

# The SAS<sup>®</sup> Programmer's PROC REPORT Handbook

*ODS Companion*



Jane Eslinger

The correct bibliographic citation for this manual is as follows: Eslinger, Jane. 2018. *The SAS® Programmer's PROC REPORT Handbook: ODS Companion*. Cary, NC: SAS Institute Inc.

**The SAS® Programmer's PROC REPORT Handbook: ODS Companion**

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-63526-281-0 (Hard copy)

ISBN 978-1-63526-353-4 (EPUB)

ISBN 978-1-63526-354-1 (MOBI)

ISBN 978-1-63526-355-8 (PDF)

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

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

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

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

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

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

March 2018

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

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

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.



The SAS® Programmer's PROC REPORT Handbook: ODS Companion. Full book available for purchase [here](#).

# Contents

<b>About this Book .....</b>	<b>vii</b>
<b>About the Author .....</b>	<b>xi</b>
<b>Preface .....</b>	<b>xiii</b>
<b>Chapter 1 Introduction: Overview of Chapters .....</b>	<b>1</b>
<b>Chapter 2 PROC TEMPLATE Overview .....</b>	<b>5</b>
2.1 Introduction .....	5
2.2 Default Style Templates .....	6
2.3 Location of Style Templates .....	7
2.4 Viewing Style Templates .....	9
2.5 General Syntax of PROC TEMPLATE .....	12
2.6 STYLE versus CLASS Statements .....	14
2.7 Inheritance .....	14
2.8 Style Attributes .....	15
2.9 Key Takeaways .....	16
<b>Chapter 3 Style Elements .....</b>	<b>17</b>
3.1 Introduction .....	17
3.2 Style Elements Used by PROC REPORT .....	18
3.3 Setup for Chapter Examples .....	18
3.4 Plain Output Generated without Attribute Changes .....	20
3.5 Header Style Element Changes .....	24
3.6 LineContent Style Element Changes .....	28
3.7 DataEmphasis Style Element Changes .....	32
3.8 Data Style Element Changes .....	36
3.9 Table Style Element Changes .....	40

3.10 Other Style Elements .....	46
3.11 Key Takeaways .....	55
<b>Chapter 4 Style Overrides within PROC REPORT .....</b>	<b>57</b>
4.1 Introduction .....	57
4.2 STYLE= Default Locations and Order of Precedence .....	58
4.3 STYLE= in PROC REPORT Statement .....	59
4.4 STYLE= in DEFINE Statement.....	61
4.4.1 Using CELLWIDTH= Style Attribute.....	61
4.4.2 Applying Trafficlighting.....	63
4.4.3 Using Multiple Locations in the Same DEFINE Statement.....	64
4.5 STYLE= in BREAK and RBREAK Statements.....	65
4.6 STYLE= in COMPUTE Statement .....	66
4.7 CALL DEFINE STATEMENT.....	67
4.8 Key Takeaways .....	69
<b>Chapter 5 PDF Destination .....</b>	<b>71</b>
5.1 Introduction .....	71
5.2 Style Templates with ODS PDF .....	72
5.3 Border Manipulation .....	73
5.3.1 Manipulate Borders Already in the Table.....	74
5.3.2 Insert New Borders .....	82
5.3.3 Border Width Values.....	84
5.4 Paging .....	84
5.5 PROC REPORT SPANROWS Option .....	85
5.6 Accessibility.....	93
5.7 Key Takeaways .....	93
<b>Chapter 6 RTF and Tagsets.RTF Destinations .....</b>	<b>95</b>
6.1 Introduction .....	95
6.2 Differences between the RTF and Tagsets.RTF Destinations .....	96
6.2.1 Titles and Footnotes.....	96
6.2.2 CONTINUED Text .....	97
6.2.3 Watermark Text .....	99
6.3 Effect of Style Element Changes in Tagsets.RTF.....	101
6.3.1 Header Style Element Changes .....	102
6.3.2 LineContent Style Element Changes.....	103
6.3.3 DataEmphasis Style Element Changes.....	103
6.3.4 Data Element Style Changes.....	104

6.3.5 Table Element Style Changes .....	104
6.3.6 Container Element Style Changes .....	105
6.4 Table Bottom Border.....	106
6.5 Tagsets.RTF Options That Affect PROC REPORT .....	110
6.5.1 TABLEROWS .....	110
6.5.2 VSPACE .....	111
6.6 PROC REPORT SPANROWS Option.....	114
6.7 Raw RTF Commands .....	123
6.8 Key Takeaways .....	123
<b>Chapter 7 Excel Destinations .....</b>	<b>125</b>
7.1 Introduction.....	126
7.2 ODS Destination for Excel .....	126
7.2.1 Column Widths .....	127
7.2.2 Row Heights .....	131
7.2.3 Hidden Rows .....	135
7.2.4 FROZEN_ROWHEADERS Suboption .....	136
7.2.5 Border Manipulation .....	137
7.2.6 Customized Text Summary Row .....	141
7.2.7 Leading Zeros for Character Variables .....	143
7.2.8 Style Attributes for Images .....	145
7.2.9 PROC REPORT-Triggered Worksheets .....	146
7.3 Tagsets.ExcelXP .....	148
7.3.1 Default Style .....	149
7.3.2 Hidden Rows .....	150
7.3.3 Border Manipulation .....	151
7.3.4 Leading Zeros for Character Variables .....	154
7.4 HTML, MSOffice2K, and MSOffice2K_X.....	155
7.4.1 Default Style .....	156
7.4.2 Hidden Rows .....	156
7.4.3 Border Manipulation .....	156
7.4.4 Customized Text for Summary Row .....	156
7.4.5 Excel-specific Formats.....	157
7.4.6 Style Attributes for Images .....	158
7.5 Background Color in the Headers .....	159
7.6 Date Values .....	161
7.7 Key Takeaways .....	163

<b>Chapter 8 HTML and HTML5 Destinations.....</b>	<b>165</b>
8.1 Introduction .....	165
8.2 Style Differences between the HTML and HTML5 Destinations .....	166
8.2.1 Header Style Element Changes .....	167
8.2.2 LineContent Style Element Changes.....	168
8.2.3 DataEmphasis Style Element Changes.....	169
8.2.4 Data Element Style Changes.....	170
8.2.5 Table Element Style Changes .....	170
8.2.6 Container Element Style Changes .....	171
8.3 Borders.....	172
8.4 Hyperlinks .....	174
8.5 Images.....	175
8.6 Accessibility.....	176
8.7 Key Takeaways .....	176
<b>Chapter 9 ODS Destination for PowerPoint .....</b>	<b>177</b>
9.1 Introduction .....	177
9.2 PowerPointLight and PowerPointDark Style Templates .....	178
9.3 Style Template Changes .....	180
9.4 Effects of Tables with Large Amounts of Data .....	187
9.4.1 Many Columns of Data.....	187
9.4.2 Many Columns and Rows of Data .....	192
9.4.3 Many Header Rows .....	199
9.5 Inline Formating Functions .....	203
9.6 Style Attributes for Images .....	206
9.7 Key Takeaways .....	211
<b>ODS References .....</b>	<b>213</b>
<b>Index .....</b>	<b>215</b>

# About This Book

## What Does This Book Cover?

The purpose of this book is to describe how PROC REPORT interacts with the major Output Delivery System (ODS) destinations. Each ODS destination behaves differently from the others, and programmers need to be aware that they might have to alter their PROC REPORT code or create a custom style template based on the final destination of their report.

This book covers how PROC REPORT interacts with the following destinations:

- PDF
- RTF
- Tagsets.RTF
- HTML
- HTML5
- ODS destination for Excel
- Tagsets.ExcelXP
- ODS destination for PowerPoint

This book does not discuss how to perform calculations with PROC REPORT or how to structure output. Those topics were covered in *The SAS Programmer's PROC REPORT Handbook: Basic to Advanced Reporting Techniques* (referred to as *Reporting Techniques* throughout the book).

This book does not cover style templates and PROC TEMPLATE in-depth. Those topics are briefly discussed with regard to how they relate to PROC REPORT.

## Is This Book for You?

This book is meant for SAS programmers of all skill levels in all industries who need to create reports. This book will teach you how PROC REPORT works with ODS and the steps that you can take with your code to ensure that you are getting the output in the format that you need and with all the styling enhancements that you want.

## What Are the Prerequisites for This Book?

Basic knowledge of SAS programming is necessary in order to understand the concepts and examples in this book. You need a general understanding of the basic concepts of sending output to the ODS destinations.

## Scope of This Book

This book is entirely about PROC REPORT and ODS. PROC REPORT is designed specifically for generating reports. Therefore, ODS plays a major role in the final appearance of the report. The book attempts to give an overview about ODS and each destination as it relates to PROC REPORT. A full description of every destination and every style attribute and its uses is beyond the scope of this book.

Although the TEMPLATE procedure is used in this book, it is not covered in depth. The syntax of the TEMPLATE procedure is described only enough to convey the reason the procedure is needed for use with PROC REPORT.

## What Should You Know about the Examples?

### Software Used to Develop the Book's Content

The examples in this book were developed using Base SAS 9.4TS1M4. The examples can be run in SAS 9.4TS1M0 – SAS 9.4TS1M3, with all available hot fixes installed. The examples that pertain to the ODS destinations for Excel and PowerPoint should be run in the most recent releases, SAS 9.4TS1M4 or SAS 9.4TS1M5.

All examples that are not related to the ODS destinations for Excel and PowerPoint should work in SAS 9.3 as well. But please note that if the example programs are run interactively in SAS 9.3, the NOWD option should be added to the PROC REPORT statement. This book is also compatible with SAS University Edition.

### Example Code and Data

Permission to use the Orion Star data, which was used for demonstration purposes throughout this book, was granted by Sean O'Brien and Eric Rossland. All the examples in this book use one SAS data set ORDERS.

ORDERS contains purchase order data. The data set is unique at CUSTOMER\_ID, ORDER\_ID, and PRODUCT\_ID level. A customer can have multiple orders and within each unique order, the customer can purchase multiple products. The QUANTITY variable contains the quantity ordered of a specific product.

To use the example programs from the author page, you need the ORDERS data set in the WORK location. You also need to submit the createfmts.sas program. The program creates two formats, called \$entry and typef, which are used throughout the book. The formats should be stored in a catalog in the WORK location, as well.

You do not have to read this book in its entirety to gain valuable knowledge about PROC REPORT and how it interacts with ODS. Chapters 2, 3, and 4 provide the foundation for the remaining chapters of the book. You might find it helpful to read these chapters first, and then concentrate on the chapter about the destination that is of most interest to you.

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

If you are unable to access the code through the website, send email to [saspress@sas.com](mailto:saspress@sas.com).



## SAS University Edition



This book is compatible with SAS University Edition. If you are using SAS University Edition, then begin here: [http://support.sas.com/publishing/import\\_ue.data.html](http://support.sas.com/publishing/import_ue.data.html)

## Output and Graphics

The output in this book is sent to many of the ODS destinations, including PDF, HTML, RTF, Tagsets.ExcelXP, and the Excel destination.

## Terminology Used in This Book

The words group, order, and across are capitalized when the context refers to the usage value in the DEFINE statement. All statement names, options, and variable names are capitalized in text.

## We Want to Hear from You

SAS Press books are written *by* SAS Users *for* SAS Users. We welcome your participation in their development and your feedback on SAS Press books that you are using. Please visit [sas.com/books](http://sas.com/books) to do the following:

- Sign up to review a book.
- Recommend a topic.
- Request information about how to become a SAS Press author.
- Provide feedback on a book.

Do you have questions about a SAS Press book that you are reading? Contact the author through [saspress@sas.com](mailto:saspress@sas.com) or [https://support.sas.com/author\\_feedback](https://support.sas.com/author_feedback).

SAS has many resources to help you find answers and expand your knowledge. If you need additional help, see our list of resources: [sas.com/books](http://sas.com/books).



# About the Author



**Jane Eslinger** is a Senior Technical Support Analyst at SAS world headquarters in Cary, North Carolina. Jane is a SAS Certified Advanced Programmer for SAS(R)9 and a SAS Certified Advanced Visual Business Analyst. She frequently presents PROC REPORT and ODS topics at conferences and SAS users groups. She is the author of numerous papers and *The SAS® Programmer's PROC REPORT Handbook: Basic to Advanced Reporting Techniques*.

In her day-to-day work, Jane enjoys supporting SAS customers using ODS and Base SAS procedures, with an emphasis on PROC REPORT. Prior to joining SAS, Jane served as a statistician and statistical programmer in the social science and clinical research fields. She has a B.S. in Statistics from North Carolina State University.

Learn more about this author by visiting her author page at <http://support.sas.com/Eslinger>. There you can download free book excerpts, access example code and data, read the latest reviews, get updates, and more.





The SAS® Programmer's PROC REPORT Handbook: ODS Companion. Full book available for purchase [here](#).

## Chapter 9

# ODS Destination for PowerPoint

9.1 Introduction .....	177
9.2 PowerPointLight and PowerPointDark Style Templates .....	178
9.3 Style Template Changes .....	180
9.4 Effects of Tables with Large Amounts of Data .....	187
9.4.1 Many Columns of Data .....	187
9.4.2 Many Columns and Rows of Data .....	192
9.4.3 Many Header Rows .....	199
9.5 Inline Formatting Functions .....	203
9.6 Style Attributes for Images .....	206
9.7 Key Takeaways .....	211

## 9.1 Introduction

In Chapter 3, the style templates that were created did not have a parent template explicitly set in their definition. The results provided a way to determine which style elements affected which part of the table, without complicated inheritance rules that could cause confusion. SAS ships two style templates, Styles.PowerPointLight and Styles.PowerPointDark, one of which has to be used as a parent with the ODS destination for PowerPoint: Styles.PowerPointLight and Styles.PowerPointDark. PowerPointLight is the default style template for the destination. Results from a style template that does not use one of the shipped templates as the parent are unpredictable and are not supported by SAS.

When generating PowerPoint presentations and making style changes to PROC REPORT tables, it is important to remember that they are designed for display on a large screen. To enhance readability, the table will have a larger font and more space between values than any other destination.

This chapter focuses on the PowerPoint destination and how it might force you to alter your PROC REPORT code. The first section of the chapter looks at the style templates used by the destination. Examples demonstrate the effects of making custom changes to those templates. The chapter also covers how to alter the PROC

REPORT code to make sure that the tables can fit on a PowerPoint slide in a coherent, readable form. Finally, the last sections explore inline formatting options and images.

## 9.2 PowerPointLight and PowerPointDark Style Templates

Before making alterations to the style template used for PowerPoint presentations, it is important to see what the table looks like by default. Example 9.1 creates the same output table as the examples in Chapter 3. However, the results are created using the PowerPointLight style template. The ODS POWERPOINT statement does not specify the STYLE= option because PowerPointLight is the default style template that is used by the destination. Output 9.1 shows the appearance of the default template.

Note that the examples in Sections 9.2 and 9.3 are modified slightly from what was used in Chapter 3. The RBREAK statement has been removed, and well as the compute after block with a LINE statement.

### Example 9.1 Using the PowerPointLight Style Template

```
title;
ods powerpoint file='Output9-1.pptx';
proc report data=orders(obs=20)
    style(report)=[pretext='this is pretext'];

    column ("Customer Info" customer_country customer_age_group)
        ("Order Info" order_type quantity);

    define customer_country / group format=$cntry.;
    define customer_age_group / group;
    define order_type / group format=typef.;

    compute before customer_country;
        line 'a line statement for ' customer_country $cntry.;
    endcomp;

    break after customer_country / summarize;
run;
ods powerpoint close;
```

**Output 9.1 PowerPointLight Output**

this is pretext			
Customer Info		Order Info	
Customer Country	Customer Age Group	Order Type	Quantity Ordered
a line statement for Australia			
Australia	31-45 years	Internet Sale	1
		Retail Sale	4
	46-60 years	Internet Sale	3
<i>Australia</i>			8
a line statement for United States			
United States	15-30 years	Internet Sale	2
		Retail Sale	11
	31-45 years	Retail Sale	7
	61-75 years	Retail Sale	1
<i>United States</i>			21

The PowerPointDark style template was developed to provide a high contrast between foreground and background colors. The slide color is black, and the text is white. Example 9.2 specifies the PowerPointDark style template in the STYLE= option in the ODS POWERPOINT statement. Output 9.2 shows the results.

**Example 9.2 Using the PowerPointDark Style Template**

```

Title;
ods powerpoint file='Output9-2.pptx' style=styles.PowerPointDark;
proc report data=orders(obs=20)
    style(report)=[pretext='this is pretext'];

    column ("Customer Info" customer_country customer_age_group)
        ("Order Info" order_type quantity);

    define customer_country / group format=$cntry.;
    define customer_age_group / group;
    define order_type / group format=typef.;

    compute before customer_country;
        line 'a line statement for ' customer_country $cntry.;
    endcomp;

    break after customer_country / summarize;
run;
ods powerpoint close;

```

Output 9.2 PowerPointDark Output

this is pretext			
Customer Info		Order Info	
Customer Country	Customer Age Group	Order Type	Quantity Ordered
a line statement for Australia			
Australia	31-45 years	Internet Sale	1
		Retail Sale	4
	46-60 years	Internet Sale	3
Australia			8
a line statement for United States			
United States	15-30 years	Internet Sale	2
		Retail Sale	11
	31-45 years	Retail Sale	7
	61-75 years	Retail Sale	1
United States			21

Note that the ODS destination for PowerPoint cannot toggle between style templates within one presentation.

### 9.3 Style Template Changes

Now that you know what the style templates that are supplied by SAS look like, the next step is to consider how changing the template affects the tables produced by PROC REPORT.

Recall from Chapter 3 that the border attribute changes in the Header style element were not applied (Output 3.2.5). This was a consequence of not using a parent style template. The teal borders are applied, along with the gray background and navy foreground, when the custom template inherits from PowerPointLight. Example 9.3 demonstrates how to inherit from PowerPointLight and change the Header element. Output 9.3 shows the resulting PowerPoint slide.



**Example 9.3 Header Style Element Changes**

```

proc template;
define style _ppheader;
parent=styles.powerpointlight;
class header /
    fontfamily='Albany AMT'
    color=#264666
    fontweight=bold
    fontstyle=italic
    backgroundcolor=#F0F0F0
    borderwidth=2pt
    bordercolor=#00929F
    borderstyle=solid;
end;
run;

title;
ods powerpoint file='Output9-3.pptx' style=_ppheader;
proc report data=orders(obs=20)
    style(report)=[pretext='this is pretext']
    style(column)=[width=2in];

    column ("Customer Info" customer_country customer_age_group)
        ("Order Info" order_type quantity);

    define customer_country / group format=$cntry.;
    define customer_age_group / group;
    define order_type / group format=typef.;

    compute before customer_country;
        line 'a line statement for ' customer_country $cntry.;
    endcomp;

    break after customer_country / summarize;
run;
ods powerpoint close;

```

**Output 9.3 Header Element Changes**

this is pretext

<i>Customer Info</i>		<i>Order Info</i>	
<i>Customer Country</i>	<i>Customer Age Group</i>	<i>Order Type</i>	<i>Quantity Ordered</i>
a line statement for Australia			
Australia	31-45 years	Internet Sale	1
		Retail Sale	4
	46-60 years	Internet Sale	3
<i>Australia</i>			8
a line statement for United States			
United States	15-30 years	Internet Sale	2
		Retail Sale	11
	31-45 years	Retail Sale	7
	61-75 years	Retail Sale	1
<i>United States</i>			21

The header in Output 9.3 has a gray background much like the background color in Output 9.1. Even though the shades of gray are similar, they are actually different. The lighter color in Output 9.3 is represented as #F0F0F0, which has been used throughout this book to show the results of modifying the style template.

Output 9.3 also demonstrates that the borders within the table can be controlled. For many PROC REPORT tables, controlling the borders is an important aspect of the final appearance of the table. The default, Output 9.1, has light gray lines that surround each cell in the table. The FRAME= and RULES= attributes are often used to change the outside and inside borders, respectively.

Example 9.4 sets the FRAME= and RULES= attributes inside the Table element. The goal is to have a teal border around the outside of the table and a teal border between the headers and data. The Table element has the most predictable effect for changing the FRAME= and RULES= attributes for PowerPoint.

**Example 9.4 FRAME= and RULES= in the Table Element**

```
proc template;
define style _ppborder1;
parent=styles.powerpointlight;
class table /
    frame=box
    rules=groups
    borderwidth=2pt
    bordercolor=#00929F;
end;
run;
```

```

title;
ods powerpoint file="Output9-4.pptx" style=_ppborder1;
proc report data=orders(obs=20)
    style(report)=[pretext='this is pretext']
    style(header)=[backgroundcolor=white];

    column ("Customer Info" customer_country customer_age_group)
        ("Order Info" order_type quantity);

    define customer_country / group format=$cntry.;
    define customer_age_group / group;
    define order_type / group format=typef.;

    compute before customer_country;
        line 'a line statement for ' customer_country $cntry.;
    endcomp;

    break after customer_country / summarize;
run;
ods powerpoint close;

```

#### Output 9.4 FRAME= and RULES= in the Table Element

this is pretext			
Customer Info		Order Info	
Customer Country	Customer Age Group	Order Type	Quantity Ordered
a line statement for Australia			
Australia	31-45 years	Internet Sale	1
		Retail Sale	4
	46-60 years	Internet Sale	3
<i>Australia</i>			8
a line statement for United States			
United States	15-30 years	Internet Sale	2
		Retail Sale	11
	31-45 years	Retail Sale	7
	61-75 years	Retail Sale	1
<i>United States</i>			21

The Table element came close to providing the desired output. A teal border was successfully placed around the outside of the table and between the header section of the table and data. The problem with the slide shown in Output 9.4 is that light gray lines can still be seen (not as many as in Output 9.1, but some remain).

Example 9.5 removes the remaining gray lines by using the BORDERSTYLE= attribute inside the Data style element. The results are shown in Output 9.5.

#### Example 9.5 Creating Only the Desired Borders

```
proc template;
define style _ppborder2;
parent=styles.powerpointlight;
class table /
    frame=box
    rules=groups
    borderwidth=2pt
    bordercolor=#00929F;
class data /
    borderstyle=hidden;
class header /
    borderstyle=hidden;
end;
run;

title;
ods powerpoint file="Output9-5.pptx" style=_ppborder2;
proc report data=orders(obs=20)
    style(report)=[pretext='this is pretext']
    style(header)=[backgroundcolor=white] ;

    column ("Customer Info" customer_country customer_age_group)
        ("Order Info" order_type quantity);

    define customer_country / group format=$centry.;
    define customer_age_group / group;
    define order_type / group format=typef.;

    compute before customer_country;
        line 'a line statement for ' customer_country $centry.;
    endcomp;

    break after customer_country / summarize;
run;
ods powerpoint close;
```

**Output 9.5 Teal Borders with No Gray Lines**

this is pretext			
Customer Info		Order Info	
Customer Country	Customer Age Group	Order Type	Quantity Ordered
a line statement for Australia			
Australia	31-45 years	Internet Sale	1
		Retail Sale	4
	46-60 years	Internet Sale	3
a line statement for United States			
United States	15-30 years	Internet Sale	2
		Retail Sale	11
	31-45 years	Retail Sale	7
		61-75 years	Retail Sale
United States			21

Producing your desired borders might require a bit of trial and error work. The only way to determine whether your code will produce the needed borders is to test it. If the borders cannot be created with a custom style template, then overrides within PROC REPORT can be used to generate the borders.

All the gray lines can be removed with style overrides inside the PROC REPORT code. However, it is likely that you want the gray lines to be removed from all tables that will be included in the presentation. Therefore, the most efficient way to remove the borders is by altering the style template. Example 9.6 uses the BORDERSTYLE= attribute to remove the gray lines from the five style elements used by PROC REPORT.

**Example 9.6 Removing Gray Lines with the Style Template**

```
proc template;
define style _ppnogray;
parent=styles.powerpointlight;
class header, data, dataemphasis, linecontent, table /
borderstyle=hidden;
end;
run;
title;
```

```
ods powerpoint file="Output9-6.pptx" style=_ppnogray;
proc report data=orders(obs=20)
  style(report)=[pretext='this is pretext']
  style(header)=[backgroundcolor=white] ;

  column ("Customer Info" customer_country customer_age_group)
    ("Order Info" order_type quantity);

  define customer_country / group format=$cntry.;
  define customer_age_group / group;
  define order_type / group format=typef.;

  compute before customer_country;
    line 'a line statement for ' customer_country $cntry.;
  endcomp;

  break after customer_country / summarize;
run;
ods powerpoint close;
```

**Output 9.6 Gray Lines Removed from theTable**

this is pretext			
Customer Info		Order Info	
Customer Country	Customer Age Group	Order Type	Quantity Ordered
a line statement for Australia			
Australia	31-45 years	Internet Sale	1
		Retail Sale	4
	46-60 years	Internet Sale	3
<i>Australia</i>			8
a line statement for United States			
United States	15-30 years	Internet Sale	2
		Retail Sale	11
	31-45 years	Retail Sale	7
	61-75 years	Retail Sale	1
<i>United States</i>			21

The ODS destination for PowerPoint substitutes the SAS fonts Albany AMT, Cumberland AMT, and Thorndale AMT fonts for the Windows fonts Arial, Courier New, and Times New Roman, respectively.

## 9.4 Effects of Tables with Large Amounts of Data

The most difficult aspect of learning to develop PROC REPORT code for the PowerPoint destination is dealing with the change in scale. Again, PowerPoint presentations are meant to be viewed on a large screen. Although PowerPoint presentations are not designed for printing, they are often printed out as a convenience for later reading. The amount of data that can fit on one slide is not nearly the amount of data that can fit on one page in a PDF document. This section discusses PROC REPORT techniques that you can use to get a table to fit well in PowerPoint.

### 9.4.1 Many Columns of Data

The table in Outputs 9.1 and 9.2 has one row of pretext data, two header rows, and eleven rows of data. The table also contains four columns. The table occupies the entire slide, from top to bottom. The 11 rows of data just barely fit on the slide due to the font size, spacing, and other attributes of the table. The entire table that was used in previous chapters, like Chapter 3, did not fit on one slide. The RBREAK summary row along with the final LINE statement text had to be removed in order to get the table to fit on one slide. Even fewer rows of data would have fit on the slide if the code had included a title or footnote. Including the RBREAK and LINE statement forces the last row of data and the last row of text to flow to a second slide. Adding both a title and footnote forces four rows of data and the row of text to flow to a second slide.

In addition, Examples 9.1 and 9.2 do not specify any column widths for the data. The destination determines the width of the columns based on the formatted values of the variables along with the length of the header text. For this table, none of the headers or data wrapped within their cells because there was enough space on the slide.

Example 9.7 includes a fifth column, TOTAL\_RETAIL\_PRICE. The goal of this example is to demonstrate how adding another one column of data changes how the table fits on the slide. First, as shown in Output 9.7, the table now extends to a second slide. Second, all of the headers wrap within their cells. Third, the values of the ORDER\_TYPE column wrap within their cells. Output 9.7 highlights the ORDER\_TYPE column and the new column, TOTAL\_RETAIL\_PRICE.

#### Example 9.7 Adding Another Column to the Table

```
title;
ods powerpoint file="Output9-7.pptx";
proc report data=orders(obs=20)
    style(report)=[pretext='this is pretext']
    style(header)=[backgroundcolor=#F0F0F0];

    column ("Customer Info" customer_country customer_age_group)
        ("Order Info" order_type quantity total_retail_price);

    define customer_country / group format=$cntry.;
    define customer_age_group / group;
    define order_type / group format=typef. style(header
        column)=[backgroundcolor=#CCDF9C];
    define total_retail_price / style(header column)=[backgroundcolor=#68C9D2];
```

```
compute before customer_country;  
    line 'a line statement for ' customer_country $cntry.;  
endcomp;  
  
break after customer_country / summarize;  
  
run;  
ods powerpoint close;
```

Output 9.7 Table with Five Columns

this is pretext				
Customer Info		Order Info		
Customer Country	Customer Age Group	Order Type	Quantity Ordered	Total Retail Price for This Product
a line statement for Australia				
Australia	31-45 years	Internet Sale	1	\$21.80
		Retail Sale	4	\$223.80
	46-60 years	Internet Sale	3	\$37.80
Australia			8	\$283.40
a line statement for United States				
United States	15-30 years	Internet Sale	2	\$36.00

1



Customer Info		Order Info		
Customer Country	Customer Age Group	Order Type	Quantity Ordered	Total Retail Price for This Product
		Retail Sale	11	\$447.60
	31-45 years	Retail Sale	7	\$190.90
	61-75 years	Retail Sale	1	\$50.40
<i>United States</i>			21	\$724.90

2

Output 9.7 shows that just one small change (such as increasing from four columns to five columns) can produce big consequences. Admittedly, the table could be made smaller so that it fits on one slide, without compromising readability. Changes could be made with the style template as demonstrated earlier in this chapter. However, template changes affect every table that is sent to the destination. For the ODS destination for PowerPoint, it is important to consider all of the output that is being generated. Do all tables need to be altered the same way, regardless of whether they are created by PROC REPORT, PROC TABULATE, PROC FREQ, or another procedure? If so, it makes sense to create a custom style template. If only one table within the presentation is not fitting as desired, it is better to make the changes within that one procedure.

There is no right way to make sure that a table fits on a slide. Each table is different, but some of the same techniques can be used for each table. For example, the font size in Output 9.7 is 20pt. The font size can be reduced slightly to allow the table to fit. Example 9.8 demonstrates changing the font size for the table to 18pt. It also makes one other critical change. It changes the label of the TOTAL\_RETAIL\_PRICE variable. Output 9.8 shows the table that is now contained on one slide.

#### Example 9.8 Creating a Table with 18pt Font Size

```

title;
ods powerpoint file="Output9-8.pptx";
proc report data=orders(obs=20)
    style(report)=[pretext='this is pretext']
    style(header)=[backgroundcolor=#F0F0F0]
    style(report column header summary lines)=[fontsize=18pt];

```

```
column ("Customer Info" customer_country customer_age_group)
      ("Order Info" order_type quantity total_retail_price);

define customer_country / group format=$cntry.;
define customer_age_group / group;
define order_type / group format=typef. style(header
column)=[backgroundcolor=#CCDF9C];
define total_retail_price / 'Total/Retail Price' style(header
column)=[backgroundcolor=#68C9D2];

compute before customer_country;
  line 'a line statement for ' customer_country $cntry.;
endcomp;

break after customer_country / summarize;

run;
ods powerpoint close;
```

Output 9.8 Table with 18pt Font Size

this is pretext

Customer Info		Order Info		
Customer Country	Customer Age Group	Order Type	Quantity Ordered	Total Retail Price
a line statement for Australia				
Australia	31-45 years	Internet Sale	1	\$21.80
		Retail Sale	4	\$223.80
	46-60 years	Internet Sale	3	\$37.80
Australia			8	\$283.40
a line statement for United States				
United States	15-30 years	Internet Sale	2	\$36.00
		Retail Sale	11	\$447.60
	31-45 years	Retail Sale	7	\$190.90
	61-75 years	Retail Sale	1	\$50.40
United States			21	\$724.90

1

Notice that the ORDER\_TYPE values no long wrap within their cells. This is a direct result of changing the label for the TOTAL\_RETAIL\_PRICE variable. The shorter label means that the column does not have to be as wide, leaving more room for the other columns.

The font size of the text that was generated with the PRETEXT= attribute did not change. In other destinations, the report location can be used to change the attributes for PRETEXT=. For the ODS destination for PowerPoint, the attributes for the text cannot be controlled through changes in PROC REPORT. The changes must be made through a style template.

Example 9.9 changes the font size attribute in the Note element, which controls the PRETEXT= attribute in PowerPoint. The table with the smaller font size is shown in Output 9.9.

#### Example 9.9 Changing the Pretext Font Size

```
proc template;
define style _pretext;
parent = styles.powerpointlight;
class Note /
    fontsize = 18pt
    fontfamily='Albany AMT'
    color=#8E2F8A
    fontweight=bold;
end;
run;

title;
ods powerpoint file="Output9-9.pptx" style=_pretext;
proc report data=orders(obs=20)
    style(report)=[pretext='this is pretext']
    style(header)=[backgroundcolor=#F0F0F0]
    style(report column header summary lines)=[fontsize=18pt];

    column ("Customer Info" customer_country customer_age_group)
        ("Order Info" order_type quantity total_retail_price);

    define customer_country / group format=$cntry.;
    define customer_age_group / group;
    define order_type / group format=typef.;
    define total_retail_price / 'Total/Retail Price';

    compute before customer_country;
        line 'a line statement for ' customer_country $cntry.;
    endcomp;

    break after customer_country / summarize;
run;
ods powerpoint close;
```

**Output 9.9 Table with 18pt Font Size for Pretext****this is pretext**

Customer Info		Order Info		
Customer Country	Customer Age Group	Order Type	Quantity Ordered	Total Retail Price
a line statement for Australia				
Australia	31-45 years	Internet Sale	1	\$21.80
		Retail Sale	4	\$223.80
	46-60 years	Internet Sale	3	\$37.80
<i>Australia</i>			8	\$283.40
a line statement for United States				
United States	15-30 years	Internet Sale	2	\$36.00
		Retail Sale	11	\$447.60
	31-45 years	Retail Sale	7	\$190.90
	61-75 years	Retail Sale	1	\$50.40
<i>United States</i>			21	\$724.90

1

**9.4.2 Many Columns and Rows of Data**

The ODS destination for PowerPoint automatically breaks a table into multiple panels when it is too long or too wide to fit on a single slide. For long tables, a subset of rows is placed on another slide. For wide tables, the destination creates subsets of columns, and each subset is placed on its own slide.

Suppose that one of the required tables is both long and wide. Example 9.10 produces a table that has 7 columns and 66 rows. A PDF file fits more data per page, but even in PDF, 66 rows might not fit on one page. Even though Output 9.10 shows only the first two slides that are created by Example 9.10, the entire table actually spans 21 slides!

**Example 9.10 Creating a Long and Wide Table**

```

title;
ods powerpoint file="Output9-10.pptx";
proc report data=orders
    style(header)=[backgroundcolor=#3D5AAE color=white];

    column ("Customer Info" customer_country customer_group customer_age_group)
        ("Order Info" order_type costprice_per_unit quantity
        total_retail_price);

```

```

define customer_country / group format=$centry.
    style(column)=[backgroundcolor=#F0F0F0];
define customer_group / group style(column)=[backgroundcolor=#F0F0F0];
define customer_age_group / group style(column)=[backgroundcolor=#F0F0F0];
define order_type / group format=typef.
    style(column)=[backgroundcolor=#CCDF9C];
define costprice_per_unit / style(column)=[backgroundcolor=#CCDF9C];
define quantity / style(column)=[backgroundcolor=#CCDF9C];
define total_retail_price / style(column)=[backgroundcolor=#CCDF9C];

break after customer_country / summarize;
run;
ods powerpoint close;

```

#### Output 9.10 Slides for the Long and Wide Table

Customer Info		
Customer Country	Customer Group Name	Customer Age Group
Australia	Internet/Catalog Customers	61-75 years
	Orion Club Gold members	46-60 years
	Orion Club members	31-45 years
		46-60 years

Order Info			
Order Type	Cost Price Per Unit	Quantity Ordered	Total Retail Price for This Product
Catalog Sale	\$429.60	13	\$1,061.30
Retail Sale	\$552.05	24	\$1,545.50
Catalog Sale	\$131.15	12	\$515.20
Internet Sale	\$132.60	8	\$395.70
Retail Sale	\$1,616.05	84	\$5,696.69
Catalog Sale	\$46.90	1	\$102.90
Internet Sale	\$44.00	5	\$218.20

2

The entire table must be included in the presentation, but it needs to be more readable. The table must span fewer slides to make the presentation more useful. Here are some constructive criticisms of Output 9.10:

- All the columns should stay together on one slide.
- The Total Retail Price column is too wide.
- The Total Retail Price column header is too wordy.
- The information for a country should be kept together and not span across slides.

Example 9.11 demonstrates the techniques for altering a table that is both long and wide so that it fits better in the presentation. Output 9.11 shows the first slide of the resulting presentation.

#### Example 9.11 Creating a Long and Wide Table

```

title;
ods powerpoint file="Output9-11.pptx";
proc report data=orders
    style(header)=[backgroundcolor=#3D5AAE color=white]
    style(report column header summary)=[fontsize=16pt]; ❶

    column ("Customer Info" customer_country customer_group customer_age_group)
           ("Order Info" order_type costprice_per_unit quantity
            total_retail_price);

```

```

define customer_country / group format=$cntry. 'Country'
    style(column)=[backgroundcolor=#F0F0F0]; ❷
define customer_group / group 'Group' style(column)=[backgroundcolor=#F0F0F0
    width=1.5in]; ❸
define customer_age_group / group 'Age Group'
    style(column)=[backgroundcolor=#F0F0F0]; ❹
define order_type / group format=typef. style(column)=[backgroundcolor=#CCDF9C
    width=1.3in]; ❺
define costprice_per_unit / 'Cost Price'
    style(column)=[backgroundcolor=#CCDF9C width=1.2in]; ❻
define quantity / style(column)=[backgroundcolor=#CCDF9C width=1.1in];
define total_retail_price / 'Total Retail Price'
    style(column)=[backgroundcolor=#CCDF9C width=1.2in]; ❼
    break after customer_country / summarize page; ❽
run;
ods powerpoint close;

```

- ❶ The font size is reduced to 16pt. Although this change is made in the PROC REPORT statement, it was not the first change made to get the table to fit better on the slides. Changing the font size was necessary but it should be considered only after other attempts to get the table to fit as desired have failed.
- ❷ The CUSTOMER\_COUNTRY label is reduced to one word.
- ❸ The CUSTOMER\_GROUP label is reduced to one word. The WIDTH= attribute is set to 1.5 inches.
- ❹ The CUSTOMER\_AGE\_GROUP label is changed to “Age Group”.
- ❺ The width for the ORDER\_TYPE variable is set to 1.3 inches.
- ❻ The COSTPRICE\_PER\_UNIT label is changed to “Cost Price”. The WIDTH= attribute is set to 1.2 inches.
- ❼ The label for TOTAL\_RETAIL\_PRICE is reduced, and the width is set to 1.2 inches.
- ❽ The PAGE option in the BREAK statement divides the long table into multiple short tables, one for each country. Each table, one for each country, starts on a new slide.

**Output 9.11 Modified Long and Wide Table (Slides 1 and 2)**

Customer Info			Order Info			
Country	Group	Age Group	Order Type	Cost Price	Quantity Ordered	Total Retail Price
Australia	Internet /Catalog Customers	61-75 years	Catalog Sale	\$429.60	13	\$1,061.30
	Orion Club Gold members	46-60 years	Retail Sale	\$552.05	24	\$1,545.50
	Orion Club members	31-45 years	Catalog Sale	\$131.15	12	\$515.20
			Internet Sale	\$132.60	8	\$395.70
			Retail Sale	\$1,616.05	84	\$5,696.69
		46-60 years	Catalog Sale	\$46.90	1	\$102.90
			Internet Sale	\$44.00	5	\$218.20
			Retail Sale	\$464.15	14	\$1,584.30
		61-75 years	Retail Sale	\$1,434.85	49	\$6,201.70
<i>Australia</i>				<i>\$4,851.35</i>	<i>210</i>	<i>\$17,321.49</i>



Customer Info			Order Info			
Country	Group	Age Group	Order Type	Cost Price	Quantity Ordered	Total Retail Price
Canada	Internet /Catalog Customers	31-45 years	Catalog Sale	\$132.55	5	\$679.30
	Orion Club Gold members	15-30 years	Internet Sale	\$101.70	8	\$611.50
		31-45 years	Internet Sale	\$390.65	11	\$1,812.90
		46-60 years	Internet Sale	\$161.30	11	\$815.40
	Orion Club members	15-30 years	Catalog Sale	\$583.35	18	\$4,395.58
			Internet Sale	\$226.40	18	\$1,117.50
		31-45 years	Internet Sale	\$456.00	15	\$1,654.20
		46-60 years	Internet Sale	\$124.85	7	\$343.50
		61-75 years	Catalog Sale	\$87.95	9	\$347.50
			Internet Sale	\$29.05	3	\$173.70
Canada				\$2,293.80	105	\$11,951.08

2

Although Output 9.11 displays only the first two slides, Example 9.11 generates nine slides. That is an improvement of 12 slides. The font size might be smaller, but readability has improved.

Only eight changes were made to the code in Example 9.10 to produce the code in Example 9.11. However, many attempts at creating a desirable slide were required to create the final code. It is not easy to shrink tables for the PowerPoint destination. Producing smaller, attractive tables might take trial and error. Also, it is important to keep in mind that some tables just might not fit on the slide. In those instances, a new strategy for conveying the required information might be preferable.

The PAGE option was used in the BREAK statement in Example 9.11. The PAGE option can also be used in the DEFINE statement. The PAGE option used in the DEFINE statement causes a vertical page break. In other words, it divides sets of columns into separate, smaller tables. The PAGE option in the DEFINE statement is extremely useful for very wide tables.

The PAGE option in the DEFINE statement is also a good strategy for dividing columns in a logical manner. Example 9.12 uses the PAGE option in the DEFINE statement for COSTPRICE\_PER\_UNIT so that it stays together with TOTAL\_RETAIL\_PRICE. The result is two tables. The first table contains the customer variables, CUSTOMER\_AGE\_GROUP and CUSTOMER\_GROUP, along with QUANTITY. The second table contains the ID variables and COSTPRICE\_PER\_UNIT and TOTAL\_RETAIL\_PRICE. Both tables are shown in Output 9.12.

Example 9.12 Utilizing PAGE Option in the DEFINE Statement

```
title;
ods powerpoint file="Output9-12.pptx";
proc report data=orders style(header)=[backgroundcolor=#3D5AAE color=white];
  column customer_age_group customer_group quantity costprice_per_unit
         total_retail_price;
  define customer_age_group / group style(column)=[backgroundcolor=#F0F0F0];
  define customer_group / group id style(column)=[backgroundcolor=#F0F0F0];
  define quantity / style(column)=[backgroundcolor=#6EC6F0];
  define costprice_per_unit / page style(column)=[width=1.4in
         backgroundcolor=#E7ACC6];
  define total_retail_price / "Total Retail Price" style(column)=[width=1.4in
         backgroundcolor=#E7ACC6];run;
ods powerpoint close;
```

Output 9.12 PAGE Option Generates Two Tables

Customer Age Group	Customer Group Name	Quantity Ordered
15-30 years	Orion Club Gold members	132
	Orion Club members	172
31-45 years	Internet/Catalog Customers	38
	Orion Club Gold members	102
46-60 years	Orion Club members	257
	Internet/Catalog Customers	39
	Orion Club Gold members	56
61-75 years	Orion Club members	80
	Internet/Catalog Customers	61
	Orion Club Gold members	18
	Orion Club members	123

Customer Age Group	Customer Group Name	Cost Price Per Unit	Total Retail Price
15-30 years	Orion Club Gold members	\$3,332.70	\$12,101.83
	Orion Club members	\$3,812.90	\$18,384.67
31-45 years	Internet/Catalog Customers	\$695.60	\$3,284.65
	Orion Club Gold members	\$3,438.45	\$12,641.69
	Orion Club members	\$5,254.98	\$18,271.16
46-60 years	Internet/Catalog Customers	\$719.40	\$3,288.60
	Orion Club Gold members	\$1,203.50	\$4,397.30
	Orion Club members	\$2,141.60	\$7,234.65
61-75 years	Internet/Catalog Customers	\$2,125.15	\$7,607.20
	Orion Club Gold members	\$643.40	\$2,640.20
	Orion Club members	\$2,438.45	\$10,225.52

2

### 9.4.3 Many Header Rows

Spanning headers and headers and that are generated by ACROSS variables tend to make many rows of headers. In PDF or HTML, a table with three or four rows of headers is a common practice. However, in PowerPoint, excessive headers rows might not be a good idea. The number of header rows affects the number of data rows that can fit on a slide.

Example 9.13 has an ACROSS variable, ORDER\_DATE. The variable QUANTITY becomes almost like an ACROSS variable because three statistics are stacked below it. The result of having these two ACROSS variables is shown in Output 9.13. The header section has four rows of text, but it requires five rows to display because two of the headers wrap within the cell. The table length requires a second slide.

#### Example 9.13 Using Two ACROSS Variables

```

title;
ods powerpoint file="Output9-13.pptx";
proc report data=orders(obs=70) style(header)=[backgroundcolor=#3D5AAE color=white];
    where year(order_date)<2005;

    column ("Customer Info" customer_country customer_age_group)
           order_date,(quantity, (sum min max));

```

```
define customer_country / group format=$cntry.  
    style(column)=[backgroundcolor=#F0F0F0];  
define customer_age_group / group style(column)=[background=#F0F0F0];  
define order_date / across format=year4.;  
define sum / style(column)=[backgroundcolor=#FDC88F];  
define min / style(column)=[backgroundcolor=#FDC88F];  
define max / style(column)=[backgroundcolor=#FDC88F];  
  
run;  
ods powerpoint close;
```

Output 9.13 Header Section Generated by Two ACROSS Variables

		Date Order was placed by Customer					
		2003			2004		
Customer Info		Quantity Ordered			Quantity Ordered		
Customer Country	Customer Age Group	sum	min	max	sum	min	max
Australia	31-45 years	10	1	4	6	1	2
	61-75 years	4	4	4	.	.	.
Canada	61-75 years	1	1	1	3	3	3
Germany	15-30 years	.	.	.	1	1	1
	61-75 years	8	1	3	.	.	.
Israel	15-30 years	.	.	.	2	1	1
	31-45 years	5	1	2	2	2	2
	46-60 years	4	1	1	5	2	3
United States	15-30 years	9	1	2	17	1	2

1

		Date Order was placed by Customer					
		2003			2004		
Customer Info		Quantity Ordered			Quantity Ordered		
Customer Country	Customer Age Group	sum	min	max	sum	min	max
	31-45 years	20	1	4	1	1	1
	46-60 years	10	1	3	2	2	2
	61-75 years	7	2	3	1	1	1

2

Some changes can be made to limit the number of header rows in an attempt to force the table from Example 9.13 to fit onto one slide. Keep in mind that the acceptable changes vary from table to table and according to business need. For example, the following code removes the word “quantity” from the headers. Without other context, the audience, or reader of the slides, might not know what statistics are used. You must be careful when trying to limit the headers rows. You do not want to lose the meaning of the table. The technique in Example 9.14 illustrates possible changes that could be made.

Example 9.14 demonstrates how to modify the COLUMN statement and labels for the ACROSS variables. Output 9.14 shows a table that fits on one slide.

#### Example 9.14 Modifying Labels to Change ACROSS Headers

```
options nonumber;
title;
ods powerpoint file="Output9-14.pptx";
proc report data=orders(obs=70)
    style(header)=[backgroundcolor=#3D5AAE color=white];
    where year(order_date)<2005;

    column ("Customer Info" "" ❶ customer_country customer_age_group)
        order_date,(quantity, (sum min max));

    define customer_country / group format=$cntry.
        style(column)=[backgroundcolor=#F0F0F0];
```

```
define customer_age_group / group style(column)=[backgroundcolor=#F0F0F0];
define order_date / across format=year4. ''; ❷
define quantity / ''; ❸
define sum / style(column)=[backgroundcolor=#FDC88F];
define min / style(column)=[backgroundcolor=#FDC88F];
define max / style(column)=[backgroundcolor=#FDC88F];

run;
ods powerpoint close;
```

- ❶ A blank spanning header is added so that the “Customer Info’ spanning header is moved up next to the year value. The blank spanning header ensures that the header row that contained the QUANTITY label is suppressed.
- ❷ The ORDER\_DATE variable’s label is set to missing so that the header row is suppressed.
- ❸ The QUANTITY variable’s label is set to missing so that the header row is suppressed.

Output 9.14 Modified ACROSS Labels Change Header Section

Customer Info		2003			2004		
Customer Country	Customer Age Group	sum	min	max	sum	min	max
Australia	31-45 years	10	1	4	6	1	2
	61-75 years	4	4	4	.	.	.
Canada	61-75 years	1	1	1	3	3	3
Germany	15-30 years	.	.	.	1	1	1
	61-75 years	8	1	3	.	.	.
Israel	15-30 years	.	.	.	2	1	1
	31-45 years	5	1	2	2	2	2
	46-60 years	4	1	1	5	2	3
United States	15-30 years	9	1	2	17	1	2
	31-45 years	20	1	4	1	1	1
	46-60 years	10	1	3	2	2	2
	61-75 years	7	2	3	1	1	1

Even with the modifications made in Example 9.14, the table barely fits on the slide in Output 9.14. Also, the slide does not contain the slide number. The entire table would not fit on the slide if the slide number was included.

The examples in this chapter have limited the number of observations from the input data set. Also, these examples do not have titles or footnotes. Hopefully, this demonstrates that even when using a small number of observations without titles or footnotes, and sometimes without a slide number, it is difficult to prevent a table from wrapping to a second or third slide. Even more adjustments might be required to fit tables on slides when multiple titles are used. Often, because company logos are placed on slides, they must be taken into account when altering tables to fit on slides.

Trial and error is required to get tables to fit into a presentation and in a pleasing manner. Changes such as removing header rows or breaking the table apart are techniques that help achieve the desired appearance. Each presentation is different. Remember to set and maintain reasonable expectations.

The ODS destination for PowerPoint also allows the use of ODS LAYOUT. ODS LAYOUT allows multiple objects such as tables and graphs to be placed on a single slide. If a table does not fit inside a layout container, it will be suppressed. Keeping a table small and compact is vital when placing it inside a layout container for PowerPoint.

## 9.5 Inline Formating Functions

PROC REPORT tables often contain extra formatting techniques (such as colors, images, dollar signs, and Unicode values) to enhance the display of information. Trafficlighting and numeric formatting work the same way in the PowerPoint destination as they do in all other destinations for PROC REPORT. Images require a specific image attribute, which is discussed in Section 9.6.

Inline formatting is a popular way of adding features to the report, but not all functions are supported for the destination. The PowerPoint destination supports these inline functions:

- dagger
- nbspace
- newline
- sigma
- style
- sub
- super
- unicode

This list covers most of the needs of PROC REPORT output. However, it is important to note the inline functions that are not supported by the destination. Most of the functions that are not supported are generally used outside tables and are more likely found in TITLE and FOOTNOTE statements. Also, some of the functions are specific to other ODS destinations. The PowerPoint destination does not support these functions:

- date
- dest
- lastpage
- leaders

- pageof
- raw
- thispage
- tocentryindent
- tocentrypage

Also, some cell attributes, used with inline formatting, cannot be changed. They include the following:

- background of a shape (for example, the color of a word)
- pre-text and post-text
- alignment
- first-line text indentation

You might be able to apply the attribute using a CALL DEFINE statement. However, you will need to test the attribute. Example 9.15 creates a new variable within a DATA step that has inline formatting and that includes POSTTEXT=. The asterisk does not appear in the column for C\_G in the table, Output 9.15. The asterisk does appear for CUSTOMER\_GENDER, which used a CALL DEFINE statement to apply the POSTTEXT= attribute.

#### Example 9.15 Using POSTTEXT=

```
ods escapechar="~";
data orders2;
    set orders;
    length c_g $50;
    c_g = catt('~{style [posttext="*"]',customer_gender,'}');
run;

ods powerpoint file="Output9-15.pptx" dpi=300;
proc report data=orders2 style(header)=[backgroundcolor=#CCDF9C];
    column c_g customer_gender n;
    define c_g / group;
    define customer_gender / group;

    compute customer_gender;
        if customer_gender='F' then
            call define(_col_, 'style', 'style=[posttext="*"]');
        if customer_gender='F' then
            call define(_row_, 'style', 'style=[color=#8E2F8A]');
        else call define(_row_, 'style', 'style=[color=#3D5AAE]');
    endcomp;
run;
ods powerpoint close;
```



**Output 9.15 POSTTEXT= in Inline Formatting Is Ignored**

c_g	Customer Gender	n
F	F*	277
M	M	340

1

The workaround for the first line indentation functionality is to use the supported NBSPACE and NEWLINE functions. Example 9.17 demonstrates how to concatenate three variables together, add a newline, and add a space before the second and third values. The results are shown in Output 9.17.

**Example 9.16 Using the NEWLINE and NBSPACE Functions**

```
ods escapechar="^";
ods powerpoint file="Output9-16.pptx";
proc report data=orders(obs=5) style(header)=[backgroundcolor=#62BAE9];
  column customer_name customer_gender customer_age customer;
  define customer_age / display;
  define customer / computed "Customer Info"
    style(column)=[backgroundcolor=#FFE7A1];
  compute customer / char length=100;
    customer = catx("^{\newline 1}^{\nbspace 5}",
      customer_name,customer_gender,customer_age);
  endcomp;
run;
ods powerpoint close;
```

Output 9.16 Gender and Age Appear on Separate Rows and Are Indented

Customer Name	Customer Gender	Customer Age	Customer Info
Kyndal Hooks	F	43	Kyndal Hooks F 43
Annmarie Leveille	F	23	Annmarie Leveille F 23
Najma Hicks	F	21	Najma Hicks F 21
Yan Kozlowski	M	38	Yan Kozlowski M 38
Kyndal Hooks	F	43	Kyndal Hooks F 43

## 9.6 Style Attributes for Images

Images are important for most presentations. They attract the audience’s attention and provide useful information. Images can be added above and below a PROC REPORT table with the PREIMAGE= and POSTIMAGE= attributes, respectively. However, these attributes cannot be used to include images inside table cells. Microsoft PowerPoint does not allow pre- and postimages. Only background images can be used inside table cells. In other words, the BACKGROUNDIMAGE= attribute can be used for a table cell, but the PREIMAGE= or POSTIMAGE= attributes cannot be used for a table cell. By allowing background images, the PowerPoint destination is similar to HTML, but is different from PDF or RTF.

Example 9.17 places an image above and below the table using the PREIMAGE= and POSTIMAGE= attributes within the STYLE(REPORT) location. The results are shown in Output 9.17.

**Example 9.17 Using PREIMAGE= and POSTIMAGE=**



```
ods powerpoint file="Output9-17.pptx";
proc report data=orders
  style(header)=[backgroundcolor=#E7ACC6]
  style(report)=[preimage="cologo.tif" postimage="cologo.tif"];
```

```

column customer_type n;
define customer_type / group;
run;
ods powerpoint close;

```

**Output 9.17 Images Placed Above and Below the Table**

ABC 	
Customer Type Name	n
Internet/Catalog Customers	76
Orion Club members high activity	115
Orion Club members medium activity	154
Orion Club Gold members high activity	65
Orion Club Gold members low activity	42
Orion Club Gold members medium activity	77
Orion Club members low activity	88
ABC 	

1

Example 9.18 places images inside the table cells. The BACKGROUNDIMAGE= attribute must be used. Using a different image attribute will result in a blank cell. Output 9.18 shows the results.

**Example 9.18 Using BACKGROUNDIMAGE=**




```

ods powerpoint file="Output9-18.pptx";
proc report data=orders style(header)=[backgroundcolor=#62BAE9];
  column order_type;
  define order_type / group format=typef.
    style(column)=[textalign=c width=2in height=2in];
  compute order_type;
    if order_type = 1 then call define(_col_, 'style',
      'style=[backgroundimage="goldmedal.tiff"]');
    else if order_type = 2 then call define(_col_, 'style',
      'style=[backgroundimage="silvermedal.tiff"]');
    else if order_type = 3 then call define(_col_, 'style',
      'style=[backgroundimage="bronzemedal.tiff"]');
  endcomp;
run;

ods powerpoint close;

```

**Output 9.18 Table Cells with Background Images**

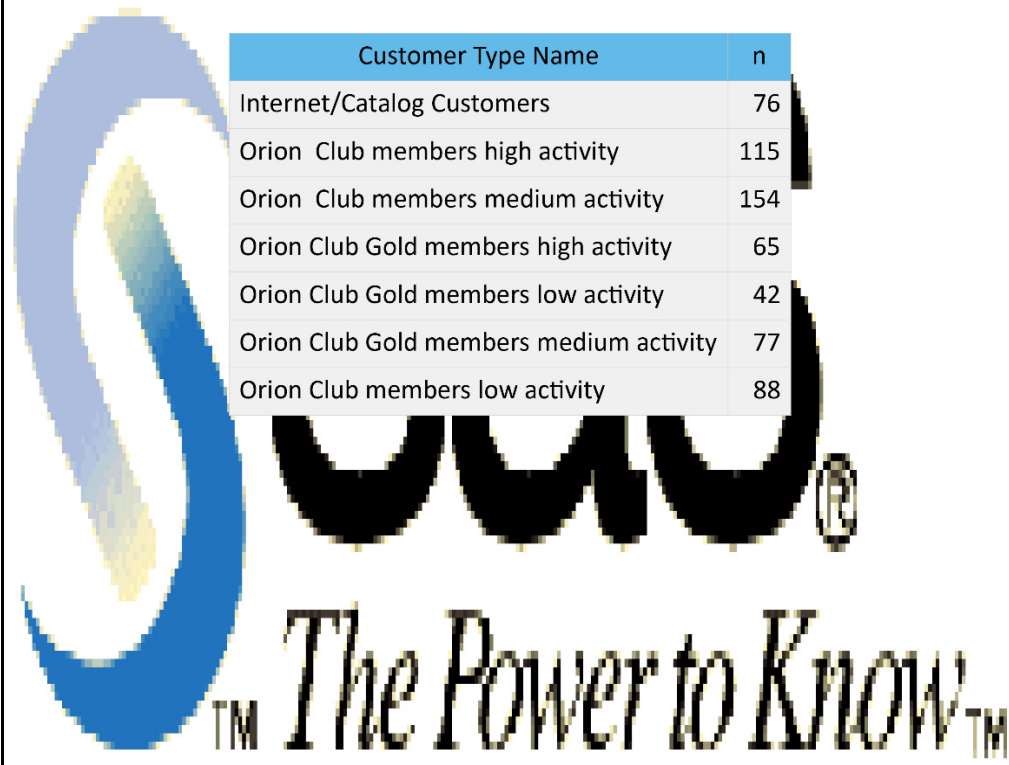
Order Type
Catalog Sale 
Internet Sale 
Retail Sale 

When working with images for this destination, it is important to understand that by default the image will be stretched or shrunk to fit the cell. The destination ensures that the image fits inside the box that encloses it.

The entire slide can have a background image, as well. Example 9.19 demonstrates how to include a background image for the slide, which is specified in the ODS POWERPOINT statement. Because no other suboptions are specified, the image is stretched to fit the full slide. A background color is set for the columns in the table so that the table values are still readable, as shown in Output 9.19.

**Example 9.19 Use BACKGROUNDIMAGE= for the Slide**

```
ods powerpoint file="Output9-19.pptx"
  options(backgroundimage="saslogo1.png");
proc report data=orders
  style(header)=[backgroundcolor=#62BAE9]
  style(column)=[backgroundcolor=#F0F0F0];
  column customer_type n;
  define customer_type / group;
run;
ods powerpoint close;
```

**Output 9.19 Background Image Stretches to Fit the Slide**



Customer Type Name	n
Internet/Catalog Customers	76
Orion Club members high activity	115
Orion Club members medium activity	154
Orion Club Gold members high activity	65
Orion Club Gold members low activity	42
Orion Club Gold members medium activity	77
Orion Club members low activity	88

The background image in Output 9.19 is pixelated because it has been stretched. There are two methods for preventing this behavior. The first is to create an image that is the size of the slide. The second option is to use the BACKGROUNDREPEAT= suboption.

Example 9.20 demonstrates how to use an image that is the size of the slide. Output 9.20 shows the image in an appropriate size.

**Example 9.20 Use an Image That Is the Size of the Slide**

```
ods powerpoint file="Output9-20.pptx"
  options(backgroundimage="saslogo_pptx.png");
proc report data=orders
  style(header)=[backgroundcolor=#62BAE9]
  style(column)=[backgroundcolor=#F0F0F0];
  column customer_type n;
  define customer_type / group;
run;
ods powerpoint close;
```

**Output 9.20 Image Fits on the Slide and Is Not Stretched**


Customer Type Name	n
Internet/Catalog Customers	76
Orion Club members high activity	115
Orion Club members medium activity	154
Orion Club Gold members high activity	65
Orion Club Gold members low activity	42
Orion Club Gold members medium activity	77
Orion Club members low activity	88

The image will retain its original size if the BACKGROUNDREPEAT= suboption is set to REPEAT, REPEAT\_X, or REPEAT\_Y. Example 9.21 demonstrates how to use the original, smaller image along with the BACKGROUNDREPEAT= suboption. Output 9.21 shows the effect of the suboption.

**Example 9.21 Using BACKGROUNDREPEAT=**

```
ods powerpoint file="Output9-21.pptx"
  options(backgroundimage="saslogo1.png" backgroundrepeat="repeat");
proc report data=orders
  style(header)=[backgroundcolor=#62BAE9]
  style(column)=[backgroundcolor=#F0F0F0];
  column customer_type n;
  define customer_type / group;
run;
ods powerpoint close;
```

Output 9.21 Image Is Repeated

Customer Type Name		n
Internet/Catalog Customers		76
Orion Club members high activity		115
Orion Club members medium activity		154
Orion Club Gold members high activity		65
Orion Club Gold members low activity		42
Orion Club Gold members medium activity		77
Orion Club members low activity		88

## 9.7 Key Takeaways

This chapter provided a lot of information about handling the unique aspects of the ODS destination for PowerPoint. The beginning of the chapter discussed the changes that can be made with the style template. The important takeaways regarding the style template are the following:

- Always use PowerPointLight or PowerPointDark as the parent style when creating a custom style template for PowerPoint.
- Make changes to the style elements only when those changes are needed for every table in the presentation.
- BORDERSTYLE=HIDDEN removes the gray border lines inside the table.

This chapter also covered programming techniques to help ensure that the PROC REPORT table displays as desired in the PowerPoint slide. Key takeaways include the following:

- Limit the number of columns and rows in a table as much as possible.
- Keep the number of header rows to a minimum.
- Use the PAGE option in the DEFINE and BREAK statements to divide large tables into smaller tables.



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



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

**[support.sas.com/newbooks](https://support.sas.com/newbooks)**

Share your expertise. Write a book with SAS.

**[support.sas.com/publish](https://support.sas.com/publish)**

 [sas.com/books](https://sas.com/books)  
for additional books and resources.

  
THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.  
Other brand and product names are trademarks of their respective companies. © 2017 SAS Institute Inc. All rights reserved. M1588358 US.0217