.5
*An Across Variable
 with an Analysis
 Variable above It*

						1
						Department
	Equipment	Facilities	Other	Staff	Travel	
DATE	BUDGET	BUDGET	BUDGET	BUDGET	BUDGET	BUDGET
31MAR92	\$109,500.00	\$31,750.00	\$46,500.00	\$170,000.00	\$4,300.00	
30JUN92	\$97,500.00	\$30,250.00	\$54,500.00	\$225,000.00	\$5,700.00	

For an example that uses an across variable see “Creating Column Headers from Variable Values with the ACROSS Option” later in this chapter.

ANALYSIS

calculates statistics for *item*, which must be a data set variable. By default, PROC REPORT calculates the SUM statistic. Specify an alternate statistic with the *statistic* option in the DEFINE statement.

Note: Naming a statistic in the DEFINE statement implies the ANALYSIS option, so you never need to specify ANALYSIS, although specifying it may make your code easier for novice users to understand. The DEFINE statements in the examples in this book always show the ANALYSIS option.

The value displayed for an analysis variable is the value of the statistic you specify calculated for the set of observations represented by that cell of the report. If the report contains one or more display or order variables, each detail row of the report represents one observation in the data set. However, if you create groups in the report, one detail row represents all observations that have a unique combination of values for all the group variables.

If you create a break in the report on a group or order variable and elect to summarize in the break, the summary row includes analysis variables. In these summary rows, the REPORT procedure calculates the statistic associated with an analysis variable over the entire set of observations that have the same value of the break variable.

For example, in Output 9.6, DEPT and ACCOUNT are group variables. BUDGET is an analysis variable used to calculate the MEAN statistic. Each detail row of the report shows the mean value of BUDGET for a unique combination of values for DEPT and ACCOUNT. For example, the first row shows that the mean value of BUDGET for the lease account in the equipment department is \$40,000.00.

The report uses DEPT as a break variable. The break lines include a summary line. Each summary line shows the mean of BUDGET for all accounts in one department.

Output 9.6
Using an Analysis
Variable with Group
Variables

DEPT	ACCOUNT	BUDGET (MEAN)
		1
Equipment	lease	\$40,000.00
	maint	\$11,000.00
	purchase	\$30,000.00
	rental	\$5,000.00
	sets	\$7,500.00
	tape	\$10,000.00
-----		-----
Equipment		\$17,250.00
-----		-----
Facilities	rent	\$24,000.00
	supplies	\$2,750.00
	utils	\$4,250.00
-----		-----
Facilities		\$10,333.33
-----		-----
Other	advert	\$30,000.00
	musicfee	\$4,000.00
	talent	\$16,500.00
-----		-----
Other		\$16,833.33
-----		-----
Staff	fulltime	\$147,500.00
	parttime	\$50,000.00
-----		-----
Staff		\$98,750.00
-----		-----
Travel	gas	\$1,000.00
	leases	\$4,000.00
-----		-----
Travel		\$2,500.00
-----		-----

Now consider using an analysis variable with order variables. In Output 9.7, DEPT and ACCOUNT are order variables. BUDGET is an analysis variable used to calculate the SUM statistic. Each detail row of the report shows the sum of BUDGET for one observation in the report. (Because the report contains one detail row for each observation in the data set, the sum of BUDGET is the same as the value of BUDGET.) Unlike the report in Output 9.6, this report does not consolidate observations for the first and second quarters that have a unique combination of values of DEPT and ACCOUNT. Therefore, this report contains twice as many detail rows, one for each account for each quarter. Although this report doesn't consolidate observations, using BUDGET as an analysis variable enables you to use BUDGET in summary rows, as shown in Output 9.7. In this report each summary row shows the sum of BUDGET for all accounts in one department.

Output 9.7
Using an Analysis
Variable with Order
Variables

DEPT	ACCOUNT	BUDGET
		1
Equipment	lease	\$40,000.00
		\$40,000.00
	maint	\$10,000.00
		\$12,000.00
	purchase	\$40,000.00
		\$20,000.00
	rental	\$4,000.00
		\$6,000.00
	sets	\$7,500.00
		\$7,500.00
	tape	\$8,000.00
		\$12,000.00
-----		-----
Equipment		\$207,000.00
-----		-----
Facilities	rent	\$24,000.00
		\$24,000.00
	supplies	\$2,750.00
		\$2,750.00
	utils	\$5,000.00
		\$3,500.00
-----		-----
Facilities		\$62,000.00
-----		-----
Other	advert	\$30,000.00
		\$30,000.00
	musicfee	\$3,000.00
		\$5,000.00
	talent	\$13,500.00
		\$19,500.00
-----		-----
Other		\$101,000.00
-----		-----
Staff	fulltime	\$130,000.00
		\$165,000.00
	parttime	\$40,000.00
		\$60,000.00
-----		-----
Staff		\$395,000.00
-----		-----
Travel	gas	\$800.00
		\$1,200.00
	leases	\$3,500.00
		\$4,500.00
-----		-----
Travel		\$10,000.00
-----		-----

For an example that uses analysis variables, see “Grouping Observations with the Group Option and an Analysis Variable” later in this chapter.

CENTER

centers the formatted values of the specified item within the column width and centers the column header over the values. Selecting this option does not alter the setting of the CENTER system option.

column-header

defines the column header for *item*. Enclose each header in single or double quotation marks. When you specify multiple column headers, PROC REPORT uses a separate line for each one. The split character also splits a column header over multiple lines. For information on the split character see the discussion of the SPLIT= option in “PROC

REPORT Statement” earlier in this chapter.

By default, PROC REPORT uses a variable’s label as its column header and the name of the statistic as the column header for a statistic. If a variable has no label, PROC REPORT uses the variable’s name. If you want to use names when labels exist, submit the following SAS statement before invoking the REPORT procedure:

```
options nolabel;
```

If the first and last characters of a header are one of the following characters, PROC REPORT uses that character to expand the header to fill the space over the column.

```
- = _ . * +
```

If the first character of a header is < and the last character is >, or vice-versa, the REPORT procedure expands the header to fill the space over the column by repeating the first character before the text of the header and the last character after it.

For examples of a variety of column headers see “Customizing Column Headers” later in this chapter.

COMPUTED

defines the specified item as a computed variable. PROC REPORT calculates the value of this variable from the compute block associated with it (see “COMPUTE Statement” later in this chapter).

For an example that uses the COMPUTED option, see “Adding Variables to a Report with a Compute Block” later in this chapter.

DESCENDING

reverses the order in which PROC REPORT displays rows or values of a group, order, or across variable. For instance, when DEPT is the leftmost order variable, the REPORT procedure, by default, displays the rows in ascending (alphabetic) order of the value of DEPT. When the characteristics for DEPT include DESCENDING, the values appear in reverse alphabetic order.

For an example that uses the DESCENDING option see “Ordering Rows with the ORDER and ORDER= Options” later in this chapter.

Note: By default, PROC REPORT orders group, order, and across variables by their formatted values. Use the ORDER= option in the DEFINE statement to specify an alternate sort order.

DISPLAY

defines *item*, which must be a data set variable, as a display variable. Display variables do not affect the order of the rows in the report. If the report contains no order variables to the left of a display variable, the order of detail rows in the report is the same as the order of observations in the data set. A report that contains one or more display variables has a detail row for each observation in the data set. Each detail row contains a value for each display variable. By default, the REPORT procedure treats all character variables as display variables.

FLOW

wraps the value of a character variable in its column. The FLOW option honors the split character. If the text contains no split character, PROC REPORT tries to split text at a blank.

For an example that uses the FLOW option, see “Wrapping the Value of a

Character Variable across Multiple Rows” later in this chapter.

FORMAT=*format*

assigns a SAS or user-defined format to the item. This format applies to *item* as PROC REPORT displays it; the format does not alter the format stored with a variable in the data set. By default, PROC REPORT uses the format you specify in the FORMAT statement when you invoke the REPORT procedure. If you do not specify a format, PROC REPORT uses the format stored in the data set. If no format for the item is stored in the data set, it uses the BEST w . format for numeric variables and the \$ w . format for character variables. The value of w is the default column width. For character variables in the input data set, the default column width is the variable’s length. For numeric variables in the input data set and for computed variables (both numeric and character), the default column width is the value of the COLWIDTH= option in the PROC REPORT statement.

For an example that uses the FORMAT= option, see “Adding Variables to a Report with a Compute Block” later in this chapter.

GROUP

enables you to consolidate data. If a report contains one or more group variables, PROC REPORT tries to consolidate into one row all observations from the data set that have a unique combination of values for all group variables. The REPORT procedure cannot consolidate observations into groups if the report contains any order variables or any display variables that do not have one or more statistics above or below them. (PROC REPORT treats a display variable with statistics above or below it as an analysis variable.) If PROC REPORT cannot create groups, it returns a message and displays group variables the same way as it displays order variables.

The GROUP option also orders the detail rows in a report according to their formatted values of *item*, which must be a data set variable. By default, the order is ascending, but you can alter it with the DESCENDING option in the DEFINE statement. If the report contains multiple group variables, the REPORT procedure establishes the order of the detail rows by sorting these variables from left to right in the report.

For example, in Output 9.8, DEPT and ACCOUNT are group variables. BUDGET is an analysis variable used to calculate the SUM statistic. Each row of this report shows the sum of the values for BUDGET for both quarters for one combination of DEPT and ACCOUNT. Detail rows are ordered according to the values of the group variables DEPT and ACCOUNT (alphabetical by department, and within a department, alphabetical by account).

Output 9.8
Creating Groups

DEPT	ACCOUNT	BUDGET
		1
Equipment	lease	\$80,000.00
	maint	\$22,000.00
	purchase	\$60,000.00
	rental	\$10,000.00
	sets	\$15,000.00
	tape	\$20,000.00
Facilities	rent	\$48,000.00
	supplies	\$5,500.00
	utils	\$8,500.00
Other	advert	\$60,000.00
	musicfee	\$8,000.00
	talent	\$33,000.00
Staff	fulltime	\$295,000.00
	parttime	\$100,000.00
Travel	gas	\$2,000.00
	leases	\$8,000.00

For an example that uses the GROUP option see “Grouping Observations with the GROUP Option and an Analysis Variable” later in this chapter.

LEFT

left-justifies the formatted values of *item* within the column width and left-justifies the column headers over the values. If the format width is the same as the width of the column, the LEFT option has no effect.

NOPRINT

suppresses the display of *item*. Use this option

- if you do not want to show the column in the report but you need to use the values in it to calculate other values you use in the report (see “Suppressing the Display of an Item with the NOPRINT Option” later in this chapter)
- to establish the order of rows in the report (see “Adding a Variable Computed from a Group, Order, Computed, or Display Variable” in Chapter 5).

The SHOWALL option in the PROC REPORT statement overrides all occurrences of the NOPRINT option.

NOZERO

suppresses the display of the selected column if its values are all zero or missing.

The SHOWALL option in the PROC REPORT statement overrides all occurrences of the NOZERO option.

ORDER

orders the detail rows in a report according to the formatted value of *item*, which must be a data set variable. By default, the order is ascending. You can alter the order with the DESCENDING option in the DEFINE statement. If the report contains multiple order variables, the REPORT procedure establishes the order of the detail rows by sorting these variables from left to right in the report. If a report contains one or more order variables, it contains a detail row for each observation in the data set.

In Output 9.9, DEPT, ACCOUNT, and DATE are all order variables. BUDGET is an analysis variable used to calculate the SUM statistic. PROC REPORT does not repeat the values of an order variable from one row to the next if the value does not change. (The value of DATE changes in every row.)

For an example that uses the ORDER option, see “Ordering Rows with the ORDER and ORDER= Options” later in this chapter.

Output 9.9
Displaying Order Variables

DEPT	ACCOUNT	DATE	BUDGET
Equipment	lease	31MAR92	\$40,000.00
		30JUN92	\$40,000.00
	maint	31MAR92	\$10,000.00
		30JUN92	\$12,000.00
	purchase	31MAR92	\$40,000.00
		30JUN92	\$20,000.00
	rental	31MAR92	\$4,000.00
		30JUN92	\$6,000.00
	sets	31MAR92	\$7,500.00
		30JUN92	\$7,500.00
	tape	31MAR92	\$8,000.00
		30JUN92	\$12,000.00
Facilities	rent	31MAR92	\$24,000.00
		30JUN92	\$24,000.00
	supplies	31MAR92	\$2,750.00
		30JUN92	\$2,750.00
utils	31MAR92	\$5,000.00	
	30JUN92	\$3,500.00	
Other	advert	31MAR92	\$30,000.00
		30JUN92	\$30,000.00
	musicfee	31MAR92	\$3,000.00
		30JUN92	\$5,000.00
talent	31MAR92	\$13,500.00	
	30JUN92	\$19,500.00	
Staff	fulltime	31MAR92	\$130,000.00
		30JUN92	\$165,000.00
	parttime	31MAR92	\$40,000.00
Travel	gas	31MAR92	\$800.00
		30JUN92	\$1,200.00
	leases	31MAR92	\$3,500.00
		30JUN92	\$4,500.00

ORDER=DATA | FORMATTED | FREQ | INTERNAL

orders the values of a GROUP, ORDER, or ACROSS variable according to the specified order, where

- DATA** orders values according to their order in the input data set.
- FORMATTED** orders values by their formatted (external) values. By default, the order is ascending.
- FREQ** orders values by ascending frequency count.
- INTERNAL** orders values by the same sequence as PROC SORT would use. This sort sequence is particularly useful for displaying dates chronologically.

The default is ORDER=FORMATTED.

► **Caution** *Default for the ORDER= Option*

In other SAS procedures, the default is ORDER=INTERNAL. The default for the option in PROC REPORT may change in a future release to be consistent with other procedures. Therefore, in production jobs where it is important to order report items by their formatted values, specify ORDER=FORMATTED even though it is currently the default. Doing so ensures that PROC REPORT will continue to produce the reports you expect even if the default changes. ▲

The DESCENDING option in the DEFINE statement reverses the sort sequence for an item.

For an example that uses the ORDER option, see “Ordering Rows with the ORDER and ORDER= Options” later in this chapter.

PAGE

inserts a page break just before printing the first column containing values of the selected item.

RIGHT

right-justifies the formatted values of the specified item within the column width and right-justifies the column headers over the values. If the format width is the same as the width of the column, RIGHT has no effect.

SPACING=*horizontal-positions*

defines the number of blank characters to leave between the selected column and the column immediately to its left. The default is SPACING=2.

Note: When the CENTER option is in effect, PROC REPORT ignores spacing that precedes the leftmost variable in the report. When the NOCENTER option is in effect, the procedure honors spacing that precedes the leftmost variable. For an example that uses the SPACING= option, see “Controlling Space between Columns” in Chapter 3.

statistic

associates a statistic with an analysis variable. You must associate a statistic with every analysis variable. If you do not use the *statistic* option in the DEFINE statement for an analysis variable, the default statistic is SUM. You cannot use the *statistic* option in a DEFINE statement for any other kind of variable.

Note: The REPORT procedure uses the name of the analysis variable as the default header for the column. You can customize the column header with the *column-header* option in the DEFINE statement.

You can use the following values for *statistic* :

N	RANGE	CV
NMISS	SUM	T
MEAN	SUMWGT	PRT
STD	USS	VAR
MIN	CSS	
MAX	STDERR	

Definitions of the statistics follow. The following notations are used where summation is over all nonmissing values:

x_i	the i th nonmissing observation on the variable.
w_i	the weight associated with x_i if a WEIGHT statement is specified, and 1 otherwise.
n	the number of nonmissing observations.
\bar{x}	$\sum w_i x_i / \sum w_i$.
s^2	$\sum w_i (x_i - \bar{x})^2 / (n - 1)$.
z_i	$(x_i - \bar{x}) / s$ standardized variables.

The statistics are

N	the number of observations with no missing values for a group, order, or across variable; or the number of nonmissing values for an analysis variable. If you use the MISSING option in the PROC REPORT statement, N includes observations with missing group, order, or across variables.
NMISS	the number of missing values for an analysis variable.
MEAN	\bar{x} , the arithmetic mean.
STD	s , the standard deviation.
MIN	the minimum value.
MAX	the maximum value.
RANGE	MAX – MIN, the range.
SUM	$\sum w_i x_i$, the weighted sum.
USS	$\sum w_i x_i^2$, the uncorrected sum of squares.
CSS	$\sum w_i (x_i - \bar{x})^2$, the sum of squares corrected for the mean.
STDERR	s / \sqrt{n} , the standard error of the mean.
SUMWGT	$\sum w_i$, the sum of weights.
CV	$100 s / \bar{x}$, the percent coefficient of variation.
T	$t = \bar{x} \sqrt{n} / s$, Student's t for H_0 : population mean=0.
PRT	the two-tailed p -value for Student's t with $n - 1$ degrees of freedom, the probability under the null hypothesis of obtaining an absolute value of t greater than the t -value observed in this sample.
VAR	s^2 , the variance.

WIDTH=*column-width*

defines the width of the column in which PROC REPORT displays *item*. The value for *column-width* can range from 1 to the value of the LINESIZE= system option. By default, the REPORT procedure assigns a column width just large enough to handle the format.

Note: When you stack items in the same column in a report, the width of the item that is at the bottom of the stack determines the width of the column.

For an example that uses the WIDTH= option, see “Wrapping the Value of a Character Variable across Multiple Rows” later in this chapter.

Interactions of Position and Usage

The position and usage of each variable in the report determine the report's structure and content. PROC REPORT orders the detail rows of the report according to the values of order and group variables, considered from left to right in the report.

For example, in Output 9.10 and Output 9.11, DEPT and DATE are both group variables and BUDGET is an analysis variable used to calculate the SUM statistic. The only difference in the reports is the relative positions of DEPT and DATE. The REPORT procedure groups first on the leftmost group variable.

Output 9.10
Grouping First by DEPT, Then by DATE

DEPT	DATE	BUDGET
Equipment	31MAR92	\$109,500.00
	30JUN92	\$97,500.00
Facilities	31MAR92	\$31,750.00
	30JUN92	\$30,250.00
Other	31MAR92	\$46,500.00
	30JUN92	\$54,500.00
Staff	31MAR92	\$170,000.00
	30JUN92	\$225,000.00
Travel	31MAR92	\$4,300.00
	30JUN92	\$5,700.00

Output 9.11
Grouping First by DATE, Then by DEPT

DATE	DEPT	BUDGET
31MAR92	Equipment	\$109,500.00
	Facilities	\$31,750.00
	Other	\$46,500.00
	Staff	\$170,000.00
	Travel	\$4,300.00
30JUN92	Equipment	\$97,500.00
	Facilities	\$30,250.00
	Other	\$54,500.00
	Staff	\$225,000.00
	Travel	\$5,700.00

Similarly, PROC REPORT orders columns of the report according to the values of across variables, considered from top to bottom. The difference in the appearance of the reports in Output 9.12 and Output 9.13 is due only to the difference in position of DATE and DEPT. In both cases DATE and DEPT are across variables and BUDGET is an analysis variable used to calculate the SUM statistic. In Output 9.12 DATE appears above DEPT, whereas in Output 9.13 DEPT appears above DATE.

Output 9.12
Grouping (across) First by DATE, Then by DEPT

DATE					
31MAR92			30JUN92		
Department			Department		
Other	Staff	Travel	Other	Staff	Travel
BUDGET	BUDGET	BUDGET	BUDGET	BUDGET	BUDGET
\$46,500	\$170,000	\$4,300	\$54,500	\$225,000	\$5,700

Output 9.13
Grouping (across) First by DEPT, Then by DATE

DEPT					
Other		Staff		Travel	
Date		Date		Date	
31MAR92	30JUN92	31MAR92	30JUN92	31MAR92	30JUN92
BUDGET	BUDGET	BUDGET	BUDGET	BUDGET	BUDGET
\$46,500	\$54,500	\$170,000	\$225,000	\$4,300	\$5,700

As you can see in Output 9.12 and Output 9.13, several items can collectively define the contents of a column in a report. For instance, in Output 9.13, the value for BUDGET that appears in each column is collectively determined by the variables BUDGET, DATE, and

DEPT. Each value represents the sum of values of BUDGET for a particular department and a particular quarter. When you use multiple items to define the contents of a column, you can have at most one of the following in a column:

- a display variable with or without a statistic above or below it
- an analysis variable with or without a statistic above or below it
- an order variable
- a group variable
- a computed variable
- the N statistic.

Having more than one of these items in a column creates a conflict for the REPORT procedure about which values to display.

Table 9.2 shows with which other items each type of report item can share a column.

Note: Group and order variables cannot share a column with any other item.

Table 9.2 Report Items that Can Share Columns

	Display	Analysis	Order	Group	Computed	Across	Statistic
Display						X*	X
Analysis						X	X
Order							
Group							
Computed variable						X	
Across variable	X*	X			X	X	X
Statistic	X	X				X	

*When a display variable and an across variable share a column, the report must also contain another variable that is not in the same column.

The following items can stand alone in a column:

- display variable
- analysis variable
- order variable
- group variable
- computed variable
- across variable
- N statistic.

Note: The values in a column occupied only by an across variable are frequency

counts.

Information for the Windowing Environment

The DEFINE statement supports these options in a windowing environment:

- COLOR=
- ITEMHELP=

For more information, see *SAS Guide to the REPORT Procedure*.

ENDCOMP Statement

The ENDCOMP statement ends a compute block. A COMPUTE statement must precede the ENDCOMP statement. For information on compute blocks, see “COMPUTE Statement” earlier in this chapter.

Syntax

The general form of the ENDCOMP statement is

ENDCOMP;

FREQ Statement

The FREQ statement specifies a numeric variable in the input SAS data set whose value represents the frequency of each observation.

You can use the FREQ statement in combination with a WEIGHT statement.

Note: The information from the FREQ statement is not saved by the OUTREPT= option in the PROC REPORT statement.

Syntax

The general form of the FREQ statement is

FREQ *frequency-variable*;

Requirements

You must use the following argument in the FREQ statement:

frequency-variable

a numeric variable in the input data set whose value represents the frequency of each observation. If you use the FREQ statement, PROC REPORT assumes that each observation in the input data set represents n observations, where n is the value of the FREQ variable. If n is not an integer, the REPORT procedure truncates it. If n is less than 1 (which includes missing), PROC REPORT skips the observation.

LINE Statement

The LINE statement writes customized break lines. The break lines can contain

- text
- values calculated for either a set of rows or for the whole report.

The LINE statement is valid only in compute blocks for which the COMPUTE statement contains the BEFORE or AFTER argument. The LINE statement provides a subset of the features of the PUT statement, which you use in the DATA step to write lines to the SAS log, the SAS output file, or an external file.

Note: You cannot use the LINE statement in conditional statements (IF-THEN, IF-THEN/ELSE, and SELECT) because it does not take effect until PROC REPORT has executed all other statements in the compute block.

Syntax

The general form of the LINE statement is

LINE *specification(s)*;

- *specification(s)* can be one or more of the following:
 - *item item-format*
 - *'character-string'*
 - *number-of-repetitions*'character-string'*
 - *pointer-control*

Requirements

You must use one or more of the following kinds of specifications:

item item-format

specifies the item to display and the format to use to display it, where

item is the name of a data set variable, a computed variable, or a statistic in

the report. When you specify a statistic calculated on the values of an analysis variable, use the compound name that identifies both the statistic and the variable with which it is associated. For example, to write the SUM statistic for BUDGET, use the name BUDGET.SUM.

item-format is a SAS or user-defined format. You must specify a format for each item.

'character-string'
specifies a string of text to display. When the string is a blank and nothing else is in *specification(s)*, PROC REPORT prints a blank line.

number-of-repetitions'character-string'*
specifies a character string and the number of times to repeat it. For example, the following LINE statement displays the text 'End of Report' four times:

```
line 4*'End of Report' ;
```

pointer-control
specifies the column in which the REPORT procedure displays the next specification. You can use either of the following forms for pointer controls:

@column-number specifies the number of the column in which to begin displaying the next item in the specification list.

+column-increment specifies the number of columns to skip before beginning to display the next item in the specification list.

For example, the following LINE statement displays BUDGET.SUM with a DOLLAR11.2 format starting in column 25, then skips 10 columns and displays ACTUAL.SUM with the same format:

```
line @25 budget.sum dollar11.2 +10 actual.sum dollar11.2;
```

Differences between the LINE and PUT Statements

The LINE statement does not support the following features of the PUT statement:

- automatic labeling signaled by an equals sign (=), also known as named output
- the `_ALL_`, `INFILE`, and `_PAGE_` arguments and the `OVERPRINT` option
- grouping items and formats to apply one format to a list of items
- pointer control using variables and expressions
- line pointer controls (`#` and `/`)
- trailing “at” signs (`@` and `@@`)
- format modifiers
- array elements.

RBREAK Statement

The RBREAK statement produces a default summary at the beginning or end of a report. The *break lines* that make up a default report summary can

- visually separate observations
- summarize statistics and computed variables for all rows of the report. A break line that summarizes information is a *summary line*.

For an example of the RBREAK statement and its options, see “Creating a Default Report Summary with the RBREAK Statement” later in this chapter.

Syntax

The general form of the RBREAK statement is

RBREAK *location* < / *break-option(s)* >;

- *location* is one of the following:

AFTER

BEFORE

- *break-option(s)* can be one or more of the following:

DOL

DUL

OL

PAGE

SKIP

SUMMARIZE

UL

Requirements

You must specify a location in the RBREAK statement.

location

controls the placement of the break lines and is either

AFTER places the break lines at the end of the report.

BEFORE places the break lines at the beginning of the report.

Options

The RBREAK statement supports the following options:

DOL

(for double overlining) uses the thirteenth character in the string defined by the SAS system option FORMCHAR= to overline each value

- that appears in the summary line
- that would appear in the summary line if you specified the SUMMARIZE option.

By default, the thirteenth character in the FORMCHAR= string is an equals sign (=). Use the FORMCHAR= system option to change it to another character.

If you specify both the OL and DOL options, PROC REPORT honors only OL.

DUL

(for double underlining) uses the thirteenth character in the string defined by the SAS system option FORMCHAR= to underline each value

- that appears in the summary line
- that would appear in the summary line if you specified the SUMMARIZE option.

By default, the thirteenth character in the FORMCHAR= string is an equals sign (=). Use the FORMCHAR= system option to change it to another character.

If you specify both the UL and DUL options, PROC REPORT honors only UL.

OL

(for overlining) uses the second character in the string defined by the SAS system option FORMCHAR= to overline each value

- that appears in the summary line
- that would appear in the summary line if you specified the SUMMARIZE option.

By default, the second character in the FORMCHAR= string is a hyphen (-). Use the FORMCHAR= system option to change it to another character.

If you specify both the OL and DOL options, PROC REPORT honors only OL.

PAGE

starts a new page after the last break line of a break located at the beginning of the report.

SKIP

writes a blank line after the last break line of a break located at the beginning of the report.

SUMMARIZE

includes a summary line as one of the break lines. A summary line at the beginning or end of a report contains values for

- statistics

- analysis variables
- computed variables.

The following table shows how PROC REPORT calculates the value for each kind of report item in a summary line created by the RBREAK statement:

If the report item is ...	then its value is ...
a statistic	the value of the statistic over all observations in the set
an analysis variable	the value of the statistic specified as the usage option in the DEFINE statement. PROC REPORT calculates the value of the statistic over all observations in the set. The default usage is SUM.
a computed variable	the results of the calculations based on the code in the corresponding compute block (see “COMPUTE Statement” later in this chapter.)

UL

(for underlining) uses the second character in the string defined by the SAS system option FORMCHAR= to underline each value

- that appears in the summary line
- that would appear in the summary line if you specified the SUMMARIZE option.

By default, the second character in the FORMCHAR= string is a hyphen (-). Use the FORMCHAR= system option to change it to another character.

If you specify both the UL and DUL options, PROC REPORT honors only UL.

Order of Break Lines

When a default summary contains more than one break line, the order in which the break lines appear is

1. overlining or double overlining (OL or DOL)
2. summary line (SUMMARIZE)
3. underlining or double underlining (UL or DUL)
4. skipped line (SKIP)
5. page break (PAGE).

Note: If you define a customized summary for the break, customized break lines appear after underlining or double underlining. For more information on customized break lines, see “COMPUTE Statement” and “LINE Statement” later in this chapter.

Information for the Windowing Environment

The RBREAK statement supports the COLOR= option in the windowing environment. For more information, see *SAS Guide to the REPORT Procedure*.

WEIGHT Statement

The WEIGHT statement specifies a numeric variable in the input data set whose value weights each analysis variable.

Note: The information from the WEIGHT statement is not saved by the OUTREPT= option in the PROC REPORT statement.

Syntax

The general form of the WEIGHT statement is

```
WEIGHT weight-variable ;
```

Requirements

You must use the following argument in the WEIGHT statement:

weight-variable

a numeric variable in the input data set whose value weights each analysis variable. The WEIGHT variable need not be an integer and does not affect the degrees of freedom. If the value of *weight-variable* is less than 0 (which includes missing), PROC REPORT uses a value of 0.

If you use a WEIGHT statement, PROC REPORT uses the value of the WEIGHT variable to calculate weighted statistics. For information on how the WEIGHT value affects the calculation of statistics, refer to w_j in the discussion of the *statistic* option in “DEFINE Statement” earlier in this chapter.

Examples

The examples in this section combine multiple features of the REPORT procedure. For examples that focus on individual tasks, refer to the usage part of the book.

Specifying and Ordering Columns to Display with the COLUMN Statement

This example produces a report that contains only three of the five variables in the input data set. The following statements produce Output 9.14:

```
proc report data=budget2 nowindows;
  column account date budget;
  title 'Budgets for Each Account in Each Quarter';
run;
```

Output 9.14 Selecting and Ordering Items to Display

Budgets for Each Account in Each Quarter			1
ACCOUNT	DATE	BUDGET	
fulltime	31MAR92	\$130,000.00	
fulltime	30JUN92	\$165,000.00	
parttime	31MAR92	\$40,000.00	
parttime	30JUN92	\$60,000.00	
lease	31MAR92	\$40,000.00	
lease	30JUN92	\$40,000.00	
purchase	31MAR92	\$40,000.00	
purchase	30JUN92	\$20,000.00	
tape	31MAR92	\$8,000.00	
tape	30JUN92	\$12,000.00	
sets	31MAR92	\$7,500.00	
sets	30JUN92	\$7,500.00	
maint	31MAR92	\$10,000.00	
maint	30JUN92	\$12,000.00	
rental	31MAR92	\$4,000.00	
rental	30JUN92	\$6,000.00	
rent	31MAR92	\$24,000.00	
rent	30JUN92	\$24,000.00	
utils	31MAR92	\$5,000.00	
utils	30JUN92	\$3,500.00	
supplies	31MAR92	\$2,750.00	
supplies	30JUN92	\$2,750.00	
leases	31MAR92	\$3,500.00	
leases	30JUN92	\$4,500.00	
gas	31MAR92	\$800.00	
gas	30JUN92	\$1,200.00	
advert	31MAR92	\$30,000.00	
advert	30JUN92	\$30,000.00	
talent	31MAR92	\$13,500.00	
talent	30JUN92	\$19,500.00	
musicfee	31MAR92	\$3,000.00	
musicfee	30JUN92	\$5,000.00	

Program Notes

- The COLUMN statement specifies the items to appear in the report.
- The order of the items in the COLUMN statement determines the order in which the items appear in the report.

Ordering Rows with the ORDER and ORDER= Options

This example orders the rows according to the positions of the variables in the COLUMN statement and the orders specified by the ORDER= option for each variable. The following statements produce Output 9.15:

```
proc report data=budget2 nowindows;
  column dept account date budget actual;
  define dept    / order;
  define account / order;
  define date    / order order=internal descending;
  define budget  / analysis sum;
  define actual  / analysis sum;

  title 'Budgets and Expenditures for All Accounts';
run;
```

Output 9.15
Ordering Rows by
DEPT, ACCOUNT,
and DATE

Budgets and Expenditures for All Accounts					1
DEPT	ACCOUNT	DATE	BUDGET	ACTUAL	
Equipment	lease	30JUN92	\$40,000.00	\$40,000.00	
		31MAR92	\$40,000.00	\$40,000.00	
	maint	30JUN92	\$12,000.00	\$10,675.29	
		31MAR92	\$10,000.00	\$7,542.13	
	purchase	30JUN92	\$20,000.00	\$17,769.15	
		31MAR92	\$40,000.00	\$48,282.38	
	rental	30JUN92	\$6,000.00	\$5,482.94	
		31MAR92	\$4,000.00	\$3,998.87	
	sets	30JUN92	\$7,500.00	\$8,079.62	
		31MAR92	\$7,500.00	\$8,342.68	
	tape	30JUN92	\$12,000.00	\$11,426.73	
		31MAR92	\$8,000.00	\$6,829.42	
Facilities	rent	30JUN92	\$24,000.00	\$24,000.00	
		31MAR92	\$24,000.00	\$24,000.00	
	supplies	30JUN92	\$2,750.00	\$2,742.48	
		31MAR92	\$2,750.00	\$2,216.55	
	utils	30JUN92	\$3,500.00	\$3,444.81	
		31MAR92	\$5,000.00	\$4,223.29	
Other	advert	30JUN92	\$30,000.00	\$37,325.64	
		31MAR92	\$30,000.00	\$32,476.98	
	musicfee	30JUN92	\$5,000.00	\$4,875.95	
		31MAR92	\$3,000.00	\$2,550.50	
	talent	30JUN92	\$19,500.00	\$18,424.64	
		31MAR92	\$13,500.00	\$12,986.73	
Staff	fulltime	30JUN92	\$165,000.00	\$166,345.75	
		31MAR92	\$130,000.00	\$127,642.68	
	parttime	30JUN92	\$60,000.00	\$56,018.96	
		31MAR92	\$40,000.00	\$43,850.12	
Travel	gas	30JUN92	\$1,200.00	\$984.93	
		31MAR92	\$800.00	\$537.26	
	leases	30JUN92	\$4,500.00	\$3,889.65	
		31MAR92	\$3,500.00	\$3,045.15	

Program Notes

- The ORDER option in the DEFINE statements for DEPT, ACCOUNT, and DATE defines each of them as an order variable. PROC REPORT orders the rows first according to the values of DEPT, then ACCOUNT, then DATE.
- The order for each item is determined by the ORDER= and the DESCENDING options in the corresponding DEFINE statement. If ORDER= is not specified, PROC REPORT uses ORDER=FORMATTED. The DESCENDING option reverses the default sort order.
 - The DEFINE statement for DEPT orders values of DEPT by their ascending formatted order.
 - The DEFINE statement for ACCOUNT orders values of ACCOUNT (within each department) by their ascending formatted values.
 - The ORDER= and DESCENDING options in the DEFINE statement for DATE order values of DATE (within each account) by their descending internal (reverse chronological) order.

Grouping Observations with the GROUP Option and an Analysis Variable

This example creates groups by consolidating into one row all observations for each department. The following statements produce Output 9.16:

```
proc report data=budget2 nowindows;
  column dept budget;

  define dept / group;
  define budget / analysis sum;

  title 'Departmental Budgets';
run;
```

Output 9.16
*Creating a Single
 Row for Each
 Department*

Departmental Budgets		1
DEPT	BUDGET	
Equipment	\$207,000.00	
Facilities	\$62,000.00	
Other	\$101,000.00	
Staff	\$395,000.00	
Travel	\$10,000.00	

Program Notes

- The GROUP option in the DEFINE statement for DEPT consolidates into one row of the report all observations with the same value of DEPT.
- The SUM option in the DEFINE statement for BUDGET specifies that the value for BUDGET in each row of the report is the sum of the values of BUDGET for the individual observations that make up that row. For example, in the first row of the report, the value of BUDGET is \$207,000. This number is the sum of BUDGET for all observations in the data set for which DEPT='Equipment'. These observations have different values for ACCOUNT and DATE, but they all have the same value of DEPT.

Creating Column Headers from Variable Values with the ACROSS Option

This example creates a column header for each value of DATE. Under each column header are two columns: one for BUDGET and one for ACTUAL.

The following statements produce Output 9.17:

```
proc report data=budget2 nowindows;
  column dept date, (budget actual);

  define dept / group;
  define date / across order=internal;
  define budget / analysis sum;
  define actual / analysis sum;

  title 'Departmental Budgets and Expenditures for Each Quarter';
run;
```

Output 9.17
Creating a Column Header for Each Value of DATE

Departmental Budgets and Expenditures for Each Quarter					1
DEPT	DATE				
	31MAR92		30JUN92		
	BUDGET	ACTUAL	BUDGET	ACTUAL	
Equipment	\$109,500.00	\$114,995.48	\$97,500.00	\$93,433.73	
Facilities	\$31,750.00	\$30,439.84	\$30,250.00	\$30,187.29	
Other	\$46,500.00	\$48,014.21	\$54,500.00	\$60,626.23	
Staff	\$170,000.00	\$171,492.80	\$225,000.00	\$222,364.71	
Travel	\$4,300.00	\$3,582.41	\$5,700.00	\$4,874.58	

Program Notes

- The COLUMN statement includes DEPT, DATE, BUDGET, and ACTUAL in the report.
- The comma after DATE indicates that the next item or list of items is under DATE.
- The parentheses around BUDGET and ACTUAL indicate that they both get treated the same way (in this case, placed under DATE).

- The ACROSS option in the DEFINE statement for DATE creates a column header for each value of DATE.

Note: The COLUMN statement and the DEFINE statement for DATE work together to create this report layout.

Printing All Values for a Long Row on Consecutive Lines with the WRAP Option

This example contrasts two PROC REPORT steps. They are identical except that one uses the WRAP and NAMED options in the PROC REPORT statement and the other does not. The report in Output 9.18 cannot fit values for all variables into one row, so it uses two pages to print the report. This layout is the default.

Output 9.18
Default Placement
of Long Rows

Amounts Budgeted and Spent for Equipment for the First and Second Quarters					1
		DATE			
		31MAR92		30JUN92	
DEPT	ACCOUNT	BUDGET	ACTUAL	BUDGET	
Equipment	lease	\$40,000.00	\$40,000.00	\$40,000.00	
	maint	\$10,000.00	\$7,542.13	\$12,000.00	
	purchase	\$40,000.00	\$48,282.38	\$20,000.00	
	rental	\$4,000.00	\$3,998.87	\$6,000.00	
	sets	\$7,500.00	\$8,342.68	\$7,500.00	
	tape	\$8,000.00	\$6,829.42	\$12,000.00	

Amounts Budgeted and Spent for Equipment for the First and Second Quarters		2
		DATE
		30JUN92
		ACTUAL
		\$40,000.00
		\$10,675.29
		\$17,769.15
		\$5,482.94
		\$8,079.62
		\$11,426.73

The following statements are identical to the ones that produced Output 9.18 except that they include the NAMED and WRAP options in the PROC REPORT statement. These statements produce Output 9.19:

```
proc report data=budget2 nowindows ls=64 named wrap;
  where dept='Equipment';
  column dept account date,(budget actual);
```

```

define dept    / group;
define account / group;
define date    / across;
define budget  / analysis sum;
define actual  / analysis sum;

break after account / skip;

title 'Amounts Budgeted and Spent for Equipment';
title2 'for the First and Second Quarters';
run;

```

Output 9.19
*Wrapping Values
 onto Consecutive
 Rows*

Amounts Budgeted and Spent for Equipment			1
for the First and Second Quarters			
DEPT=Equipment	ACCOUNT=lease	BUDGET= \$40,000.00	
ACTUAL= \$40,000.00	BUDGET= \$40,000.00	ACTUAL= \$40,000.00	
DEPT=	ACCOUNT=maint	BUDGET= \$12,000.00	
ACTUAL= \$10,675.29	BUDGET= \$10,000.00	ACTUAL= \$7,542.13	
DEPT=	ACCOUNT=purchase	BUDGET= \$20,000.00	
ACTUAL= \$17,769.15	BUDGET= \$40,000.00	ACTUAL= \$48,282.38	
DEPT=	ACCOUNT=rental	BUDGET= \$6,000.00	
ACTUAL= \$5,482.94	BUDGET= \$4,000.00	ACTUAL= \$3,998.87	
DEPT=	ACCOUNT=sets	BUDGET= \$7,500.00	
ACTUAL= \$8,079.62	BUDGET= \$7,500.00	ACTUAL= \$8,342.68	
DEPT=	ACCOUNT=tape	BUDGET= \$12,000.00	
ACTUAL= \$11,426.73	BUDGET= \$8,000.00	ACTUAL= \$6,829.42	

Program Notes

- The NAMED option removes column headers from the top of the page and writes each one in front of the corresponding value.
- The WRAP option wraps the columns for a single row of the report onto consecutive lines instead of using multiple pages.

Note: Values for DEPT are missing after the first row of the report because PROC REPORT does not repeat the values of a group variable from one row to the next if the value does not change.

Creating a Group for Missing Values with the MISSING Option

The reports in this example use a data set, TESTMISS, which is identical to BUDGET2 except that observations in BUDGET2 that had DEPT='Equipment' have a missing value for DEPT in TESTMISS.

This example consolidates into one row all observations with the same value of the group variable DEPT. By default, PROC REPORT does not include observations with a missing value for a group variable. The report in Output 9.20 contains rows only for groups that do not have missing values. The sum for BUDGET (\$568,00.00) does not include the

budget for the equipment department.

Output 9.20
Ignoring Group Variables with Missing Values

Sum of Departmental Budgets		1
DEPT	BUDGET	
Facilities	\$62,000.00	
Other	\$101,000.00	
Staff	\$395,000.00	
Travel	\$10,000.00	

	\$568,000.00	

The following statements are identical to the ones that produced Output 9.20 except that they include the MISSING option in the PROC REPORT statement. These statements produce Output 9.21:

```
proc report data=testmiss nowindows missing;
  column dept budget;

  define dept / group;
  define budget / analysis sum;

  rbreak after / ol summarize;

  title 'Sum of Departmental Budgets';
run;
```

Output 9.21
Including Observations with Missing Values of DEPT

Sum of Departmental Budgets		1
DEPT	BUDGET	
	\$207,000.00	
Facilities	\$62,000.00	
Other	\$101,000.00	
Staff	\$395,000.00	
Travel	\$10,000.00	

	\$775,000.00	

Program Notes

- The MISSING option includes a row for observations with a missing value of the group variable, DEPT. The sum for BUDGET (\$775,000.00) includes the budget for the equipment department.

Customizing Column Headers

This example customizes column headers for every column and creates a header that spans two columns. The following statements produce Output 9.22:

```
proc report data=budget2 nowindows;
  column dept account ('_Canadian Dollars_' budget actual);

  define dept    / group 'Department' '==';
  define account / group 'Account' '==';
  define budget  / analysis sum 'Amount/in Budget' '==';
  define actual  / analysis sum 'Amount Spent' '==';

  title 'Budgets and Expenditures for Each Account';
run;
```

Output 9.22
Customizing
Column Headers

		Canadian Dollars		1
		Amount	Amount	
Department	Account	in Budget	Spent	
=====	=====	=====	=====	
Equipment	lease	\$80,000.00	\$80,000.00	
	maint	\$22,000.00	\$18,217.42	
	purchase	\$60,000.00	\$66,051.53	
	rental	\$10,000.00	\$9,481.81	
	sets	\$15,000.00	\$16,422.30	
	tape	\$20,000.00	\$18,256.15	
Facilities	rent	\$48,000.00	\$48,000.00	
	supplies	\$5,500.00	\$4,959.03	
	utils	\$8,500.00	\$7,668.10	
Other	advert	\$60,000.00	\$69,802.62	
	musicfee	\$8,000.00	\$7,426.45	
	talent	\$33,000.00	\$31,411.37	
Staff	fulltime	\$295,000.00	\$293,988.43	
	parttime	\$100,000.00	\$99,869.08	
Travel	gas	\$2,000.00	\$1,522.19	
	leases	\$8,000.00	\$6,934.80	

Program Notes

- The COLUMN statement creates the header “Canadian Dollars”, which spans the columns for the variables BUDGET and ACTUAL. The underscores that precede and follow the header in the program extend the header in the report to fill both columns.
- The DEFINE statements for DEPT, ACCOUNT, BUDGET, and ACTUAL contain quoted strings that specify the headers for the columns. In each case the second quoted string (==) expands to underline the text of the column header.
- The DEFINE statement for BUDGET uses the split character to control the line break in the header.

Making Multipanel Reports with the PANELS= Option

This example uses a multi-panel format. The following statements produce Output 9.23:

```
proc report data=budget2 nowindows panels=99 ps=19 ls=100;
  column date dept account budget;

  title 'Departmental Budgets';
run;
```

Output 9.23
Creating a Report
with Panels

Departmental Budgets								1
DATE	DEPT	ACCOUNT	BUDGET	DATE	DEPT	ACCOUNT	BUDGET	
31MAR92	Staff	fulltime	\$130,000.00	31MAR92	Facilities	rent	\$24,000.00	
30JUN92	Staff	fulltime	\$165,000.00	30JUN92	Facilities	rent	\$24,000.00	
31MAR92	Staff	parttime	\$40,000.00	31MAR92	Facilities	utils	\$5,000.00	
30JUN92	Staff	parttime	\$60,000.00	30JUN92	Facilities	utils	\$3,500.00	
31MAR92	Equipment	lease	\$40,000.00	31MAR92	Facilities	supplies	\$2,750.00	
30JUN92	Equipment	lease	\$40,000.00	30JUN92	Facilities	supplies	\$2,750.00	
31MAR92	Equipment	purchase	\$40,000.00	31MAR92	Travel	leases	\$3,500.00	
30JUN92	Equipment	purchase	\$20,000.00	30JUN92	Travel	leases	\$4,500.00	
31MAR92	Equipment	tape	\$8,000.00	31MAR92	Travel	gas	\$800.00	
30JUN92	Equipment	tape	\$12,000.00	30JUN92	Travel	gas	\$1,200.00	
31MAR92	Equipment	sets	\$7,500.00	31MAR92	Other	advert	\$30,000.00	
30JUN92	Equipment	sets	\$7,500.00	30JUN92	Other	advert	\$30,000.00	
31MAR92	Equipment	maint	\$10,000.00	31MAR92	Other	talent	\$13,500.00	
30JUN92	Equipment	maint	\$12,000.00	30JUN92	Other	talent	\$19,500.00	
31MAR92	Equipment	rental	\$4,000.00	31MAR92	Other	musicfee	\$3,000.00	
30JUN92	Equipment	rental	\$6,000.00	30JUN92	Other	musicfee	\$5,000.00	

Program Notes

- The PANELS= option creates a multi-panel report. The value for the option (99) is larger than the number of panels that PROC REPORT can fit on a page. In such a case, the procedure creates as many panels as it can.

Adding Variables to a Report with a COMPUTE Block

This example adds to the report a variable, BALANCE, that is not in the input data set. The following statements produce Output 9.24:

```
proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept      / group;
  define account  / group;
  define budget   / analysis sum;
  define actual   / analysis sum;
  define balance  / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
  endcomp;

  title 'Account Balances';
run;
```

Output 9.24
Adding the
Computed Variable
BALANCE to the
Report

Account Balances					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
Facilities	tape	\$20,000.00	\$18,256.15	\$1,743.85	
	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
Other	utils	\$8,500.00	\$7,668.10	\$831.90	
	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
Staff	talent	\$33,000.00	\$31,411.37	\$1,588.63	
	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
Travel	parttime	\$100,000.00	\$99,869.08	\$130.92	
	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	

Program Notes

- The COLUMN statement includes BALANCE in the report.
- The DEFINE statement for BALANCE defines it as a computed variable and assigns it a format of DOLLAR11.2.
- The compute block calculates the value of BALANCE. The assignment statement in the compute block executes on every row of the report.

Note: The assignment statement uses the compound names BUDGET.SUM and ACTUAL.SUM because BUDGET and ACTUAL are analysis variables used to calculate the SUM statistic. (See “How to Reference Data Set Variables in a Compute Block” in “COMPUTE Statement” earlier in this chapter.)

Creating Default Group Summaries with the BREAK Statement

This example creates two default group summaries: one each time the value of DEPT changes, the other each time the value of ACCOUNT changes. The break between values of ACCOUNT visually separates the groups with a blank line. The break between values of DEPT visually separates the groups and contains summary information for analysis and computed variables.

The following statements produce Output 9.25:

```
proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
  endcomp;

  break after account / skip;
  break after dept / ol ul summarize suppress skip;

  title 'Departmental Budgets, Expenditures, and Balances';
run;
```

Output 9.25
*Summing BUDGET,
 ACTUAL, and
 BALANCE for Each
 Department and
 Double-Spacing
 between Accounts*

Departmental Budgets, Expenditures, and Balances					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	
			----- \$207,000.00 -----	----- \$208,429.21 -----	----- \$-1,429.21 -----
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
			----- \$62,000.00 -----	----- \$60,627.13 -----	----- \$1,372.87 -----
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
		----- \$101,000.00 -----	----- \$108,640.44 -----	----- \$-7,640.44 -----	
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
			----- \$395,000.00 -----	----- \$393,857.51 -----	----- \$1,142.49 -----
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	
			----- \$10,000.00 -----	----- \$8,456.99 -----	----- \$1,543.01 -----

Program Notes

- The SKIP option in the first BREAK statement (BREAK AFTER ACCOUNT) puts a blank line between rows each time the value of ACCOUNT changes.
- A variety of options in the second BREAK statement (BREAK AFTER DEPT) control the appearance of these break lines:
 - The OL option overlines each value in the summary line.
 - The UL option underlines each value in the summary line.
 - The SUMMARIZE option summarizes statistics and computed variables for the group.
 - The SUPPRESS option suppresses the value of the break variable (DEPT) from the summary line.
 - The SKIP option puts a blank line at the end of the summary.

Creating a Default Report Summary with the RBREAK Statement

This example creates a default report summary at the end of the report. The report break visually separates the detail rows from the summary line. The summary line contains information for analysis and computed variables. The following statements produce Output 9.26:

```
proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
  endcomp;

  rbreak after / dol dul summarize;

  title 'Company Budget, Expenditures, and Balance';
run;
```

Output 9.26
Summing BUDGET,
ACTUAL, and
BALANCE at End of
the Report

Company Budget, Expenditures, and Balance					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	
Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	
Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	
Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	
Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	
		=====	=====	=====	
		\$775,000.00	\$780,011.28	\$-5,011.28	
		=====	=====	=====	

Program Notes

- The DOL option overlines each value in the summary line.
- The DUL option underlines each value in the summary line.
- The SUMMARIZE option summarizes statistics and computed variables for the group.

Creating Customized Group Summaries with the COMPUTE Statement

This example creates a customized group summary that appears each time the value of DEPT changes. The summaries include much the same information as the default summary in “Creating Default Group Summaries with the BREAK Statement”, but the customized summaries contain text and control the placement of the text. The following statements produce Output 9.27:

```
proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
  endcomp;
  compute after dept;
    line ' ';
    line @15 51*'-' ;
    line @15 '| The budget for this department was ' budget.sum dollar11.2'. '|';
    line @15 '| The department spent ' actual.sum dollar11.2 '.' @65 '|';
    line @15 51*'-' ;
    line ' ';
  endcomp;

  title 'Departmental Budgets, Expenditures, and Balances';
run;
```

Output 9.27
*Customizing
 Summary Text for
 Each Department*

Departmental Budgets, Expenditures, and Balances					1
DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE	
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00	
	maint	\$22,000.00	\$18,217.42	\$3,782.58	
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53	
	rental	\$10,000.00	\$9,481.81	\$518.19	
	sets	\$15,000.00	\$16,422.30	\$-1,422.30	
	tape	\$20,000.00	\$18,256.15	\$1,743.85	

The budget for this department was \$207,000.00.					
The department spent \$208,429.21.					

Facilities	rent	\$48,000.00	\$48,000.00	\$0.00	
	supplies	\$5,500.00	\$4,959.03	\$540.97	
	utils	\$8,500.00	\$7,668.10	\$831.90	

The budget for this department was \$62,000.00.					
The department spent \$60,627.13.					

Other	advert	\$60,000.00	\$69,802.62	\$-9,802.62	
	musicfee	\$8,000.00	\$7,426.45	\$573.55	
	talent	\$33,000.00	\$31,411.37	\$1,588.63	

The budget for this department was \$101,000.00.					
The department spent \$108,640.44.					

Staff	fulltime	\$295,000.00	\$293,988.43	\$1,011.57	
	parttime	\$100,000.00	\$99,869.08	\$130.92	

The budget for this department was \$395,000.00.					
The department spent \$393,857.51.					

Travel	gas	\$2,000.00	\$1,522.19	\$477.81	
	leases	\$8,000.00	\$6,934.80	\$1,065.20	

The budget for this department was \$10,000.00.					
The department spent \$8,456.99.					

Program Notes

This example uses two compute blocks. The first computes values for the variable BALANCE. The second (COMPUTE AFTER DEPT) creates a customized group summary each time the value of DEPT changes. In this compute block

- The LINE statements with blanks for arguments write blank lines as the first and last lines of the summary.
- The LINE statements with `51*~` write fifty-one hyphens as the second and second to last summary lines.
- The LINE statements with “at” signs (@) use pointer control to place the text in the specified column.
- The variables BUDGET.SUM and ACTUAL.SUM appear in the summary lines with the format assigned to them in the LINE statement. The format follows the variable name.
- Pointer control places the line-drawing characters so that they draw a box around the text of the summaries.

Wrapping the Value of a Character Variable across Multiple Rows

This example wraps the value of COMMENT over as many lines as necessary. The value of COMMENT is either a blank or “This department has overdrawn its budget.” The following statements produce Output 9.28:

```
proc report data=budget2 nowindows;
  column dept budget actual comment;

  define dept    / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define comment / computed width=20 flow;

  compute comment / char length=41;
    if budget.sum-actual.sum < 0 then
      comment='This department has overdrawn its budget.';
    else comment=' ';
  endcomp;

  title 'Report of Departments that Overspent Their Budgets';
run;
```

Output 9.28
Wrapping the Value
of COMMENT
across Multiple
Rows

Report of Departments that Overspent Their Budgets				1
DEPT	BUDGET	ACTUAL	COMMENT	
Equipment	\$207,000.00	\$208,429.21	This department has overdrawn its budget.	
Facilities	\$62,000.00	\$60,627.13		
Other	\$101,000.00	\$108,640.44	This department has overdrawn its budget.	
Staff	\$395,000.00	\$393,857.51		
Travel	\$10,000.00	\$8,456.99		

Program Notes

- The WIDTH= option in the DEFINE statement for COMMENT sets the width of that column to 20 characters.
- The FLOW option in the DEFINE statement for COMMENT wraps the value of COMMENT over as many lines as necessary. Without the FLOW option, PROC REPORT would truncate the value of COMMENT after 20 characters.

Suppressing the Display of an Item with the NOPRINT Option

This example uses the variables BUDGET and ACTUAL to calculate values of BALANCE but does not display values for BUDGET and ACTUAL. The following statements produce Output 9.29:

```
proc report data=budget2 nowindows;
  column dept account budget actual balance;

  define dept    / group;
  define account / group;
  define budget  / analysis sum noprint;
  define actual  / analysis sum noprint;
  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
  endcomp;

  title 'Account Balances';
run;
```

Output 9.29
Suppressing the
Display of BUDGET
and ACTUAL

Account Balances			1
DEPT	ACCOUNT	BALANCE	
Equipment	lease	\$0.00	
	maint	\$3,782.58	
	purchase	\$-6,051.53	
	rental	\$518.19	
	sets	\$-1,422.30	
	tape	\$1,743.85	
Facilities	rent	\$0.00	
	supplies	\$540.97	
	utils	\$831.90	
Other	advert	\$-9,802.62	
	musicfee	\$573.55	
	talent	\$1,588.63	
Staff	fulltime	\$1,011.57	
	parttime	\$130.92	
Travel	gas	\$477.81	
	leases	\$1,065.20	

Program Notes

- The COLUMN statement includes BUDGET and ACTUAL in the report so that their values are available for computing BALANCE.
- The NOPRINT option in the DEFINE statements for BUDGET and ACTUAL suppresses their display.

Writing a Report for Use in a Windowing Environment

This example produces a simple, straightforward report in the nonwindowing environment. However, although the report was developed in a nonwindowing environment and can be run in that environment, it contains several features that make it particularly useful to someone who runs it in a windowing environment. If you submit this PROC REPORT step with the WINDOWS option

- negative balances appears in yellow.
- the user can select any balance and issue the EXECUTE command. This command loads another report definition and shows a more detailed report for the corresponding department.

Creating the Report Definition that the EXECUTE Command Loads

The following statements create the report definition for the detailed report that the users load with the EXECUTE command. Run this PROC REPORT step to make the report definition available.

```
proc report data=budget2 nowindows outrept=user.reports.details;

    column dept account budget actual balance;

    define dept      / group;
    define account  / group;
    define budget   / analysis sum;
    define actual   / analysis sum;
    define balance  / computed format=dollar11.2;

    compute balance;
        balance=budget.sum-actual.sum;
    endcomp;
run;
```

If the catalog USER.REPORTS does not already exist, PROC REPORT creates it.

Building the Report to Use in a Windowing Environment

The following statements create the report definitions that users will use to create their reports in the windowing environment.

```
proc report data=budget2 outrept=user.reports.interact;

  column dept budget actual balance;

  define dept    / group;
  define budget / analysis sum noprint;
  define actual  / analysis sum noprint;
  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
    if balance<0 then call define(_col_,'color','yellow');
    string='rload user.reports.details;
           where dept="' ||dept||'";
           call define('balance','command',string);
  endcomp;

  break after dept / skip;
run;
```

Program Notes

- The OUTREPT= option stores the report definition in USER.REPORTS.INTERACT.
- If BALANCE is less than 0, the first CALL DEFINE statement sets to yellow the color of the current column.
- The assignment statement for STRING creates a string of SAS commands to pass to the EXECUTE command. These commands

1. Load the previously created report definition, USER.REPORTS.DETAILS
2. Select for this report only observations with the current value of DEPT.
The three sets of quotation marks in the assignment statement are crucial:
 - a. The outer set of single quotation marks defines STRING as a character string.
 - b. The double quotation marks ensure that everything inside them (||dept||) is treated as a string, so that the WHERE clause reads

```
where dept='current-value-of-DEPT'
```

- c. The inner set of quotation marks become part of the WHERE clause (see previous bullet).
- The second CALL DEFINE statement makes STRING the argument to the EXECUTE command for the column for BALANCE. The value of STRING could go directly in

the CALL DEFINE statement, in quotation marks, but it is clearer to see the value of STRING when it is stored as a separate variable.

Using the Report Definition in the Windowing Environment

To use the report definition, users enter the following PROC REPORT step:

```
proc report data=budget2 windows report=user.reports.interact;
  title 'Summary of Departmental Budgets';
run;
```

Display 9.1 appears.

Display 9.1
*Report from which
Users Can Access a
More Detailed
Report*

DEPT	BALANCE
Equipment	\$-1,429.21
Facilities	\$1,372.87
Other	\$-7,640.44
Staff	\$1,142.49
Travel	\$1,543.01

A user in the windowing environment who selects the value of BALANCE for the Equipment department and issues the EXECUTE command sees Display 9.2.

Display 9.2
Detailed Report
Loaded with the
EXECUTE
Command

Command ==>

Summary of Departmental Budgets 1

DEPT	ACCOUNT	BUDGET	ACTUAL	BALANCE
Equipment	lease	\$80,000.00	\$80,000.00	\$0.00
	maint	\$22,000.00	\$18,217.42	\$3,782.58
	purchase	\$60,000.00	\$66,051.53	\$-6,051.53
	rental	\$10,000.00	\$9,481.81	\$518.19
	sets	\$15,000.00	\$16,422.30	\$-1,422.30
	tape	\$20,000.00	\$18,256.15	\$1,743.85

RKEYS

Delete <F4> Move <F5> Define <F6> Pageback <F7>
Page <F8>

Note: To return to the previous report, the user must issue these commands:

```
end
where undo
```

You may want to build this capability into your own customized pmenus for the REPORT procedure. For information on customizing pmenus see Chapter 26 “The PMENU Procedure” in *SAS Procedures Guide*. For information on using customized pmenus with PROC REPORT see “Specifying Alternative Action Bars and Pull-Down Menus” in the discussion of the PROFILE window in Chapter 9 of *SAS Guide to the REPORT Procedure*.

Chapter 10 How PROC REPORT Builds a Report

Introduction 243

Sequence of Events 243

Compound Names 244

Building a Report that Uses Groups and a Report Summary 245

Building a Report that Uses DATA Step Variables 248

Introduction

This chapter first explains the process of building a report. Following this explanation are illustrations of how PROC REPORT creates two sample reports.

To understand the process of building a report, you must understand the difference between report variables and DATA step variables. Variables that appear only in one or more compute blocks are *DATA step variables*. Variables that appear in one or more columns of the report are *report variables*. A report variable may or may not appear in a compute block.

Sequence of Events

PROC REPORT constructs a report as follows:

1. It consolidates the data by group, order, and across variables. It calculates all statistics for the report, those for detail rows as well as those for summary lines in breaks. Statistics include analysis variables. PROC REPORT calculates statistics for summary lines whether or not they appear in the report. It stores all this information in a temporary file.
2. It initializes all DATA step variables to missing.
3. It begins constructing the rows of the report.
 - a. At the beginning of each row, it initializes all report variables to missing.
 - b. It fills in values for report variables from left to right.
 - Values for computed variables come from executing the statements in the corresponding compute blocks.
 - Values for all other variables come from the temporary file created at the beginning of the report-building process.
 - c. Whenever it comes to a break, PROC REPORT first constructs the break lines created with the BREAK or RBREAK statement (for example, overlining and underlining). It then executes the statements in the compute block attached to the break (if there is one).

Construction of Summary Lines

PROC REPORT constructs a summary line for a break if either of the following conditions is true:

- You use the SUMMARIZE option in the corresponding BREAK or RBREAK statement.
- You use a compute block at the break (whether or not there is a corresponding BREAK or RBREAK statement).

The summary line that PROC REPORT constructs at this point is preliminary. If no compute block is attached to the break, the preliminary summary line becomes the final summary line. However, if a compute block is attached to the break, the statements in the compute block can alter the values in the preliminary summary line.

PROC REPORT prints the summary line only if you use the SUMMARIZE option.

Because of the way PROC REPORT builds a report, you can

- Use group statistics in compute blocks for a break before the group variable.
- Use statistics for the whole report in a compute block at the beginning of the report.

You reference these statistics with the appropriate compound name.

Compound Names

When you use statistics in a report, you refer to them in compute blocks by a compound name like BUDGET.SUM. However, in different parts of the report, that same name takes on different meanings. Consider the report in Output 10.1:

Output 10.1
Three Different
Meanings of
BUDGET.SUM.

Summary of Departmental Budgets			1
DEPT	ACCOUNT	BUDGET	
Equipment	lease	\$80,000.00	
	maint	\$22,000.00	
	purchase	\$60,000.00	
	rental	\$10,000.00	
	sets	\$15,000.00	
	tape	\$20,000.00	
-----		-----	
Equipment		\$207,000.00	
Facilities	rent	\$48,000.00	
	supplies	\$5,500.00	
	utils	\$8,500.00	
-----		-----	
Facilities		\$62,000.00	
Other	advert	\$60,000.00	
	musicfee	\$8,000.00	
	talent	\$33,000.00	
-----		-----	
Other		\$101,000.00	
Staff	fulltime	\$295,000.00	
	parttime	\$100,000.00	
-----		-----	
Staff		\$395,000.00	

(continued)

Travel	gas	\$2,000.00
	leases	\$8,000.00

Travel		\$10,000.00
=====		=====
Total:		\$775,000.00
=====		=====

Here BUDGET.SUM takes on three different meanings:

- In detail rows, the value is the budget for one account in a department. For example, the first detail row of the report shows that the budget for the lease account in the equipment department is \$80,000.00.
- In the group summary lines, the value is the budget for one department. For example, the first group summary line shows that the budget for the whole equipment department is \$207,000.00.
- In the report summary line, the value (\$775,000.00) is the budget for the entire company.

Building a Report that Uses Groups and a Report Summary

The report in Output 10.2 contains five columns:

- DEPT is a group variable.
- BUDGET and ACTUAL are analysis variables used to calculate the SUM statistic.
- BALANCE is a computed variable based on BUDGET and ACTUAL.
- The N statistic indicates how many observations each row represents.

At the end of the report a break summarizes the statistics and computed variables in the report and assigns the value of **TOTAL** to DEPT.

The following statements produce Output 10.2:

```
proc report data=budget2 nowindows headline headskip;
  column dept budget actual balance n;

  define dept    / group;
  define budget  / analysis sum;
  define actual  / analysis sum;
  define balance / computed format=dollar11.2;

  compute balance;
    balance=budget.sum-actual.sum;
  endcomp;

  rbreak after / dol dul summarize;

  compute after;
```

```

dept='TOTAL';
endcomp;
run;

```

Output 10.2
*Report with Groups
and a Report
Summary*

Summary of Departmental Budgets				1
DEPT	BUDGET	ACTUAL	BALANCE	N
Equipment	\$207,000.00	\$208,429.21	\$-1,429.21	12
Facilities	\$62,000.00	\$60,627.13	\$1,372.87	6
Other	\$101,000.00	\$108,640.44	\$-7,640.44	6
Staff	\$395,000.00	\$393,857.51	\$1,142.49	4
Travel	\$10,000.00	\$8,456.99	\$1,543.01	4
=====	=====	=====	=====	=====
TOTAL	\$775,000.00	\$780,011.28	\$-5,011.28	32
=====	=====	=====	=====	=====

A description of how PROC REPORT builds this report follows:

1. PROC REPORT starts building the report by consolidating the data (DEPT is a group variable) and calculating the statistics (BUDGET.SUM, ACTUAL.SUM, and N) for each detail row and for the break at the end of the report. It stores these values in a temporary file.
2. Now, the REPORT procedure is ready to start building the first row of the report. This report does not contain a break at the beginning of the report or a break before any groups, so the first row of the report is a detail row. The procedure initializes all report variables to missing, as Figure 10.1 illustrates. Missing values for a character variable are represented by a blank, and missing values for a numeric variable are represented by a period.

Figure 10.1
*Initializing Values
for the First Detail
Row*

DEPT	BUDGET	ACTUAL	BALANCE	N

3. Figure 10.2 illustrates the construction of the first three columns of the row. PROC REPORT fills in values for the row from left to right. Values come from the temporary file created at the beginning of the report-building process.

Figure 10.2
*Filling in Values
from Left to Right*

DEPT	BUDGET	ACTUAL	BALANCE	N
Equipment

DEPT	BUDGET	ACTUAL	BALANCE	N
Equipment	\$207,000.00	.	.	.

DEPT	BUDGET	ACTUAL	BALANCE	N
Equipment	\$207,000.00	\$208,429.21	.	.

- The next column in the report contains the computed variable BALANCE. When it gets to this column, PROC REPORT executes the statement in the compute block attached to BALANCE. This statement calculates the difference between BUDGET and ACTUAL:

```
balance=budget.sum-actual.sum;
```

The row now looks like Figure 10.3.

Note: PROC REPORT processes rows from left to right. Therefore, when you compute a variable, the variables on which you base the computation must be to the left of the computed variable in the COLUMN statement.

Figure 10.3
Adding a Computed Variable to the Row

DEPT	BUDGET	ACTUAL	BALANCE	N
Equipment	\$207,000.00	\$208,429.21	\$-1,429.21	.

- Next, PROC REPORT fills in the value for the N statistic. The value comes from the temporary file created at the beginning of the report-building process. Figure 10.4 illustrates the completed row.

Figure 10.4
First Complete Detail Row

DEPT	BUDGET	ACTUAL	BALANCE	N
Equipment	\$207,000.00	\$208,429.21	\$-1,429.21	12

- The procedure writes the completed row to the report.

7. PROC REPORT repeats steps 2, 3, 4, 5 and 6 for each detail row in the report.
8. At the break at the end of the report, PROC REPORT constructs the break lines described by the RBREAK statement. These lines include double underlining, double overlining, and a preliminary version of the summary line. The statistics for the summary line were calculated earlier (see step 1). Values for computed variables are calculated when PROC REPORT reaches the appropriate column, just as they are in detail rows. PROC REPORT uses these values to create the preliminary version of the summary line (see Figure 10.5).

Figure 10.5
Preliminary
Summary Line

DEPT	BUDGET	ACTUAL	BALANCE	N
	\$775,000.00	\$780,011.28	\$-5,011.28	32

9. If no compute block is attached to the break, the preliminary version of the summary line is the same as the final version. However, in this example, a compute block is attached to the break. Therefore, the next thing that PROC REPORT does is execute the statements in that compute block. In this case, the compute block contains one statement:

```
dept='TOTAL';
```

This statement replaces the value of DEPT, which in the summary line is missing by default, with the word **TOTAL**. After PROC REPORT executes the statement, it modifies the summary line to reflect this change to the value of DEPT. The final version of the summary line appears in Figure 10.6.

Figure 10.6
Final Summary Line

DEPT	BUDGET	ACTUAL	BALANCE	N
TOTAL	\$775,000.00	\$780,011.28	\$-5,011.28	32

10. Finally, PROC REPORT writes all the break lines—underlining, overlining, and the final summary line—to the report.

Building a Report that Uses DATA Step Variables

PROC REPORT initializes report variables to missing at the beginning of each row of the report. The value for a DATA step variable is initialized to missing before PROC REPORT begins constructing the rows of the report and remains missing until you specifically assign it a value. PROC REPORT retains the value of a DATA step variable from the execution of one compute block to another.

Because all compute blocks share the current values of all variables, you can initialize

DATA step variables at a break at the beginning of the report or at a break before a break variable. This report initializes one variable at a break at the beginning of the report and one variable at a break before DEPT.

► **Caution** *Timing at Breaks*

PROC REPORT creates a preliminary summary line for a break before it executes the corresponding compute block. If the summary line contains computed variables, the computations are based on the values of the contributing variables in the preliminary summary line. If you want to recalculate computed variables based on values you set in the compute block, you must do so explicitly in the compute block. This report illustrates this technique.

If no compute block is attached to a break, the preliminary summary line becomes the final summary line. ▲

The report in Output 10.3 contains five columns:

- DEPT and ACCOUNT are group variables
- BUDGET is an analysis variable used to calculate the SUM statistic
- PCTDEPT is a computed variable based on the values of BUDGET and a DATA step variable, DEPTBUD, which is the budget for the whole department.
- PCTCOMP is a computed variable based on the values of BUDGET and a DATA step variable, COMPBUD, which is the budget for the whole company.

At the beginning of the report, a customized report summary tells what the budget for the entire company is. At a break before each group of observations for a department, a default summary summarizes the data for that department. At the end of each group a break inserts a blank line.

The following statements produce Output 10.3.

Note: Calculations of percentages do not multiply their results by 100 because PROC REPORT prints them with the PERCENT. format.

```
proc report data=budget2 nowindows headline headskip;
  column dept account budget pctdept pctcomp;

  define dept      / group;
  define account  / group;
  define budget   / analysis sum;
  define pctdept  / computed format=percent9.2;
  define pctcomp  / computed format=percent9.2;

  compute before;
    compbud=budget.sum;
    pctcomp=budget.sum/compbud;
    line 'The budget for the company is ' compbud dollar11.2 '.';
    line ' ';
  endcomp;

  compute before dept;
    deptbud=budget.sum;
    pctdept=budget.sum/deptbud;
  endcomp;
```

```

compute pctdept;
  pctdept=budget.sum/deptbud;
endcomp;

compute pctcomp;
  pctcomp=budget.sum/compbud;
endcomp;

break before dept / ul ol summarize;

break after dept / skip;
run;

```

Output 10.3
Report with DATA
Step Variables

Summary of Departmental Budgets					1
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	

The budget for the company is \$775,000.00.					

Equipment		\$207,000.00	100.00%	26.71%	

Equipment	lease	\$80,000.00	38.65%	10.32%	
	maint	\$22,000.00	10.63%	2.84%	
	purchase	\$60,000.00	28.99%	7.74%	
	rental	\$10,000.00	4.83%	1.29%	
	sets	\$15,000.00	7.25%	1.94%	
	tape	\$20,000.00	9.66%	2.58%	

Facilities		\$62,000.00	100.00%	8.00%	

Facilities	rent	\$48,000.00	77.42%	6.19%	
	supplies	\$5,500.00	8.87%	0.71%	
	utils	\$8,500.00	13.71%	1.10%	

Other		\$101,000.00	100.00%	13.03%	

Other	advert	\$60,000.00	59.41%	7.74%	
	musicfee	\$8,000.00	7.92%	1.03%	
	talent	\$33,000.00	32.67%	4.26%	

Staff		\$395,000.00	100.00%	50.97%	

Staff	fulltime	\$295,000.00	74.68%	38.06%	
	parttime	\$100,000.00	25.32%	12.90%	

Travel		\$10,000.00	100.00%	1.29%	

Travel	gas	\$2,000.00	20.00%	0.26%	
	leases	\$8,000.00	80.00%	1.03%	

A description of how PROC REPORT builds this report follows:

1. PROC REPORT starts building the report by consolidating the data (DEPT and ACCOUNT are group variables) and calculating the statistics (BUDGET.SUM) for each detail row, for the break at the beginning of the report, for the breaks before each group, and for the breaks after each group. It stores these values in a temporary file.
2. PROC REPORT initializes all DATA step variables to missing (see Figure 10.7).

Figure 10.7 Initializing DATA Step Variables

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
					.	.

- Because this PROC REPORT step contains a COMPUTE BEFORE statement, the procedure constructs a preliminary summary line for the break at the beginning of the report. This preliminary summary line contains values for the statistic (BUDGET.SUM) and the computed variables (PCTDEPT and PCTCOMP).
At this break, BUDGET.SUM is the budget for the entire company. PROC REPORT takes the value for BUDGET.SUM from the temporary file that it created at the beginning of the report-building process.
The values for PCTDEPT and PCTCOMP come from executing the statements in the corresponding compute blocks. Because the values of DEPTBUD and COMPBUD are missing, PROC REPORT cannot calculate values for PCTDEPT and PCTCOMP. Therefore, in the preliminary summary line, these variables also have missing values (see Figure 10.8).

Note: The COMPUTE BEFORE statement creates a break at the beginning of the report. You do not need to use an RBREAK statement.

Figure 10.8 Preliminary Summary Line for the Break at the Beginning of the Report

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
		\$775,000.00

- PROC REPORT creates the final version of the summary line by executing the first two statements in the COMPUTE BEFORE block. These statements execute only once for the entire report.
 - The first statement assigns the value of BUDGET.SUM, which in that part of the report represents the budget for the entire company, to the variable COMPBUD.
 - The second statement completes the summary line by recalculating PCTCOMP from the new value of COMPBUD. Figure 10.9 shows the final summary line.

► **Caution** *Recalculating Values in the Final Summary Line*

If you do not recalculate the value for PCTCOMP, it will be incorrect because it will be based on the preliminary value of COMPBUD rather than the final value that you assign in the compute block. ▲

Figure 10.9 Final Summary Line for the Break at the Beginning of the Report

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
		\$775,000.00	.	100.00%	.	\$775,000.00

5. Because the program does not include an RBREAK statement with the SUMMARIZE option, PROC REPORT does not write the final summary line to the report. Instead, it uses the third and fourth lines in the COMPUTE BEFORE block to write a customized summary.
 - The third statement (the first LINE statement) writes the text between quotation marks and the value of COMPBUD (with the DOLLAR11.2 format). It takes the value of COMPBUD from the value in the final summary line.
 - The fourth statement (the second LINE statement) writes a blank line.
6. Next, PROC REPORT constructs a preliminary summary line for the break before the first group of observations. (This break both uses the SUMMARIZE option in the BREAK statement and has a compute block attached to it. Either of these conditions generates a summary line.) The preliminary summary line contains values for the break variable (DEPT), the statistic (BUDGET.SUM), and the computed variables (PCTDEPT and PCTCOMP). At this break, BUDGET.SUM is the budget for one department (**Equipment**). PROC REPORT takes the values for BUDGET.SUM and DEPT from the temporary file that it created at the beginning of the report-building process.

The values for PCTDEPT and PCTCOMP come from executing the statements in the corresponding compute blocks. Because the value of DEPTBUD is missing, PROC REPORT cannot calculate a value for PCTDEPT. Therefore, in the preliminary summary line, PCTDEPT has a missing value (see Figure 10.10).

Figure 10.10 Preliminary Summary Line for the Break before the First Group of Observations

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Equipment		\$207,000.00	.	26.71%	.	\$775,000.00

7. PROC REPORT creates the final version of the summary line by executing the statements in the COMPUTE BEFORE DEPT compute block. These statements execute once each time the value of DEPT changes.
 - The first statement assigns the value of BUDGET.SUM, which in that part of the

report represents the budget for one department, to the variable DEPTBUD.

- The second statement completes the summary line by recalculating PCTDEPT from the new value of DEPTBUD. Figure 10.11 shows the final summary line.

Figure 10.11 Final Summary Line for the Break before the First Group of Observations

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Equipment		\$207,000.00	100.00%	26.71%	\$207,000.00	\$775,000.00

8. Because the program contains a BREAK BEFORE statement with the SUMMARIZE option, PROC REPORT writes the final summary line to the report. The OL and UL options in the BREAK statement overline and underline the summary line.
9. Now, the REPORT procedure is ready to start building the first detail row of the report. It initializes all report variables to missing. Values for DATA step variables do not change. Figure 10.12 illustrates the first detail row at this point.

Figure 10.12 Initializing Values for the First Detail Row

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
		.	.	.	\$207,000.00	\$775,000.00

10. Figure 10.13 illustrates the construction of the first three columns of the row. PROC REPORT fills in values for the row from left to right. The values come from the temporary file it created at the beginning of the report-building process.

Figure 10.13 Filling in Values from Left to Right

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Equipment		.	.	.	\$207,000.00	\$775,000.00

DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Equipment	lease	.	.	.	\$207,000.00	\$775,000.00

DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Equipment	lease	\$80,000.00	.	.	\$207,000.00	\$775,000.00

11. The next column in the report contains the computed variable PCTDEPT. When it gets to this column, PROC REPORT executes the statement in the compute block attached to PCTDEPT. This statement calculates the percentage of the department’s budget that is allocated to the current account:

```
pctdept=budget.sum/deptbud;
```

The row now looks like Figure 10.14.

Figure 10.14 Adding the First Computed Variable to the Row

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Equipment	lease	\$80,000.00	38.65%	.	\$207,000.00	\$775,000.00

12. The next column in the report contains the computed variable PCTCOMP. When it gets to this column, PROC REPORT executes the statement in the compute block attached to PCTCOMP. This statement calculates the percentage of the company’s budget that is allocated to the current account:

```
pctcomp=budget.sum/compbud;
```

The first detail row is now complete (see Figure 10.15).

Figure 10.15 First Complete Detail Row

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Equipment	lease	\$80,000.00	38.65%	10.32%	\$207,000.00	\$775,000.00

13. PROC REPORT writes the detail row to the report. It repeats steps 9, 10, 11, and 12 for each detail row in the group.
14. After writing the last detail row in the group to the report, PROC REPORT constructs the default group summary. Because no compute block is attached to this break and because the BREAK AFTER statement does not include the SUMMARIZE option, PROC REPORT does not construct a summary line. The only action at this break is that the SKIP option in the BREAK AFTER statement writes a blank line after the last detail row of the group.
15. Now the value of the break variable changes from **Equipment** to **Facilities**. PROC REPORT constructs a preliminary summary line for the break before this group of observations. This summary line contains values for the break variable (DEPT), the statistic (BUDGET.SUM), and the computed variables (PCTDEPT and PCTCOMP). At this break, BUDGET.SUM is the budget for the Facilities department. Because the COMPUTE BEFORE DEPT block has not yet executed, the value of DEPTBUD is still \$207,000.00, the value for the Equipment department. Thus, the value that PROC REPORT calculates for PCTDEPT in this preliminary summary line is incorrect (see Figure 10.16). The statements in the compute block for this break calculate the correct value (see the following step).

Figure 10.16 Preliminary Summary Line for the Break before the Second Group of Observations

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Facilities		\$62,000.00	29.95%	8.00%	\$207,000.00	\$775,000.00

► **Caution** **Synchronizing Values for Computed Variables in Break Lines**

If the PROC REPORT step does not recalculate PCTDEPT in the compute block attached to the break, the value in the final summary line will not be synchronized with the other values in the summary line, and the report will be incorrect. ▲

16. PROC REPORT creates the final version of the summary line by executing the statements in the COMPUTE BEFORE DEPT compute block. These statements execute once each time the value of DEPT changes.

- The first statement assigns the value of BUDGET.SUM, which in that part of the report represents the budget for the Facilities department, to the variable DEPTBUD.
- The second statement completes the summary line by recalculating PCTDEPT from the new, appropriate value of DEPTBUD. Figure 10.17 shows the final summary line.

Figure 10.17 Final Summary Line for the Break before the Second Group of Observations

Report Variables					DATA Step Variables	
DEPT	ACCOUNT	BUDGET	PCTDEPT	PCTCOMP	DEPTBUD	COMPBUD
Facilities		\$62,000.00	100.00%	8.00%	\$62,000.00	\$775,000.00

Because the program contains a BREAK BEFORE statement with the SUMMARIZE option, PROC REPORT writes the final summary line to the report. The OL and UL options in the BREAK statement overline and underline the summary line.

17. Now, the REPORT procedure is ready to start building the first row for this group of observations. It repeats steps 8 through 16 until it has processed all observations in the input data set (stopping with step 14 for the last group of observations).

Glossary

across variable

a variable used so that each formatted value of the variable forms a column in the report. If the variable does not have a format, each value forms a column.

alias

an alternate name for a report item. An alias enables you to set different characteristics for the same item when you use the item in more than one way in the report.

analysis variable

a numeric variable used to calculate statistics. You must associate a statistic with an analysis variable. You specify the statistic in the DEFINE statement. In conjunction with group variables, an analysis variable can consolidate multiple observations that have a unique combination of values for all group variables into one row. The value displayed for an analysis variable is the statistic associated with it calculated for the set of observations represented by that row and column of the report.

By default, the REPORT procedure treats a numeric variable as an analysis variable used to calculate the SUM statistic.

break

a section of the report that does one or more of the following: visually separates parts of the report; summarizes statistics and computed variables; displays text, values calculated for a set of rows of the report, or both; executes DATA step statements. You can create breaks when the value of a selected variable changes or at the beginning or end of a report. See also break variable.

break line

a line of a report that contains one of the following: characters that visually separate parts of the report; summaries of statistics and computed variables (called a summary line); text, values calculated for a set of rows of the report, or both.

break variable

a group or order variable you select to determine the location of break lines. The REPORT procedure performs the actions you specify for the break each time the value of this variable changes.

cell

a single unit of a table produced by a SAS procedure, such as the REPORT procedure.

compound name

a name that has the following form:

variable-name.statistic

A compound name refers to the value of a statistic.

compute block

programming statements that the REPORT procedure executes either at a specified location or on every row of the report. A compute block begins with a COMPUTE statement and ends with an ENDCOMP statement.

computed variable

a variable whose value is calculated by statements in a compute block.

DATA step variable

a variable that appears in one or more compute blocks but does not appear in a column of the report.

detail row

a row of a report that either contains information from a single observation in the data set or consolidates the information for a group of observations that have a unique combination of values for all group variables.

display variable

in the REPORT procedure, a variable that does not affect the order of the rows of the report. A report that contains one or more display variables has a detail row for each observation in the data set.

By default, the REPORT procedure treats character variables as display variables.

group variable

a variable that orders the detail rows in a report according to their formatted values and consolidates multiple observations that have a unique combination of values for all group variables into one row.

header

a string of characters that spans the top of one or more columns in the report. A header can occupy multiple lines. See also split character.

item

a data set variable, a statistic, or a computed variable. An item can occupy one or more columns in a report. Under some circumstances, multiple items can share a column.

order variable

a variable that orders the detail rows in a report according to their formatted values. A report that contains one or more order variables has a detail row for every observation in the data set.

panel

a set of columns in a report. A familiar example of a report with panels is a telephone book, which contains multiple panels of names and telephone numbers on a single page.

report break

a break at the beginning or end of a report. See also break.

report variable

a variable that constitutes one or more columns of the report. The variable may or may not appear in one or more code segments.

You can suppress the display of a report variable with the NOPRINT or the NOZERO option in the DEFINE statement.

SAS data library

a collection of one or more SAS files that are recognized by the SAS System and that are referenced and stored as a unit. Each file is a member of the library.

split character

a character that splits headers across multiple lines. If you use the split character in a column header, the REPORT procedure breaks the header when it reaches that character and continues the header on the next line. The split character itself is not part of the column header.

summary line

in the REPORT procedure, a break line that summarizes statistics and computed variables.

Index

A

- ACROSS option, DEFINE statement 202–203
 - creating column headers 19, 224–225
 - purpose and use 6, 202–203
 - required for using ORDER= option 14
- across variables
 - adding computed variables based on shared columns 82–84
 - analysis variable above 203
 - combining observations counts 30–31
 - counting observations for each value 30
 - default sort order 206
 - locating before or after analysis variable 19
 - N statistic as default 19, 202
 - no items above or below 202
 - placing statistics above or below 25
 - referencing in compute blocks 77
 - separating from analysis variables in column specifications 19
 - sharing column with multiple analysis variables 21
 - sharing column with one analysis variable 20
 - sharing column with other items 19, 213
- adding computed variables 75–86
 - See also computed variables
 - character variable 85–86
 - column not shared with computed variable 82–83
 - column shared with computed variable 83–84
 - computed from group, order, computed, or display variables 78–79
 - computed from statistic 80–81
 - using compute block 230
- AFTER argument
 - BREAK statement 89–90, 188, 255
 - compute blocks 200
 - COMPUTE statement 200
 - RBREAK statement 132, 217
- aliases
 - assigning to variables 54–55
 - defining 197
 - invalid in compute blocks 77, 92, 135, 197
 - specifying for statistics 27, 27–28
 - syntax 27, 54
 - using in column headers 196–197
- ANALYSIS option, DEFINE statement
 - purpose and use 6, 203–205
 - statistics required 7
- analysis variables
 - calculating percentages over all groups 169–170
 - calculation at breaks 88, 190, 219
 - grouping observations 223–224
 - numeric variable as default 7
 - placing across variable before or after
 - analysis variable 19
 - placing in column headers 19, 21
 - placing statistics above or below 25
 - referencing in compute blocks 77, 92, 135, 199
 - reporting multiple statistics for one variable 27–28
 - separating from across variables in column specifications 19
 - sharing column with multiple analysis variables 21
 - sharing column with one analysis variable 20
 - sharing columns with other items 213
 - statistics required 16, 210
 - SUM statistic as default statistic 203, 210
 - using with group variable 203–204
 - using with order variable 204–205
 - value when no group is present 6
 - values at beginning or end of reports 132
 - weighting 220
- ascending order as default for detail rows 208
- assignment statements
 - assigning text to character variables in summary lines 109–110, 146–148
 - calculating cumulative percentages for whole report 176–177
 - calculating cumulative percentages within groups 173–174
 - calculating percentages over all groups 169–170
 - calculating percentages within groups 166–167
 - controlling space required for writing character variables 114–116
 - controlling space required for writing numeric variables 117–119
 - initializing DATA step variables 153–155
 - initializing variables for each group 122–124
- asterisk (*)
 - expanding to fill space over column 66, 196, 206
- at (@) pointer control
 - controlling horizontal position of text between groups 120–121
 - controlling horizontal position of text in report summaries 151–152
- attributes, specifying 193
- automatic variables
 - _COL_ 193
 - _LAST_ 182

B

BEFORE argument
 BREAK statement 89-90, 188, 256
 COMPUTE statement 173, 176, 200, 251, 252, 253
 RBREAK statement 132, 217
 BESTw. format 207
 blank characters
 between columns 44-45, 186, 210
 between panels 186
 controlling space for writing character variables 114-116
 controlling space for writing numeric variables 117-119
 separating analysis variables in column specifications 19
 blank lines, inserting
 after summary line at beginning of report 142-143
 beneath column headers 70
 between groups 97-98
 multiple lines after summary line at beginning of report 144-145
 multiple lines between groups 99-100
 using LINE statement 99, 112-113, 149
 BLINK attribute 193
 break lines
 See also group summaries
 See also report summaries
 See also summary lines
 analysis variable used with group variable 203-204
 construction by REPORT procedure 243-244, 248, 251
 creating group summaries 87-88
 definition 87, 131
 order of appearance 191, 219
 purpose and use 188
 suppressing underlining and overlining 190
 variable values at beginning or end of reports 132
 BREAK statement 188-191
 AFTER argument 89-90, 188, 255
 BEFORE argument 89-90, 188, 256
 COLOR= option 191
 creating default group summaries 231-232
 creating group summaries 89-90
 DOL option 103, 189
 DUL option 105, 189
 information for windowing environment 191
 not required with COMPUTE statement 200
 OL option 103-104, 189, 232, 253
 order of break lines 191
 PAGE option 93-94, 189
 requirements 188-189
 SKIP option 97-98, 189, 231-232, 255
 specifying break variable 89, 189
 SUMMARIZE option 101-102, 189-190, 232, 252, 253, 256
 SUPPRESS option 107-108, 190, 232
 syntax 188
 UL option 105-106, 190-191, 232, 253
 break variables
 calculation for summary lines 190

group or order variable required 188
 missing values for character variables on right side 88
 setting to missing 128
 specifying in BREAK statement 89, 189
 specifying in COMPUTE statement 201
 suppressing 190
 breaks
 See also break lines
 definition 131
 synchronizing computed variables during report construction 249, 255-256
 building of reports by REPORT procedure
 See report construction by REPORT procedure
 BY line, suppressing 95-96
 BY statement 191-192
 BY processing requirements 192
 compared with group summary for producing page breaks 93
 DESCENDING option 192
 information for windowing environment 192
 NOTSORTED option 192
 omitted from stored report definitions 56, 184
 syntax 191
 BY variables
 methods of ordering 192
 using group variable as BY variable 93, 94-96
 #BYLINE specifications 96
 #BYVAL specifications 96
 #BYVAR specifications 96

C

calculating percentages 163-177
 cumulative percentages for whole report 175-177
 cumulative percentages within groups 172-174
 effect of report construction 163
 initializing variables 164-165
 missing in preliminary summary line during report construction 252
 over all groups 169-171
 recalculating during report construction 254, 256
 recalculating during report construction (caution) 251
 using compound names 164
 within groups 166-168
 calculation of computed variables
 breaks between groups 88
 during report construction 248
 position of variables (caution) 77
 recalculating (synchronizing) during report construction 249, 255-256
 summary lines 190, 219
 calculation of statistics
 break lines 88
 default for analysis variable 203

- during report construction 246, 247, 250
- summary lines 190, 219
- calculation of summary lines
 - REPORT procedure 190, 219, 248
 - statistics associated with analysis variable 203
- CALL DEFINE statement 192-193
 - compute block requirements 192
 - requirements 192-193
 - running report in windowing environment 239-240
 - syntax 192
- canceling REPORT procedure 1
- catalog entries, specifying
 - OUTREPT= option 184
 - report definitions 186
- CATALOG procedure 185
- CATALOG window 185
- catalogs, specifying
 - OUTREPT= option 184
 - profiles 185
 - report definitions 186
- cell
 - definition 19
- CENTER option, DEFINE statement 205
- CENTER option, REPORT statement
 - centering reports 181-182
 - effect on column spacing 186, 210
- CENTER system option
 - centering reports 181-182
 - centering summary text 111, 120, 148, 151
 - not affected by DEFINE statement CENTER option 205
- CHARACTER option, COMPUTE statement
 - not required with LENGTH= option 75
 - purpose and use 201
 - specifying character variables 75
- character variables
 - assigning length 125, 156
 - assigning text in summary lines 146-148
 - changing value in summary lines 109-110
 - controlling space required for writing 114-116
 - default as display variables 7
 - including in computed variables 85-86
 - listing all observations and all variables 8-9
 - missing values when at right side of break variable 88
 - relationship with column width 182
 - using in compute blocks 75
 - values at beginning or end of reports 132
 - wrapping values across multiple rows 52-53
- CLASS statement 16
- _COL_ automatic variable 193
- collapsing multiple observations
 - See consolidating multiple observations into one row
- COLOR attribute 193
- COLOR= option
 - BREAK statement 191
 - DEFINE statement 214
 - RBREAK statement 220
- column headers
 - adding blank line beneath column headers 70
- commas for separating multiple report items 19, 195
- controlling line breaks 64-65
- creating with ACROSS option 19, 20, 21, 224-225
- customizing 228
- customizing text 62-63
- default behavior of REPORT procedure 61
- defining for statistics 27-28
- expanding characters to fill column headers 66-69, 196, 206, 228
- inequality signs in extended headers 68-69
- parentheses for specifying multiple items 19, 195-196
- quotation marks for controlling line breaks 64-65
- sharing column with analysis variable 20
- sharing column with multiple analysis variables 21
- showing frequency counts 19-20
- spanning multiple columns with one header 73-74, 196
- special characters in extended headers 66-69, 196, 206
- specifying statistics 194
- specifying with DEFINE statement 205-206
- split character in headers 196, 205
- split characters in headers 64-65
- stacking multiple items 19-22, 195
- statistics as default headers 61
- suppressing variable labels 61, 206
- suppressing with NOHEADER option 184
- underlining headers and spaces between columns 71
- underlining headers but not spaces between columns 72
- underscore character in extended headers 66-67
- using macro variables in headers 196
- variable labels used by default 206
- column name
 - referencing variables in compute blocks 76, 92, 134, 192
 - specifying in CALL DEFINE statement 192
- column number
 - referencing variables in compute blocks 76, 92, 134, 199
 - specifying in CALL DEFINE statement 193
- column spacing
 - controlling spaces between columns 44-45, 186
 - effect of CENTER option 186, 210
 - specifying spacing for all columns 44
 - specifying spacing for one column 45
 - specifying with SPACING= option 44, 186, 210
- COLUMN statement 193-197
 - adding computed variables 78, 80, 82, 85
 - aliases for statistics 27, 27-28, 196-197
 - calculating cumulative percentages for whole report 175-177
 - calculating cumulative percentages within groups 172-174
 - calculating percentages over all groups 169-170

- COLUMN statement (continued)**
 calculating percentages within groups
 166-167
 creating column headers 224-225
 customizing column headers 228
 excluded by use of REPORT = option 59,
 186, 194
 multiple report items in parentheses
 195-196
 multiple report items separated by commas
 195
 ordering rows 222-223
 purpose 5
 reporting multiple statistics for one
 variable 27, 27-28
 requirements 194-197
 single report item 194
 single report item and alias 196-197
 spanning multiple columns with customized
 column header 73-74
 specifying and ordering columns 11-12, 221
 specifying statistics 25, 26, 29
 syntax 194
- column width**
 BESTw. format 207
 default width 207
 relationship with variable type (table) 182
 size determination for shared items 21, 211
 specifying with COLWIDTH= option 182
 specifying with WIDTH= option 48,
 48-49, 182
 \$w. format 207
- columns**
 See also COLUMN statement
 See also panels
 adding computed variables based on shared
 columns 82-84
 conflicts caused by multiple items 213
 identifying current column with _COL_
 automatic variable 193
 selecting observations for processing 42-43
 sharing with across and analysis variables
 20, 21
 specifying and ordering columns 11-12, 221
 stacking multiple items in columns 19-22,
 195
 suppressing with NOPRINT option 50-51,
 208, 237
- COLWIDTH= option, REPORT statement**
 default column width 207
 purpose and use 182
- combining variables**
 See computed variables
- comma (,)**
 separating analysis and across variables 19
 separating report items 195
 separating statistics from variables 25
- COMMAND attribute 193**
- COMMAND option, REPORT statement 187**
- compound names**
 calculating percentages within groups
 166-167
 different meanings in different parts of
 report 164-165, 244-245
 initializing variables in compute blocks
 164-165
 referencing variables in compute blocks 76,
 92, 134, 199
 referring to both statistic and variable 216
- compute blocks**
 adding variables to report 75-77, 230
 AFTER argument 200
 aliases not valid 77, 92, 135, 197
 assigning text to character variables in group
 summaries 109-110
 assigning text to character variables in report
 summaries 146-148
 BEFORE argument 200
 calculating cumulative percentages 173-174,
 176-177
 calculating percentages 166-167, 169-170
 calculation at breaks 88
 capabilities 198
 controlling execution of statements between
 groups 128-130
 controlling execution of statements in report
 summaries 159-161
 controlling execution with conditional
 statements 123
 creating customized group summaries
 91-92, 234-235
 creating customized report summaries
 133-135, 252
 definition 198
 execution by REPORT procedure 193, 200,
 247, 254, 255
 initializing DATA step variables 248-249
 initializing variables at beginning of report
 153-155
 initializing variables for each group 122-124
 inserting multiple blank lines between
 groups 99-100
 inserting multiple lines after summary line at
 beginning of report 144-145
 LINE statement supported 91
 placing conditional text between groups 126
 position of LINE statement 99
 referencing report variables 76-77, 92,
 134-135, 199
 REPORT procedure features supported 198
 SAS language features supported 76, 134,
 198
 specifying text and variable values between
 groups 111-112
 specifying text or variable values at beginning
 or end of report summa 148-150
 structure 75
 timing at breaks (synchronizing computed
 variables) 249, 255-256
 using CALL DEFINE statement 192
 variable values at beginning or end of
 reports 132
- COMPUTE statement 198-201**
 See also compute blocks
 AFTER argument 200
 BEFORE argument 173, 176, 200, 251,
 252, 253
 BREAK and RBREAK statements not
 required 200

- capabilities 198
 - CHARACTER option 75, 201
 - compute block processing 200
 - creating customized group summaries 234-235
 - LENGTH= option 75, 201
 - referencing data set variables in compute blocks 199
 - requirements 200
 - specifying break variable 201
 - specifying location 200
 - specifying variable, computed variable, or statistic 200
 - syntax 198-199
 - COMPUTED option, DEFINE statement
 - adding computed variables 78, 80, 82, 85
 - purpose and use 206
 - computed variables
 - See also compute blocks
 - See also computed variables, adding
 - calculating percentages within groups 166-167
 - calculation at breaks 88
 - calculation during report construction 248
 - calculation for summary lines 190, 219
 - position related to calculation (caution) 77
 - recalculating (synchronizing) during report construction 249, 255-256
 - referencing in compute blocks 77, 92, 135, 199
 - relationship with column width 182
 - set to missing if positioned incorrectly 77
 - sharing column with other items 213
 - specifying with COMPUTED option 206
 - values at beginning or end of reports 132
 - computed variables, adding 75-86
 - character variable 85-86
 - column not shared with computed variable 82-83
 - column shared with computed variable 83-84
 - computed from group, order, computed, or display variables 78-79
 - computed from statistic 80-81
 - report construction process 254
 - using compute block 230
 - conditional statements
 - compute block execution in group summaries 123
 - compute block execution in report summaries 154, 161
 - execution of statements between groups 128-130
 - execution of statements in report summaries 159-161
 - placing conditional text at report beginning or end 156-158
 - storing text in character variables 125-127
 - consolidating multiple observations into one row 16-18
 - groups based on value of multiple variables 17-18
 - groups based on value of one variable 16-17
 - construction of reports by REPORT procedure
 - See report construction by REPORT procedure
 - CSS statistic 24, 211
 - cumulative percentages, calculating
 - for whole report 175-177
 - within groups 172-174
 - customized summaries, creating
 - See also group summaries
 - See also report summaries
 - group summaries 90-92
 - report summaries 133-135, 249, 252
 - customizing column headers 61-74
 - See also column headers
 - adding blank line beneath column headers 70
 - controlling line breaks 64-65
 - example 228
 - extending column headers to fill column 66-69
 - inequality signs in extended headers 68-69
 - quotation marks for controlling line breaks 64-65
 - spanning multiple columns with one header 73-74
 - special characters in extended headers 67-68
 - text 62-63
 - underlining headers and spaces between columns 71
 - underlining headers without underlining spaces between columns 72
 - underscore character in extended headers 66-67
 - using split character 64-65
 - CV statistic 24, 211
- D**
- DATA= option, REPORT procedure
 - omitted from stored report definitions 56, 184
 - DATA= option, REPORT statement
 - purpose and use 182
 - specifying input data set 1
 - when to use 8
 - data set variables
 - See also variables
 - controlling with usage options 6, 7
 - default usage 7
 - listing all observations and all variables 8-10
 - referencing in compute blocks 199
 - using as column headers 19-22
 - data sets
 - See SAS data sets
 - DATA step variables
 - compared with report variables 122, 153
 - definition 164
 - initialized to missing before report construction 248, 250-251
 - initializing at beginning of report 153-155, 165
 - initializing for each group 122-124

- DATA step variables (*continued*)
 - initializing in compute blocks 164–165, 248–249
 - report construction by REPORT procedure 248–256
 - retaining cumulative values for percentages 174, 177
- DATA steps
 - calculating cumulative percentages for whole report 175–177
 - calculating cumulative percentages within groups 172–174
 - controlling space required for writing numeric variables 117–119
 - controlling space used to write character variables 114–116
- DATA value, ORDER= option 13, 209
- dates
 - ordering rows by SAS dates 14–15
- default group summaries 89–90, 231–232, 255
- default report summaries 132–133, 136, 233–234, 249
- DEFINE statement 201–214
 - See also CALL DEFINE statement
 - ACROSS option 6, 14, 19, 202–203, 224–225
 - aliases for statistics 27, 27–28
 - aliases for variables 54, 55
 - aliases in column headers 196–197
 - ANALYSIS option 6, 7, 203–205
 - calculating cumulative percentages for whole report 176–177
 - calculating percentages over all groups 169–170
 - calculating percentages within groups 166–167
 - CENTER option 205
 - COLOR= option 214
 - column-header option 205–206
 - COMPUTED option 78, 80, 82, 85, 206
 - controlling variables with usage options 6, 7
 - creating column headers with ACROSS option 224–225
 - customizing column headers 228
 - DESCENDING option 13, 206, 222–223
 - DISPLAY option 6, 206
 - FLOW option 52–53, 186, 206–207, 236
 - FORMAT= option 46, 207
 - GROUP option 6, 14, 16–18, 88, 207–208, 223–224
 - grouping observations 223–224
 - information for windowing environment 214
 - interactions of position and usage 211–213
 - ITEMHELP= option 214
 - LEFT option 208
 - NOPRINT option 50, 50–51, 208, 237
 - NOZERO option 208
 - options overridden by SHOWALL option 186
- DEFINE statement
 - ORDER= option 13–15, 209–210, 222–223
 - ORDER option 6, 13, 14, 15, 208–209, 222–223
 - ordering rows 222–223
 - PAGE option 210
 - purpose 5, 201
 - requirements 202
 - RIGHT option 210
 - SPACING= option 44, 186, 210
 - specifying statistics (caution) 25, 29
 - statistic option 210–211
 - syntax 201–202
 - when to use usage options 6, 7
 - WIDTH= option 21, 48, 48–49, 182, 211, 236
- DESCENDING option, BY statement 192
- DESCENDING option, DEFINE statement
 - ordering rows 222–223
 - purpose and use 13, 206
- descriptive statistics
 - See statistics
- detail rows
 - See also rows, ordering
 - ascending order as default order 208
 - consolidating multiple observations 16–18
 - construction by REPORT procedure 253–255
 - default order 208
 - ordering by GROUP option 207
 - values filled in from left to right during report construction 246, 247, 253–254
 - values initialized to missing during report construction 246
 - wrapping character variable values across multiple rows 52–53
- DISPLAY option, DEFINE statement 6, 206
- display variables
 - adding computed variables based on display variables 78–79
 - calculation for summary lines 190
 - character variable as default 7
 - group variables not used in same report 190
 - placing statistics above or below 25
 - referencing in compute blocks 77, 92, 135, 199
 - sharing columns with other items 213
 - specifying numeric variable in reports with no character variables 8
 - specifying with DISPLAY option 206
 - statistics required for using in groups 88, 92
- DMS system option 185, 187
- DOL option, BREAK statement
 - OL honored if both OL and DOL specified 103
 - overlining summary information 103–104
 - purpose and use 189
- DOL option, RBREAK statement
 - creating default report summary 234
 - OL honored if both OL and DOL specified 138
 - overlining summary information 138–139
 - purpose and use 218
- DOLLAR11. format 46
- DOLLAR11.2 format 46
- double overline character (=)
 - overlining summary information 103

- specifying with DOL option 189, 218
- suppressing in break lines 190
- double underline character (=)
 - specifying with DUL option 189, 218
 - suppressing in break lines 190
 - underlining summary information 105
- DROP= option, REPORT statement 11
- DUL option, BREAK statement
 - purpose and use 189
 - UL honored if both UL and DUL specified 105
 - underlining summary information 105-106
- DUL option, RBREAK statement
 - creating default report summary 234
 - purpose and use 218
 - UL honored if both UL and DUL specified 140
 - underlining summary information 140-141

E

- ENDCOMP statement
 - syntax 214
 - terminating compute blocks 198, 200
- enhancing reports
 - See report enhancements
- entries
 - See catalog entries, specifying
- equal sign (=)
 - expanding to fill space over column 66, 196, 206
 - representing double overlining character 103, 138, 189, 218
 - representing double underlining character 105, 140, 189, 218
- EXECUTE command 238-240
- expanding column headers 66-69
 - expanding characters to fill space over column 228
 - inequality signs in extended headers 68-69
 - spanning multiple columns with one header 73-74, 196
 - special characters in dateval. format 67-68
 - underscore character in extended headers 66-67
 - using special characters 66-69, 196, 206

F

- FIRST. variable 192
- FLOW option, DEFINE statement
 - purpose and use 206-207
 - split character supported 186, 206
 - wrapping character variable values across multiple rows 52, 52-53, 236
- FOOTNOTE statement
 - adding footnotes to reports 36-37
 - omitted from stored report definitions 56, 184
- FORMAT= option, DEFINE statement 46, 207
- FORMAT statement 46-47

- formats
 - BESTw. format 207
 - DOLLAR11. 46
 - DOLLAR11.2 46
 - PERCENT. 249
 - PERCENT8.1 167, 170
 - required for variables in LINE statement 148
 - specifying for statistics 27
 - specifying with FORMAT= option 46, 207
 - specifying with FORMAT statement 46
 - using special characters in dateval. format 67-68
 - \$VARYING. 114, 116
 - \$w. format 207
- FORMATTED value, ORDER= option 13, 209
- FORMCHAR= system option 182
- FREQ statement 214-215
 - omitted from stored report definitions 56, 184
 - requirements 214
 - syntax 214
 - using with WEIGHT statement 214
- FREQ value, ORDER= option 13, 209
- frequency counts
 - See also N statistic
 - combining observations counts 30-31
 - counting all observations in one row 29
 - counting observations for each value of across variable 30
 - showing frequency counts 19-20

G

- grand totals
 - See group summaries
 - See report summaries
- greater than sign (>)
 - expanding to fill space over column 66, 196, 206
 - extending column headers with inequality signs 68-69
- GROUP option, DEFINE statement 207-208
 - compared with CLASS statement 16
 - creating groups 88
 - grouping observations 223-224
 - groups based on value of multiple variables 17-18
 - groups based on value of single variable 16-17
 - ordering of detail rows 207
 - purpose and use 6, 16, 207-208
 - required for using ORDER= option 14
- group summaries
 - changing value of character variable 109-110
 - compared with BY variable for producing page breaks 93
 - controlling execution of statements between groups 128-130
 - creating customized summaries 90-92
 - creating default group summaries 89-90, 231-232

- group summaries (*continued*)
 - creating with breaks 87-88
 - overlining summary information 103-104
 - report construction process 243-244, 252-253, 255-256
 - summarizing variables and statistics 101-102
 - suppressing value of group variable 107-108
 - types of summaries 87
 - underlining summary information 105-106
 - using group variable as BY variable 94-96
 - group variables
 - adding computed variables based on group variables 78-79
 - analysis variable used with 203-204
 - calculating percentages within groups 166-167
 - calculation for summary lines 190
 - creating group for missing values 226-227
 - default order 16, 88
 - display variable not used in same report 190
 - groups based on value of multiple variables 17-18
 - groups based on value of single variable 16-17
 - inability to share columns with other items 213
 - multiple group variables 16
 - order variable not used in same report 88, 92, 188
 - ordering of detail rows 207-208
 - referencing in compute blocks 77, 92, 135, 199
 - restrictions on group and order variables 207
 - suppressing value in summary information 107-108
 - used as break variable 188
 - using as BY variable 94-96
 - groups
 - See also group summaries
 - See also group variables
 - calculating cumulative percentages 172-174, 175-177
 - calculating percentages 166-168, 169-171
 - calculation of variables at breaks 88
 - consolidation by REPORT procedure 16, 88, 207, 245-248, 250
 - controlling execution of statements in compute block 128-130
 - controlling horizontal position of text between groups 120-121
 - controlling space used for character values 114-116
 - controlling space used for numeric values 117-119
 - creating 88-89
 - default order 16, 88
 - default statistic for analysis variables 88
 - definition 16, 88
 - display variable without statistics not allowed 88, 92
 - initializing variables for each group 122-124
 - inserting blank lines between groups 97-98
 - inserting multiple blank lines between groups 99-100
 - order variable not used in same report 88, 92, 188
 - placing conditional text between groups 125-127
 - producing separate page for each group 93-96
 - report construction process 245-248
 - specifying text and variable values between groups 111-113
 - using group variable as BY variable 94-96
- ## H
- headers
 - See column headers
 - HEADLINE option, REPORT statement 71, 182
 - HEADSKIP option, REPORT statement 70, 182
 - HELP= option, REPORT statement 187
 - hiding columns 50-51, 237
 - HIGHLIGHT attribute 193
 - highlighting summary lines
 - See overlining summary lines
 - See underlining
 - horizontal position of text, controlling
 - at beginning or end of report 120-121
 - between groups 120-121
 - hyphen (-)
 - expanding to fill space over column 66, 196, 206
 - representing overlining character 103, 138, 189, 218
 - representing underlining character 105, 140, 191, 218
 - used by HEADLINE option 182
- ## I
- IF statement
 - See conditional statements
 - initializing variables
 - effect on percentage calculations 164-165
 - group summaries 122-124
 - report construction process 246, 248-249, 250-251, 253
 - report summaries 153-155
 - input data set, specifying
 - See DATA = option, REPORT statement
 - interactions of variable position and usage 211-213
 - INTERNAL value, ORDER= option 13, 209
 - invoking REPORT procedure 1, 8, 181
 - ITEMHELP= option, DEFINE statement 214

J

justification of variable values
 left justification 114, 208
 right justification 117, 210

K

KEEP= option, REPORT statement 10, 11

L

LABEL statement 62
 labels
 See column headers
 LAST. variable 192
 LAST automatic variable 182
 LAST= system option 182
 layout of reports
 See report layout
 LEFT function 117-118
 left justification of variable values 114, 208
 LEFT option, DEFINE statement 208
 LENGTH function
 determining length of character variables
 114-115
 determining length of numeric variables
 117-118
 LENGTH= option, COMPUTE statement 75,
 201
 LENGTH statement
 placing conditional text at report beginning
 or end 156-157
 placing conditional text between groups
 125-127
 less than sign (<)
 expanding to fill space over column 66,
 196, 206
 extending column headers with inequality
 signs 68-69
 librefs, specifying
 profiles 185
 report definitions 56, 184, 186
 line breaks in column headers, controlling
 See also blank lines, inserting
 using quotation marks 64-65
 using split character 64, 65, 196, 205
 LINE statement 215-216
 compared with PUT statement 216
 creating customized group summaries
 234-235
 formatting variables (caution) 111
 inserting multiple lines after summary line
 144-145
 inserting multiple lines between groups
 99-100
 limitations 215
 placing conditional text at report beginning
 or end 156-157
 placing conditional text between groups
 125-127
 requirements 215-216
 specifying formats for variables (caution)
 148

specifying text and variable values between
 groups 111-113
 specifying text or variable values at beginning
 or end of report summa 148-150
 syntax 215

linesize, specifying

 See also pagesize, specifying
 LINESIZE= system option 183
 LS= option 38-39, 183

LINESIZE= system option 38, 183

LIST option, REPORT statement

 effect on windowing environment 187
 omitted from stored report definitions 56,
 184

 purpose and use 182-183

 writing report definitions to SAS log 56

listing all observations and variables

 report with character and numeric variables
 8-9

 report with only numeric variables 9-10

listing created by LIST option

 compared with submitted statements
 182-183

 omitted statements 183

 REPORT statement options omitted 183

 SAS system options omitted 183

long variable values

 See wrapping variable values

LS= option, REPORT statement

 overriding LINESIZE= system option 38
 purpose and use 183
 specifying linesize 38

M

macro variables

 using in column headers 196

MAX statistic 24, 25, 211

MEAN statistic 24, 203-204, 211

MIN statistic 24, 25, 211

MISSING option, REPORT statement

 creating group for missing values 226-227

 effect on execution of conditional code
 (caution) 128, 159

 purpose and use 183

missing values

 character variables on right side of break
 variables 88

 computed variables set to missing if
 positioned incorrectly 77

 creating group for missing values 226-227

 restricting processing of groups 128-130

 restricting processing of groups (caution)
 128, 159

 variables initialized to missing in report
 construction process 246, 248,
 250-251, 253

N

N statistic

 calculation during report construction 247

 combining observations counts 30-31

- N statistic (*continued*)
 - counting all observations in one row 29
 - counting observations for each value of across variable 30
 - default for across variables 19, 202
 - default when no analysis variable appears above or below statistics 25
 - definition 24, 211
 - including in COLUMN statement or DEFINE statement 29
 - sharing of columns with other items 213
 - NAMED option, REPORT statement 183
 - printing all values for long row on consecutive lines 225-226
 - purpose and use 183
 - using with WRAP option 183, 187
 - NMISS statistic 24, 211
 - NOBYLINE option 95
 - NOHEADER option, REPORT statement 184, 187
 - NOLABEL option 61, 206
 - nonwindowing environment
 - See also NOWINDOWS option, REPORT statement
 - See also windowing environment
 - determining from profile 185, 187
 - determining with DMS system option 185, 187
 - invoking REPORT procedure 1
 - specifying with NOWINDOWS option 187
 - NOPRINT option, DEFINE statement 208
 - overridden by SHOWALL option 208
 - suppressing display of items 50, 50-51, 237
 - when to use 208
 - NORKEYS option, REPORT statement 187
 - NOTSORTED option, BY statement 192
 - NOWINDOWS option, REPORT statement
 - invoking REPORT procedure 1
 - omitted from stored report definitions 56, 184
 - purpose and use 187
 - when to use 8
 - NOZERO option, DEFINE statement 208
 - numeric variables
 - controlling space required for writing 117-119
 - converting numeric variable value to character string 117
 - default as analysis variable 7
 - listing all observations and all variables 8-10
 - relationship with column width 182
 - report with character variables and numeric variables 8-10
 - report with only numeric variables 9-10
 - consolidating multiple observations 16, 16-18
 - counting observations for each value of across variable 30
 - grouping observations 223-224
 - listing all observations and all variables 8-10
 - ordering 13-15
 - selecting with WHERE statement or WHERE= data set option 42-43
 - OL option, BREAK statement
 - creating default report summary 232
 - OL honored if both OL and DOL specified 103
 - overlining summary lines 103-104, 253
 - purpose and use 189
 - OL option, RBREAK statement
 - OL honored if both OL and DOL specified 138
 - overlining summary lines 138-139
 - purpose and use 218
 - ORDER= option, DEFINE statement 209-210
 - compared with ORDER option 14
 - DATA value 13, 209
 - default order (caution) 209
 - FORMATTED value 13, 209
 - FORMATTED value as default 13, 209
 - FREQ value 13, 209
 - INTERNAL value 13, 209
 - INTERNAL value as default for other SAS procedures 13, 209
 - ORDER, GROUP, or ACROSS option required 14
 - ordering rows 222-223
 - purpose and use 13, 209-210
 - ORDER option, DEFINE statement 208-209
 - compared with ORDER= option 14
 - displaying order variables 208-209
 - ordering rows 222-223
 - purpose and use 6, 13, 208-209
 - required for using ORDER= option 14
 - specifying order variable 15
 - order variables
 - adding computed variables based on order variables 78-79
 - calculation for summary lines 190
 - default sort order 13, 206
 - displaying 208-209
 - group variable not used in same report 88, 92, 188
 - inability to share columns with other items 213
 - multiple order variables in reports 13
 - referencing in compute blocks 77, 135, 199
 - specifying sort order with ORDER= option 13, 209-210
 - specifying with ORDER option 13, 208-209
 - used as break variable 188
 - using with analysis variable 204-205
 - ordering columns 11-12, 221
 - ordering data with BY variables 192
 - ordering rows
 - default order according to group variable 16, 88
 - NOPRINT option 208
- O**
- observations
 - See also groups
 - See also ordering rows
 - combining methods of counting observations 30-31

- ordering of detail rows by group variable
 - 207-208
 - using DEFINE statement options 13-15
 - using DESCENDING option 206
- OUTREPT= option, REPORT statement
 - See also report definitions
 - omitted from stored report definitions 56, 184
 - purpose and use 184
 - storing report definitions 56, 239
- overline character (-)
 - See also double overline character (=)
 - specifying with OL option 189, 218
 - suppressing in break lines 189
- overlining summary lines 103-104, 138-139

P

- page breaks
 - inserting 210
 - inserting between groups 93-94
- PAGE option
 - BREAK statement 93-94, 189
 - DEFINE statement 210
 - RBREAK statement 218
- pagesize, specifying
 - See also linesize, specifying
 - PS= option 40-41, 185-186
- PAGESIZE= system option 40, 185-186
- panels
 - default of one per page 185
 - definition 184
 - making multi-panel reports 229
 - setting space between panels 186
- PANELS= option, REPORT statement
 - making multi-panel reports 229
 - purpose and use 184-185
- parentheses ()
 - spanning multiple columns with customized column header 73-74
 - specifying multiple items in column headers 19, 195-196
 - specifying multiple statistics 25
- PERCENT. format 249
- percentages, calculating 163-177
 - cumulative percentages for whole report 175-177
 - cumulative percentages within groups 172-174
 - effect of report construction 163
 - initializing variables 164-165
 - missing in preliminary summary line during report construction 252
 - over all groups 169-171
 - recalculating during report construction 251, 254, 256
 - using compound names 164
 - within groups 166-168
- PERCENT8.1 format 167, 170
- period (.)
 - expanding to fill space over column 66, 196, 206
 - placing after macro variables in column headers 196

- plus sign (+)
 - expanding to fill space over column 66, 196, 206
- pmenus, customizing 241
- pointer controls
 - controlling horizontal position of text at report beginning or end 151-152
 - controlling horizontal position of text between groups 120-121
 - specifying with LINE statement 216, 235
- PROFILE= option, REPORT statement 185
- PROFILE window 185, 187
- profiles
 - creating 185
 - storing 185
- PROMPT option, REPORT statement 187
- PRT statistic 24, 211
- PS= option, REPORT statement
 - overriding PAGESIZE= system option 40
 - purpose and use 185-186
 - specifying pagesize 40
- PSSPACE= option, REPORT statement 186
- PUT function
 - converting numeric variable value to character string 117-118
- PUT statement
 - compared with LINE statement 216

Q

- QUIT command 1
- quotation marks
 - controlling line breaks in column headers 64, 64-65
 - enclosing labels for column headers 205
 - enclosing macro variables in column headers 196
 - specifying text between groups 111-112
 - specifying text in column headers 62-63, 73

R

- RANGE statistic 24, 211
- RBREAK statement 217-220
 - AFTER argument 132, 217
 - BEFORE argument 132, 217
 - COLOR= option 220
 - creating default report summaries 132, 233-234
 - DOL option 138-139, 218, 234
 - DUL option 140-141, 218, 234
 - information for windowing environment 220
 - not required with COMPUTE statement 200
 - OL option 138-139, 218
 - PAGE option 218
 - purpose 217
 - requirements 217
 - SKIP option 142-143, 218
 - SUMMARIZE option 136-137, 218-219, 234
 - syntax 217
 - UL option 140-141, 219

- report construction by REPORT procedure
 - 243–256
 - adding computed variables 247–248, 254
 - break line construction 243–244, 248
 - calculating frequency 247
 - calculation of statistics 246, 247, 250
 - compound names in different parts of report 244–245
 - compute block processing 247, 248
 - compute block processing (caution) 249, 255
 - consolidating groups 245–248, 250
 - filling in values from left to right 246, 247, 253–254
 - final summary line 247, 252–253
 - initialization of DATA step variables 248–249, 250–251
 - initialization of variables to missing 246, 248, 250–251, 253
 - preliminary summary line 247, 249, 251, 252, 255
 - recalculating percentages 251, 254, 255, 256
 - recalculation of computed variables for summaries (caution) 249, 251, 255
 - report summaries 245–248, 249, 251–253
 - sequence of events 163, 243–245
 - timing at breaks (synchronizing computed variables) 249, 255–256
 - using DATA step variables 248–256
- report definitions
 - designing to run in windowing environment 238–241
 - difference from submitted statements 56, 184
 - REPORT statement options omitted 56, 184
 - SAS system options omitted 56, 184
 - specifying with REPORT= option 59, 186
 - statements omitted 56, 184
 - storing with OUTREPT= option 56–58, 184
 - three-level names 56
 - viewing in SAS log 57
- report enhancements 33–60
 - adding footnotes 36–37
 - adding titles 34–35
 - changing linesize 38–39
 - changing pagesize 40–41
 - controlling space between columns 44–45
 - presenting one variable in two different ways 54–55
 - selecting observations to process 42–43
 - specifying column width 48–49
 - specifying formats 46–47
 - storing report definitions 56–58
 - suppressing display of items 50–51
 - using stored report definitions 59–60
 - wrapping character variable values across rows 52–53
- report layout 5–22
 - consolidating multiple observations into one row 16–18
 - determining layout 5
 - listing all observations and variables 8–10
 - ordering rows 13–15
 - specifying and ordering columns 11–12
 - specifying statistics 7
 - specifying variables with usage options 6
 - statements that determine layout 5–6
 - using variable values as column headers 19–22
 - when to specify usage options 7
- REPORT= option, REPORT statement
 - COLUMN statement excluded by 59, 186, 194
 - omitted from stored report definitions 56, 184
 - purpose and use 186
 - specifying report definitions 59
- REPORT procedure 179–220
 - See also report construction by REPORT procedure
 - See also REPORT procedure examples
 - See also REPORT statement
 - BREAK statement 89–90, 188–191, 231–232
 - BY statement 191–192
 - CALL DEFINE statement 192–193
 - COLUMN statement 5, 11–12, 25, 26, 27, 29, 78, 80, 82, 85, 166–167, 169–170, 172–174, 175–177, 193–197
 - COMPUTE statement 173, 176, 198–201
 - consolidation of groups 16, 88, 207, 245–248, 250
 - DEFINE statement 5, 25, 29, 54, 55, 166–167, 169–170, 176–177, 201–214
 - ENDCOMP statement 214
 - features 1
 - FREQ statement 214–215
 - invoking in nonwindowing environment 1, 8, 181
 - LINE statement 99–100, 111–113, 125–127, 144–145, 148–150, 156–157, 215–216
 - PROC REPORT statement 181–187
 - RBREAK statement 132, 217–220, 233–234
 - syntax 180–181
 - terminating 1
 - WEIGHT statement 220
- REPORT procedure examples 220–241
 - adding variables to report 230
 - column headers, creating from variable values 224–225
 - customized group summaries, creating 234–235
 - customizing column headers 228
 - default group summaries, creating 231–232
 - default report summary, creating 233–234
 - group for missing values, creating 226–228
 - grouping observations 223–224
 - making multi-panel reports 229
 - ordering rows 222–223
 - printing all values for long row on consecutive lines 225–226
 - specifying and ordering columns 221
 - suppressing display of items 237
 - wrapping character variable values across multiple rows 236
 - writing reports for windowing environment 238–241

- REPORT statement 181–187
 CENTER option 181–182, 186, 210
 COLWIDTH= option 182, 207
 COMMAND option 187
 DATA= option 1, 8, 56, 182, 184
 DROP= option 11
 HEADLINE option 71, 182
 HEADSKIP option 70, 182
 HELP= option 187
 information for windowing environment 187
 KEEP= option 10
 LIST option 38, 56, 182–183, 184, 187
 LS= option 183
 MISSING option 128, 159, 183, 226–227
 NAMED option 183, 187, 225–226
 NOHEADER option 184, 187
 NORKEYS option 187
 NOWINDOWS option 1, 8, 56, 184, 187
 OUTREPT= option 56, 184, 239
 PANELS= option 184–185, 229
 PROFILE= option 185
 PROMPT option 187
 PS= option 40, 185–186
 PSSPACE= option 186
 REPORT= option 56, 59, 184, 186, 194
 SHOWALL option 186, 208
 SPACING= option 44, 186
 SPLIT= option 64, 65, 186
 syntax 181
 WINDOWS option 56, 184, 187
 WRAP option 183, 187, 225–226
- report summaries 131–161
 assigning text to character variables in summaries 146–147
 calculating cumulative percentages 175–177
 construction by REPORT procedure 243–244, 245–248, 251–253
 controlling execution of statements in compute block 159–161
 controlling horizontal position of text at beginning or end 151–152
 customized report summaries 133–135, 249, 252
 default report summaries 132–133, 136, 233–234, 249
 initializing variables at beginning 153–155
 inserting blank line after summary at beginning 142–143
 inserting multiple blank lines after summary at beginning 144–145
 overlining summary information 138–139
 placing conditional text at beginning or end 156–158
 specifying text at beginning or end 148–150
 specifying variable values at beginning or end 132, 148–150
 summarizing variables and statistics 136–137
 underlining summary information 140–141
- report variables
 See also variables
 calculating cumulative percentages for whole report 175–177
 calculating percentages over all groups 169–170
 calculating percentages within groups 166–167
 compared with DATA step variables 122, 153
 definition 164
 initialized to missing during report construction 246, 248
 referencing in compute blocks 76–77, 92, 134–135, 199
 right justification of variable values 117, 210
 RIGHT option, DEFINE statement 210
- rows
 See also detail rows
 consolidating multiple observations 16–18
 values filled in from left to right during report construction 246, 247
 wrapping character variable values across multiple rows 52–53, 236
- rows, ordering
 default order according to group variable 16, 88
 NOPRINT option 208
 ordering of detail rows by group variable 207–208
 using DEFINE statement options 13–15
 using DESCENDING option 206
 RSVVIDEO attribute 193
- S**
- sample SAS data sets
 creating BUDGET2 2–3
 TESTMISS 226
- SAS catalogs
 See catalogs, specifying
- SAS data sets
 See also sample SAS data sets
 name stored in `._LAST_` automatic variable 182
 specifying with DATA= option 182
- SAS dates
 ordering rows 14–15
- SAS log
 viewing stored report definitions 57
 writing report definitions with LIST option 56
- SAS system options
 CENTER 111, 120, 148, 151, 181–182, 205
 DMS 185
 FORMCHAR= 182
`._LAST_`= 182
 LINESIZE= 183
 omitted from stored report definitions 56, 184
 PAGESIZE= 40, 185–186
 SASUSER.PROFILE catalog 185
 selecting columns and variables
 See columns
 See variables
 sharing columns
 See columns

- SHOWALL option, REPORT statement
 - overriding of NOPRINT option 208
 - overriding of NOZERO option 208
 - overriding options in DEFINE statement 186
 - purpose and use 186
- SKIP option, BREAK statement
 - creating default group summaries 231-232
 - inserting blank lines between groups 97-98, 255
 - purpose and use 189
- SKIP option, RBREAK statement
 - inserting blank line after summary line at beginning of report 142-143
 - purpose and use 218
- slash (/)
 - default split character 64, 186
- SORT procedure 192
- sorting columns and rows
 - See ordering columns
 - See ordering rows
- space required for writing variables
 - character variables 114-116
 - numeric variables 117-119
- spaces (blanks)
 - See blank characters
- SPACING= option, DEFINE statement
 - effect on CENTER and NOCENTER options 210
 - effect on SPACING= option of REPORT statement 186
 - overriding default column spacing 44
 - purpose and use 210
- SPACING= option, REPORT statement
 - effect of SPACING= option of DEFINE statement 186
 - purpose and use 186
 - specifying default number of blank characters between columns 44
- spanning multiple columns with one column header 73-74, 196
- special characters in column headers
 - See expanding column headers
- split character
 - controlling line breaks in column headers 64, 65
 - honored by FLOW option 186
 - slash (/) as default character 64, 186
 - specifying with SPLIT= option 186
 - using in column headers 196, 205
- SPLIT= option, REPORT statement
 - purpose and use 186
 - specifying split character 64, 65
- stacking multiple items in columns 19-22, 195
- statistics 23-31
 - above variable 25-26
 - adding variables computed from statistic 80-81
 - available statistics 7, 23-24
 - below variable 25-26
 - calculation at breaks 88
 - calculation during report construction 246, 247, 250
 - calculation for summary lines 190, 219
 - combining observations counts 30-31
 - compound name for statistic and variable 216
 - counting observations for each value of across variable 30
 - counting observations in each row 29
 - CSS 24, 211
 - CV 24, 211
 - default column header 61
 - listing multiple statistics in parentheses 25
 - MAX 24, 25, 211
 - MEAN 24, 211
 - MIN 24, 25, 211
 - multiple statistics for one variable 27
 - N 24, 25, 29-31, 202, 211, 213
 - N as default when no analysis variable appears above or below 25
 - NMISS 24, 211
 - notations for summation over all nonmissing values 23-24, 210
 - placing above or below variable 25, 25-26
 - PRT 24, 211
 - RANGE 24, 211
 - reporting multiple statistics for one variable 27, 27-28
 - required with ANALYSIS option 7
 - separating from variables with commas 25
 - sharing of columns with other items 213
 - showing frequency counts 29-31
 - specifying for column headers 194
 - specifying with COLUMN statement 26, 29
 - specifying with DEFINE statement 210-211
 - STD 24, 211
 - STDERR 24, 211
 - SUM 24, 88, 166-167, 195, 203, 211, 224
 - SUMWGT 24, 211
 - T 24, 211
 - USS 24, 211
 - values at beginning or end of reports 132
 - VAR 24, 211
- STD statistic 24, 211
- STDERR statistic 24, 211
- storing report definitions 56-58, 184
 - See also report definitions
- SUM statistic
 - calculating percentages over all groups 169-170
 - calculating percentages within groups 166-167
 - default for analysis variables 16, 88, 203, 210
 - definition 24, 211
 - grouping observations 223-224
 - omitted from report in some cases 195
- SUMMARIZE option, BREAK statement
 - creating default report summary 232
 - generating summary lines 252, 253, 256
 - purpose and use 189-190
 - summarizing variables and statistics for groups 101-102
- SUMMARIZE option, RBREAK statement
 - creating default report summary 136-137, 234
 - purpose and use 218-219
- summary lines
 - See also break lines

See also group summaries
 See also report summaries
 calculation of statistics associated with
 analysis variable 203
 calculation of values by REPORT procedure
 190, 219, 248
 construction by REPORT procedure 244,
 248, 251-253, 255-256
 creating default group summaries 231-232
 creating default report summary 233-234
 definition 188
 generating with SUMMARIZE option 252,
 253, 256
 overlining summary information 103-104
 recalculating during report construction
 (caution) 251
 specifying with SUMMARIZE option
 189-190
 suppressing underlining and overlining 190
 underlining summary information 105-106
 SUMWGT statistic 24, 211
 SUPPRESS option, BREAK statement
 creating default report summary 232
 purpose and use 190
 suppressing value of group variable
 107-108
 suppressing
 break variables 190
 BY lines 95-96
 display of items 50-51, 237
 repeated values of variables 16-18
 underlining and overlining of break lines
 190
 value of group variable 107-108
 variable labels in column headers 206

T

T statistic 24, 211
 templates for reports
 See report definitions
 terminating REPORT procedure 1
 text
 assigning to character variables 109-110
 assigning to character variables in summary
 lines 146-148, 248, 252
 controlling horizontal position of text at
 report beginning or end 151-152
 controlling horizontal position of text
 between groups 120-121
 customizing in column headers 62-63
 placing conditional text at report beginning
 or end 156-158
 placing conditional text between groups
 125-127
 specifying text and variable values between
 groups 111-113
 specifying values at beginning or end of
 report summaries 148-150
 TITLE statement
 adding titles to reports 34-35
 omitted from stored report definitions 56,
 184

totals
 See group summaries
 See report summaries

U

UL option, BREAK statement 190-191
 creating default report summary 232
 purpose and use 190-191
 UL honored if both UL and DUL specified
 105
 underlining summary lines 105-106, 253
 UL option, RBREAK statement 219
 purpose and use 219
 UL honored if both UL and DUL specified
 140
 underlining summary information 140-141
 underline character (-)
 See also double underline character (=)
 expanding to fill space over column 66,
 66-67, 196, 206
 specifying with UL option 190-191
 underlining
 column headers and spaces between
 columns 71
 headers but not spaces between columns 72
 summary lines 105-106, 140-141
 suppressing in break lines 190
 usage options
 See DEFINE statement
 user-defined formats
 See formats
 USS statistic
 definition 24, 211

V

VAR statistic 24, 211
 variables
 See also across variables
 See also analysis variables
 See also automatic variables
 See also break variables
 See also character variables
 See also computed variables
 See also data set variables
 See also group variables
 See also macro variables
 See also numeric variables
 See also order variables
 calculation at breaks 88
 controlling space required for writing
 character variables 114-116
 controlling space required for writing
 numeric variables 117-119
 initialization by report construction process
 246, 248-249, 250-251, 253
 initializing before calculating percentages
 164-165
 initializing for report summaries 153-155
 initializing variables for each group 122-124
 interactions of position and usage 211-213

variables (*continued*)

- labels used as default column headers 61, 206
- listing all observations and all variables 8-10
- placing statistics above or below 25, 25-26
- referencing in compute blocks 76-77, 92, 199
- relationship with column width 182
- selecting with COLUMN statement 11-12, 221
- specifying text and variable values between groups 111-113
- specifying values at beginning or end of report summaries 148-150
- suppressing display of items 50-51, 237
- suppressing variable labels in column headers 61, 206
- using aliases 54-55, 196-197
- using as column headers 19-22, 61, 206
- values at beginning or end of reports 132

\$VARYING. format

- controlling space required for writing numeric variables 117-119
- controlling space used to write character variables 114-116

W

\$w. format 207

WEIGHT statement 220

- omitted from stored report definitions 56, 184
- requirements 220
- syntax 220
- using with **FREQ** statement 214

weight variables 220

WHERE= data set option 42-43

WHERE statement

- omitted from stored report definitions 56, 184
- selecting observations for processing 42-43

width of columns

See column width

WIDTH= option, **DEFINE** statement

See also **COLWIDTH=** option, **REPORT** statement

- first choice for column width 182
- purpose and use 211
- specifying column width 48, 48-49
- specifying width for items that share columns 21

wrapping character variable values across multiple rows 236

windowing environment

See also nonwindowing environment

- BREAK** statement effects 191
- BY** statement effects 192
- determining from profile 185, 187
- determining with **DMS** system option 185, 187
- REPORT** statement effects 187

- specifying with **WINDOWS** option 187
- writing report to run in windowing environment 238-241

WINDOWS option, **REPORT** statement

- omitted from stored report definitions 56, 184
- purpose and use 187

WRAP option, **REPORT** statement

- printing all values for long row on consecutive lines 225-226
- purpose and use 187
- using with **NAMED** option 183, 187

wrapping variable values

- printing all values for long row on consecutive lines 225-226
- wrapping across multiple rows 52-53, 236

Special Characters

, (comma)

See comma (,)

/ (slash)

See slash (/)

= (equal sign)

See equal sign (=)

(quotation marks)

See quotation marks

. (period)

See period (.)

< (less than sign)

See less than sign (<)

() (parentheses)

See parentheses ()

+ (plus sign)

See plus sign (+)

* (asterisk)

See asterisk (*)

- (hyphen)

See hyphen (-)

_ (underline character)

See underline character (_)

> (greater than sign)

See greater than sign (>)

@ pointer control

See at (@) pointer control

' (quotation marks)

See quotation marks

2

2. format 27

Your Turn

If you have comments or suggestions about SAS Technical Report P-258, *Using the REPORT Procedure in a Nonwindowing Environment, Release 6.07*, please send them to us on a photocopy of this page.

Please return the photocopy to the Publications Division (for comments about this book) or the Technical Support Division (for suggestions about the software) at SAS Institute Inc., SAS Campus Drive, Cary, NC 27513.