Part 1

Getting Started

You can use the REPORT procedure to generate a wide range of sophisticated tables and reports. The code can be complex or fairly straightforward. This part of the book shows you how to create fairly simple reports using the basic statements and their options.

Chapter 1

Creating a Simple Report

The syntax for PROC REPORT is quite different from that of most other Base SAS procedures. In most procedures, the supporting statements define the scope and options of the procedure. In a PROC REPORT step, on the other hand, the statements refer to and build on each other.

PROC REPORT can be used in two different modes, batch and interactive. This book discusses the syntax of PROC REPORT in the batch environment, and does not discuss the interactive or windowing environment.

1.1 Basic Syntax

Like most procedures, PROC REPORT can be executed with a minimal understanding of even the most basic syntax.

In its simplest form, PROC REPORT is similar to PROC PRINT in that it creates a data listing. Here is the minimum coding required:

```
PROC REPORT;
  run;
```

By default the REPORT procedure opens an interactive windowing environment. This environment is not normally used and is not discussed in this book. The following is the simplest PROC REPORT step that does not open the interactive windowing environment:

```
PROC REPORT nowd;❶
  run;
```

When executed, this simple step creates a listing of all rows and all columns in the most recently modified data table. This plain vanilla result is of course rarely what we need or want, so we must know more in order to create the report that we actually do need.

Some of the basic statements used in PROC REPORT include the following:

```
PROC REPORT ❷ DATA= datasetname <options>;
 COLUMN variable list and column specifications;
 DEFINE column / column usage and attributes;
 COMPUTE column; compute block statements; ENDCOMP;
 RUN;
```

A number of options and modifiers can be used along with these statements. Most of these are discussed throughout this book. To locate the discussion of a specific statement, option, or modifier, see Appendix 2, which provides a syntax and example reference locator for this book.

You can use the REPORT procedure to build reports interactively (LeBouton 2004). While appealing in concept, in practice this feature is rarely used and is not discussed in this book. Unfortunately, the procedure default is to initiate the interactive mode. You can disable this made by using either the NOFS, NOWD, or NOWINDOWS option. NOWD ❶ is most often used in the documentation and in SAS literature.

As for most procedures that operate against data tables, you will want to be able to specify which table PROC REPORT is to display. ❷The DATA= option is used for this specification in the REPORT procedure as it is in so many other SAS procedures.

A number of supporting statements are used in the PROC REPORT step. The following statements are three of the most common:

COLUMN	identifies all variables (report items) used in the generation of the table.
DEFINE	specifies how the column is to be used and what its attributes are to be. One DEFINE statement is used for each variable in the COLUMN statement.
COMPUTE	creates new columns and performs column-specific operations.

The following PROC REPORT step creates a simple listing of a select few of the twenty or so variables in the RPTDATA.CLINICS table:

```
* Simple report;
options nocenter;
title1 'Using Proc REPORT';
title2 'Simple Report';
proc report data=rptdata.clinics nowd;
columns region lname fname wt;
define region / display;
define lname / display;
define fname / display;
define wt / display;
run;
```

Here are the first few lines of the generated report.

```
Using Proc REPORT
Simple Report

   re
   gi              first      weight
   on  last name   name    in pounds
   5   Rose        Mary          215
   6   Nolan       Terrie        187
   9   Tanner      Heidi         177
   2   Saunders    Liz           109
   4   Jackson     Ted           201
   5   Pope        Robert        158
   8   Olsen       June          158
   4   Maxim       Kurt          179

. . . Portions of the report are not shown . . .
```

A quick inspection of the output listing shows both similarities and differences between PROC REPORT and PROC PRINT. As in PROC PRINT, variables/columns are listed across the page, while rows/observations are listed down the page. Unlike PROC PRINT, there is no OBS column, and the default is to print the variable label instead of the variable name. “Pretty” is not a default characteristic, and the remainder of this book is devoted to controlling how the report looks.

Notice in this example that the default header of the column is the variable label. In Section 2.5, several examples show how you can control this text. You can also use the system option NOLABEL to make the variable name the default column header.

MORE INFORMATION

Appendix 2, “Syntax and Example Index,” is designed to help the reader navigate this book.

SEE ALSO

A nice introduction to the PROC REPORT windowing environment is presented by LeBouton (2004). This interactive environment was first introduced in the *SAS Guide to the REPORT Procedure: Usage and Reference, Version 6* (1990).

1.2 Routing Reports to ODS Destinations

Usually we need to route the output generated by PROC REPORT to one or more Output Delivery System (ODS) destinations. The syntax and use of ODS is outside the direct scope of this book. However, because we are going to depend on ODS for a great deal of the appearance of the output generated by PROC REPORT, it is necessary to at least discuss the basics of ODS.

Since we use reports in different ways, we need to generate the reports as different types of files. We declare the type of file to be generated by specifying the ODS *destination*. This means that there is generally a correspondence between the name of the destination and the type of output that is to be created (*e.g.,* HTML, PDF, RTF).

Usually we surround the PROC REPORT step with what has been referred to as an ODS sandwich. The sandwich consists of two ODS statements that turn the Output Delivery System on and off. The physical name of the output file and the file's location are included in the first ODS statement. The second ODS statement closes (turns off) the ODS destination. In both statements, the ODS destination name immediately follows the ODS keyword.

The general form of the ODS sandwich is something like this:

```
ods destination <file=file name>;

proc report . . . . ;
. . . .
run;

ods destination close;
```

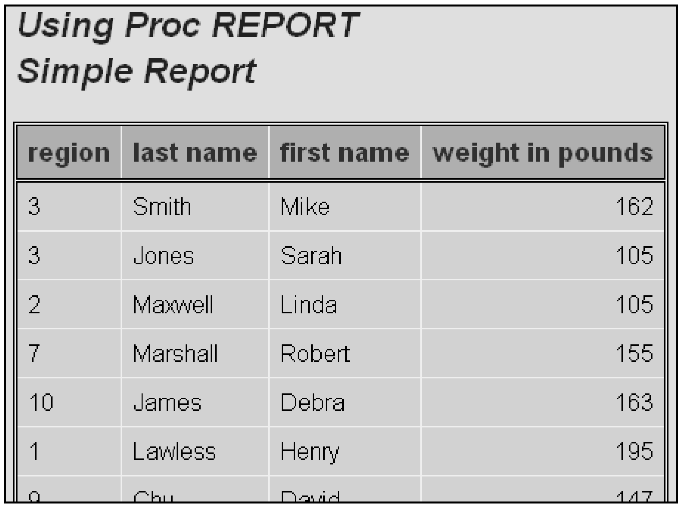
If you wanted to re-create the results of the previous step as an HTML document, you might write ODS statements like the following. Note that all physical paths in the examples are created using the macro variable &PATH. This should make it easier for you to replicate the results of these same example programs on your own computer.

```
ods html file="&path\results\simple.html";

title1 'Using Proc REPORT';
title2 'Simple Report';
proc report data=rptdata.clinics nowd;
   columns region lname fname wt;
   define region / display;
   define lname / display;
   define fname / display;
   define wt / display;
   run;

ods html close;
```

Here is a portion of the HTML report:



Using Proc REPORT
Simple Report

region	last name	first name	weight in pounds
3	Smith	Mike	162
3	Jones	Sarah	105
2	Maxwell	Linda	105
7	Marshall	Robert	155
10	James	Debra	163
1	Lawless	Henry	195
9	Chu	David	147

```
 . . .  Portions of the report are not shown . . .
```

Throughout this book you will see examples of a number of other ODS statements, options, and destinations.

MORE INFORMATION
The &PATH macro variable is used throughout the book to designate the upper portion of all location references and is described in more detail in “About This Book.” Appendix 2 contains a list of ODS-related references within this book. A number of other sections in this book contain examples that utilize features of ODS. Chapter 8, “Using PROC REPORT with ODS,” and Chapter 9, “Reporting Specifics for ODS Destinations,” are devoted to the topic.

SEE ALSO
Haworth (2001, 2003) and Gupta (2003) provide very good information on the Output Delivery System and show how to get started using it. Kumar (2006) introduces ODS along with a PROC REPORT example.

1.3 Other Reporting Tools: A Brief Comparison of Capabilities

Since SAS provides a variety of reporting tools, there is sometimes some confusion about which tool should be used in a given situation. Three of the primary reporting tools are the PRINT, REPORT, and TABULATE procedures. All three have enough flexibility to produce a fairly diverse set of reports. However, they are not the same and do not have the same overall capabilities.

All three of these procedures work well with the ODS environment, and each supports the use of the STYLE= option (see Section 8.1 for an introduction to this option).

1.3.1 PROC REPORT vs. PROC PRINT

Both the PRINT and REPORT procedures can perform detail-level reporting (reporting of individual data values). Although a number of supporting statements are available, PROC PRINT has the advantage of being a fairly simple procedure and is generally one of the first procedures that is learned by a new user.

Although both procedures are good at creating simple detail reports, the only real summary capability of PROC PRINT is to calculate column totals. When the SUM statement is combined with the BY statement, SUM (and SUMBY) can calculate group and sub-group totals. Unlike PROC PRINT, PROC REPORT is not limited to group totals. PROC REPORT can calculate all of the usual statistics that can be calculated by other procedures such as MEANS, SUMMARY, and UNIVARIATE. In fact, the reason that PROC REPORT can calculate some of these same statistics is that the MEANS/SUMMARY process is used behind the scenes for summarizing the data set used with PROC REPORT.

Most users find that PROC PRINT is fine for simple straightforward detailed reports. However, if you find that the limitations of PROC PRINT are causing extra work, then it is probably an indication that it is time to switch to PROC REPORT.

SEE ALSO
Burlew (2005, pp. 18–19) provides a comparison of the default behaviors of these two procedures.

1.3.2 PROC REPORT vs. PROC TABULATE

Both the REPORT and TABULATE procedures can create summary reports, and each has access to the same standard suite of summary statistics.

Unlike PROC TABULATE, the REPORT procedure can provide detail reporting as well as summary reporting capabilities. PROC REPORT has the added flexibility to calculate and display columns of information based on other columns in the report.

Because of the unique way that PROC TABULATE structures the report table, it has a great deal more flexibility to present the groups, subgroups, and statistics as either rows or columns. This is especially true for vertically concatenated reports, which are very straightforward in PROC TABULATE and difficult in PROC REPORT (see Section 10.1).

SEE ALSO
Buck (1999, 2004), Bruns, Pass, and Eaton (2002), and Bruns and Pass (2004) compare the strengths and weaknesses of these two procedures.

The TABULATE procedure is fully described by Haworth (1999).

1.3.3 PROC REPORT vs. DATA _NULL_

The DATA _NULL_ step is a reporting tool that offers extreme flexibility. Because it uses the power of the DATA step, this methodology enables the user to generate reports in almost any form.

Of course, this power comes with the price of complexity. Although the user has the power to place every character "just so," the process itself can become quite difficult. In PROC REPORT, the compute block, with its access to the majority of the SAS language elements, such as logic processing, functions, and assignment statements, takes on some of the role of the DATA _NULL_ step.

1.4 The PROC REPORT Process: An Overview

For most procedures, the internal processing is of little interest to the average user. This should not be the case for PROC REPORT. Because PROC REPORT has the capability of creating columns as well as group and report summaries, the process can be quite complex. When the report is simple, such as those in this chapter and in Chapter 2, "PROC REPORT: An Introduction," the processing details are of less interest. However, as new columns are calculated and perhaps then coordinated with report and group summaries, a more complete understanding of the process becomes critically important.

MORE INFORMATION
The timing of the compute block is discussed in Section 7.1, and a detailed presentation of the processing of the PROC REPORT step is provided in Chapter 11, "Details of the PROC REPORT Process."

SEE ALSO
SAS Technical Report P-258 (1993, Chapter 10), Lavery (2003), and Russ Lavery's "An Animated Guide to the SAS REPORT Procedure," which is included on the CD that accompanies this book, discuss the sequencing of events in detail.

1.4.1 PROC REPORT Terminology

Some of the terms and concepts associated with PROC REPORT are similar to those in other types of PROC steps. However, PROC REPORT is unique in that it allows some DATA step-type processing to be performed, and thus we need some specialized words and phrases to discuss this processing. Of course, the special nature of PROC REPORT results in terminology that is unique to this step, and an overview of basic PROC REPORT processing will highlight these terms.

Term usage has evolved since the introduction of the REPORT procedure in SAS 6.06. This not only reflects the complexity of the procedure, but also the changes in how PROC REPORT operates behind the scenes.

Some of the terminology used in this book is included here.

Current Terminology

Two general types of reports can be generated by PROC REPORT. **Detail reports** are most similar to those generated by PROC PRINT and have one line in the report, called a **report row,** for each observation in the incoming data set. When the incoming data is summarized or collapsed into groups, PROC REPORT can create a **summary report**. PROC REPORT is flexible enough to create a report that has characteristics of both of these types of reports.

The report generated by PROC REPORT is called the **final report output**. Columns on the final report output can include more than variables, so the report columns are often referred to as **report items**. There are two general classes of report items used within the PROC REPORT step:

report variables	appear in the COLUMN statement and usually in one or more of the report columns. They may or may not be created or used in COMPUTE blocks.
temporary variables	are created and used in COMPUTE block calculations, but do not appear in the COLUMN statement or on the report itself.

Through the DEFINE statement, report items are assigned a **define type** or **define usage** that determines how the variables are to be processed by PROC REPORT. Report items can be used during compute block execution to build or calculate other report items. Depending on the PROC REPORT step, not all report items will necessarily be included in the final report output.

PROC REPORT builds each report row, one row at a time. However, in order for summary and break information to be available when it is constructing summary rows, PROC REPORT goes through a three-phase process to create the report. The first phase is the **evaluation phase**, and it is during this phase that the submitted code is assessed. The final two phases are of special interest to the PROC REPORT programmer, and these (the **setup phase** and the **report row phase**) are described in more detail in the section on processing phases (see Section 1.4.2).

For reports that summarize the incoming data, the summary results are determined during the setup phase and stored in memory in the **computed summary information**.

Each line of a report is a report row; for some reports, however, report rows are generated that are not ultimately written to the final report output. The **final report output** is generated one row at a time, and depending on the selection of statement options, not all summary report rows are included in the final report output.

Outdated Terminology

Since its introduction in SAS 6, PROC REPORT has been the subject of a great many papers. This unofficial documentation, as well as some of the initial official documentation, has generated fairly extensive terminology for the internal processes of PROC REPORT. Although some of this terminology reflects, to some degree, current internal processes, the majority has at best become outdated. In order to assist readers of this older literature, the following table attempts to link the older terminology with that used throughout this book.

Older, Outdated, or Inaccurate Terms	Terminology Used in This Book
Temporary Internal File, TIF	Computed summarized information computed summary information area
Report Data Vector, RDV	Report variables, report items
DATA Step Variable	Temporary variable
DATA Variable Table, DVT	Temporary variables are stored in memory and no special name is needed for this location.
DATA Step Statement DATA Step Functions	SAS language elements

SEE ALSO

Extensive discussion contrasting **report items** and **temporary variables** can be found in Chapter 10 and more specifically on pages 250-251 of SAS Technical Report P-258 (1993). When you read SAS Technical Report P-258, remember that it reflects some of the earliest documentation available for PROC REPORT and does not use the current terminologies or in some cases reflect the current processes of the PROC REPORT step.

1.4.2 Processing Phases

When a PROC REPORT step is submitted, SAS breaks down the processing into a series of steps or phases. All of this processing, as well as the results of the processing, including the computed summary information, takes place in memory.

Evaluation Phase

First, all the PROC REPORT statements are evaluated before anything else happens. The SAS language elements and LINE statements (if there are any) in the compute blocks are simply set aside to be executed as each report row is built (report row phase). This evaluation determines the resources and levels of summarization that will be needed during the setup phase.

Setup Phase

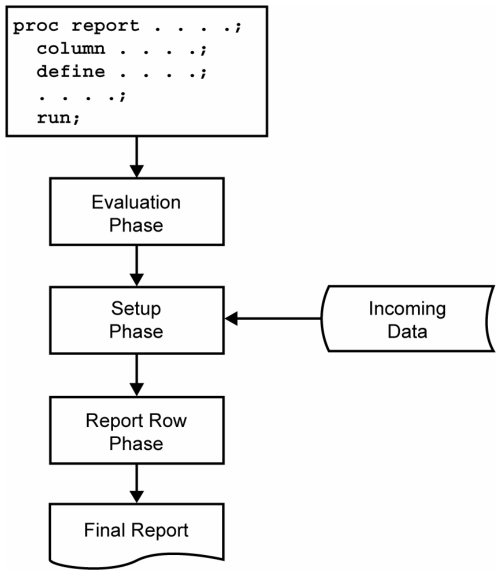
After the code is evaluated, the setup phase reads and prepares the incoming data. If necessary, the columns that will be used for summarizing are sent to the MEANS/SUMMARY engine, where the summarization takes place.

Report Row Phase

Once PROC REPORT is done with these preliminary setup phase tasks and the computed summarized information has been created, the report can be built row by row during the report row phase. Finally, after each report row is built, it is sent to all open ODS destinations (LISTING, HTML, RTF, PDF, etc.).

Summary of the Processing Phases

The following flowchart shows the general processing phases at the conceptual level described in this section.



It is tempting to try to impose DATA step concepts such as the Program Data Vector onto PROC REPORT. However, between PROC REPORT's original inception with SAS 6 and subsequent upgrades and rewrites to the underlying PROC REPORT code, these primary conceptual phases (especially the setup and report row phases) should suffice to explain how the final report is constructed.

Of primary importance is to remember that during the report row phase, processing is from left to right. The order of the variables in the COLUMN statement (see Section 2.1), therefore, becomes very important.

1.5 Chapter Exercises

1. What are the three processing phases of a PROC REPORT step?
2. What is the difference between temporary variables and report variables?
3. What two PROC REPORT statement options will you use within virtually every PROC REPORT statement?