

## A Second Look at the ODS Destination for PowerPoint

Tim Hunter, SAS Institute Inc., Cary, NC

### ABSTRACT

This paper demonstrates how to use the ODS destination for PowerPoint to create attractive presentations from your SAS® output. Packed with examples, this paper gives you a behind-the-scenes tour of how ODS creates Microsoft PowerPoint presentations. You get an in-depth look at how to customize the ODS PowerPoint style templates that control the appearance of your presentation. With this information you can quickly turn your SAS output into an engaging and informative presentation.

This paper is a follow-on to the SAS® Global Forum 2013 paper “A First Look at the ODS Destination for PowerPoint.”

### INTRODUCTION

The SAS Output Delivery System (ODS) gives you the ability to store SAS procedure and DATA step output into many different formats such as HTML and PDF. SAS 9.4 introduced the ODS destination for PowerPoint. This destination adds support for creating documents in Microsoft PowerPoint 2010 and 2013 format. Just like in other destinations, ODS offers many ways for you to customize the appearance of your PowerPoint documents. This paper explains how the destination for PowerPoint creates a slide and gives you a detailed explanation of how to customize the default appearance to meet your own particular needs.

### HOW TO USE THE DESTINATION FOR POWERPOINT

All ODS users know about the ODS sandwich. The bread in the sandwich is two ODS statements, one to start putting output into the document and one to stop. The sandwich filling is your procedure and DATA step code. The first ODS statement identifies the “destination,” that is, a document format such as Excel or HTML. For this paper, the destination is always POWERPOINT. Here’s the prototypical example of an ODS sandwich for the destination for PowerPoint:

```
ods powerpoint file="my_presentation.pptx";  
    your code goes here...  
ods powerpoint close;
```

The ODS POWERPOINT statement can also specify options to customize your output. You’ll learn about some of these options in the next section.

### A NOTE BEFORE WE BEGIN

This paper frequently describes the document created by the destination for PowerPoint as a “presentation.” It is more accurate to say that the document is SAS output in Microsoft PowerPoint format. While you can put SAS output such as tables, graphs, lists, and paragraph text in your document, there is no mechanism for adding common presentation elements such as notes pages and animations.

### WHAT’S NEW IN SAS 9.4M3

New in SAS 9.4M3 are two frequently requested features. The first is the ability to change the background of individual slides. The second is the ability to add a transition between slides.

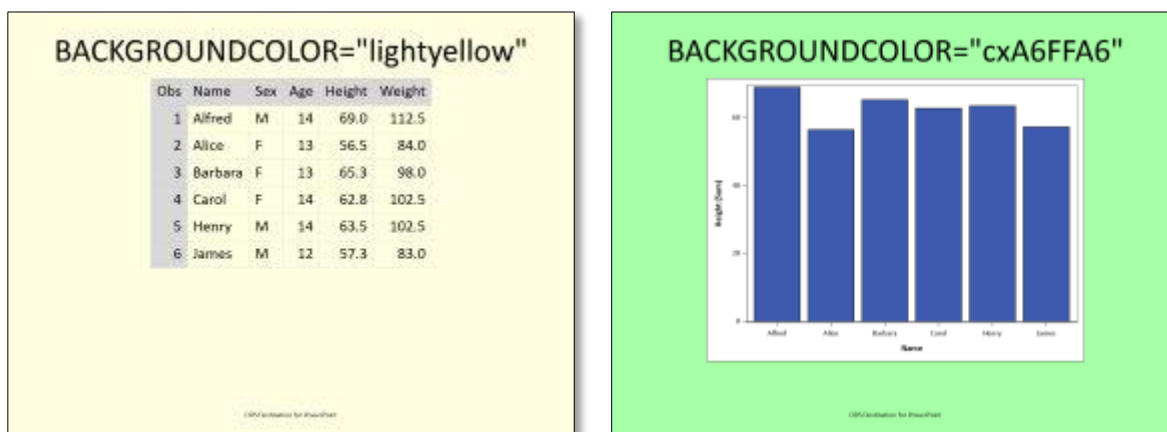
### CHANGE THE BACKGROUND OF INDIVIDUAL SLIDES

Before SAS 9.4M3, the only way to add a custom background to the slides in a presentation is to create a custom style template. This technique has the shortcoming that the same background is used for all the slides. Frequently, it is useful to use different backgrounds for different slide types. Starting in SAS 9.4M3, you can customize the background of any number of slides in your presentation by specifying either the

BACKGROUNDIMAGE suboptions in the OPTIONS option in the ODS POWERPOINT statement.

Use the BACKGROUNDIMAGE suboption to specify a solid color background. The value of the suboption can be one of the color names that are supported by SAS/GRAPH® or an RGB color with a leading pound sign (#) rather than CX. Here's an example of how to use this suboption to have a light yellow background on the first slide and a light green background on the second slide:

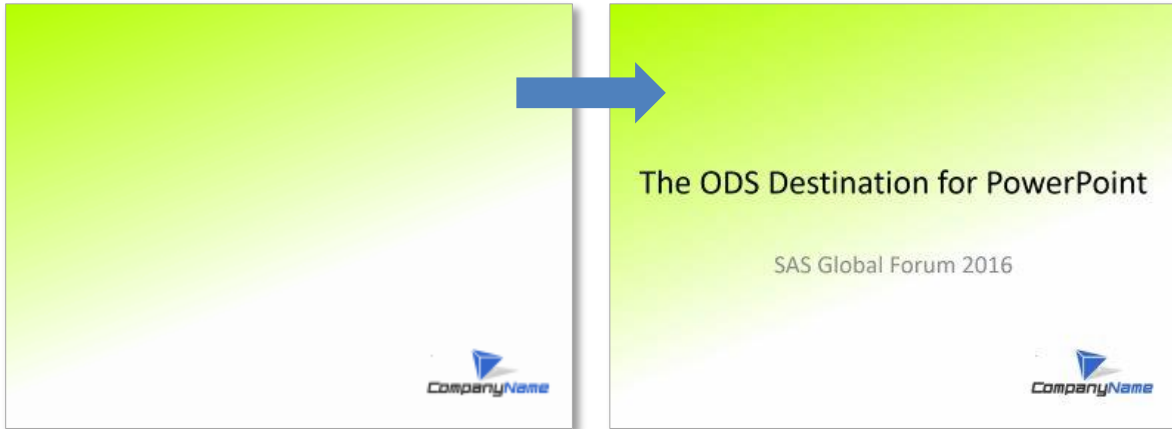
```
ods powerpoint file="Example_1.pptx" nogtitle nogfootnote;
title 'BACKGROUNDIMAGE="lightyellow"';
ods powerpoint options(backgroundcolor="lightyellow");
proc print data=sashelp.class(obs=6);
run;
title 'BACKGROUNDIMAGE="cxA6FFA6"';
ods powerpoint options(backgroundcolor="cxA6FFA6");
proc sgplot data=sashelp.class(obs=6);
vbar name / response=height;
run;
ods powerpoint close;
```



**Figure 1. Change the Background Color of Individual Slides**

Use the BACKGROUNDIMAGE suboption to specify an image or gradient background. To specify an image, specify the path to the image file or a fileref to the image file as the suboption value. Here's how to add a company logo to the background of your title slide. Start by creating your desired background image in an image editor. The image should have the same aspect ratio as your presentation. For a 4:3 presentation create an image that is 1024 pixels wide and 768 pixels tall. For a 10:9 presentation use 1280 x 720 pixels. Specify your image as the background behind the title slide. Example 2 uses the TitleSlide layout template to create a title slide with a company logo in the bottom right of the slide:

```
ods powerpoint file="Example_2.pptx" layout=titleslide
options(backgroundimage="CompanyName_bg.gif");
proc odstext;
p "The ODS Destination for PowerPoint" / style=PresentationTitle;
p "SAS Global Forum 2016" / style=PresentationTitle2;
run;
ods powerpoint close;
```



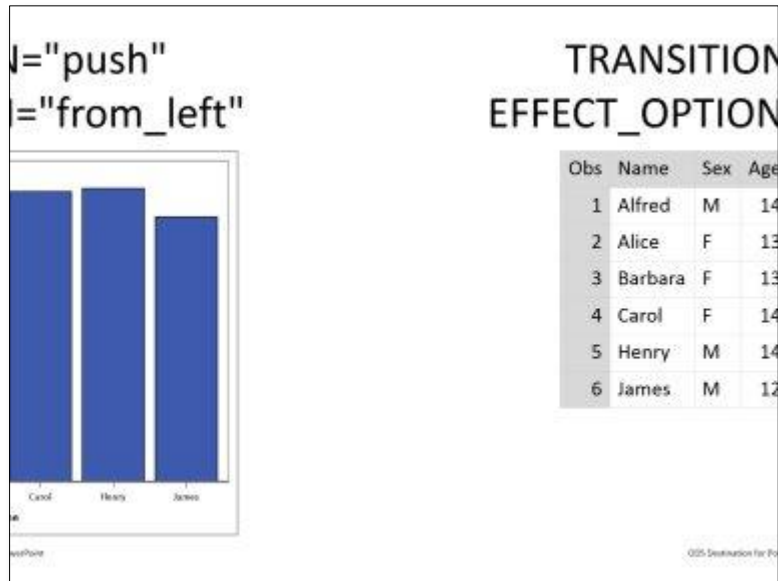
**Figure 2. Add a Company Logo to the Title Slide**

### **ADD A TRANSITION BETWEEN SLIDES**

A transition is an animation effect that occurs when you move from one slide to the next during an on-screen presentation. PowerPoint 2013 defines 47 transitions with names like “fade,” “reveal,” and “push.” Many transitions support a set of variations called “effect options.” For example, the push transition has four effect options that change the direction of the push: “from left,” “from right,” “from top”, and “from bottom.”

Transitions occur between a starting slide and an ending slide. The slide transition is associated with the ending slide. To add a transition between slides, add an ODS POWERPOINT statement immediately preceding the code that creates the ending slide. Add the OPTIONS option to the ODS POWERPOINT statement. Inside the OPTIONS option, specify the transition with TRANSITION suboption. If you want to specify an effect option, use the EFFECT\_OPTION suboption. Here’s an example of how to add the “push” transition with the “from left” effect option between two slides. Notice that the space in “from left” has been replaced by an underscore in the EFFECT\_OPTION value:

```
ods powerpoint file="Example_3.pptx" nogtitle nogfootnote;
title 'TRANSITION="push" EFFECT_OPTION="from_left"';
footnote 'ODS Destination for PowerPoint';
proc print data=sashelp.class (obs=6);
run;
ods powerpoint options(transition="push" effect_option="from_left");
proc sgplot data=sashelp.class (obs=6);
vbar name / response=height;
run;
ods powerpoint close;
```



**Figure 3. In the Middle of a Transition from the Slide on the Right to the Slide on the Left**

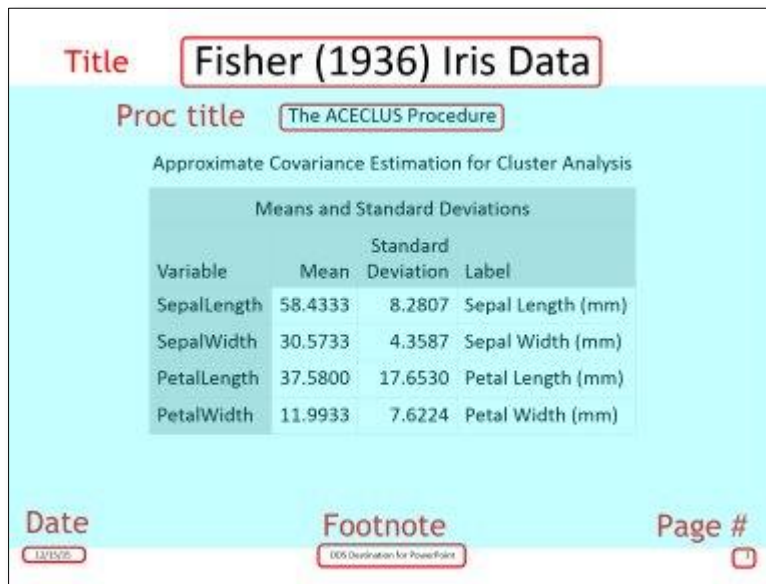
After you use the TRANSITION suboption in an ODS POWERPOINT statement, all the slides that you create will use that transition until you specify a different transition or specify TRANSITION="none".

The destination for PowerPoint supports all the transitions and effect options that PowerPoint 2013 defines. In addition, the destination for PowerPoint also defines suboptions that you can use to control the duration of the transition, associate a sound with a transition, and create presentations that advance automatically. The documentation for the TRANSITION suboption in the *SAS® 9.4 Output Delivery System: User's Guide, Fourth Edition* includes a list of all the transitions and effect options.

### HOW THE DESTINATION FOR POWERPOINT CREATES A SLIDE

The slides created by the destination for PowerPoint use large fonts, lots of white space, and landscape orientation. These attributes mean that vertical space is limited compared to other destinations. To make the most of the vertical space, it helps to understand how the destination for PowerPoint adds the slide elements to create a complete slide.

Here's a typical slide with the common elements highlighted:



**Figure 4. A Typical Slide**

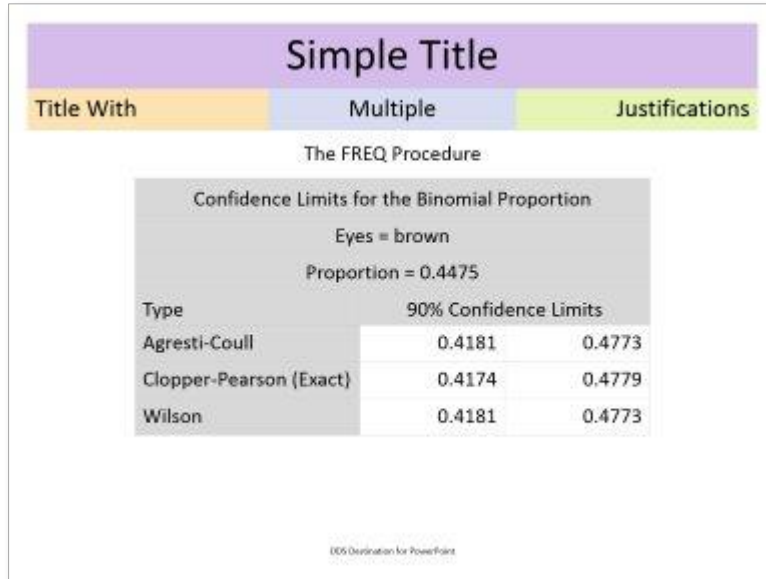
Slides created by the destination for PowerPoint are similar to pages in other ODS destinations. The titles are at the top, the footnotes are at the bottom and your content is in the middle. Unlike pages in other destinations the date and slide number appear at the bottom of the slide. This is consistent with the PowerPoint slide layouts supplied by Microsoft. You'll learn how to change the position of these elements later on in this paper. Like other ODS destinations you can eliminate the titles and footnotes by using a TITLE or FOOTNOTE statement with no arguments. You can suppress the date with the NODATE system option and the slide number with the NONUMBER system options.

When the destination for PowerPoint creates a slide, the first element added to the slide is the slide number, followed by the date, the titles, and finally the footnotes. When present, these elements consume a certain amount of vertical space on the slide. The remaining vertical space (highlighted in blue in Figure 4) is available for your content.

Given this information you can understand that the number of titles and footnotes should be kept to a minimum to allow the most vertical space for your tables and graphs. If the vertical space is too small to fit an entire table, the destination for PowerPoint will add as many rows and columns from the table as it can and put the remainder of the table (or as much as will fit) on the next slide. If the vertical space is too small to fit a graph, the destination for PowerPoint will move the graph to the next slide or shrink the graph to fit.

## TITLES AND FOOTNOTES

Text on a slide is usually contained in a PowerPoint *shape*, either a text box or a table. When changing the appearance of titles and footnotes it is important to know what shape is used to contain them because the shape constrains their appearance. Simple titles and footnotes with a single justification are rendered as text boxes. If there is more than one title or footnote, each one is in its own text box. By default, the title text box is the full width of the slide. The footnote text box is the middle 50% of the slide when the date and page number are present, or the full width of the slide when they are not present. Because each title or footnote is in a single text box, you can define only one background color for the entire title or footnote. If you use two or three JUSTIFY options in a TITLE or FOOTNOTE statement to create a title or footnote with more than one justification, then the title or footnote is rendered as a table with multiple columns. In this case you can define a separate background color for each column. In both cases, you can change the text color with the COLOR option on the TITLE or FOOTNOTE statement or by embedding an inline style function in the text.



**Figure 5. Simple Titles versus Titles with Multiple Justifications**

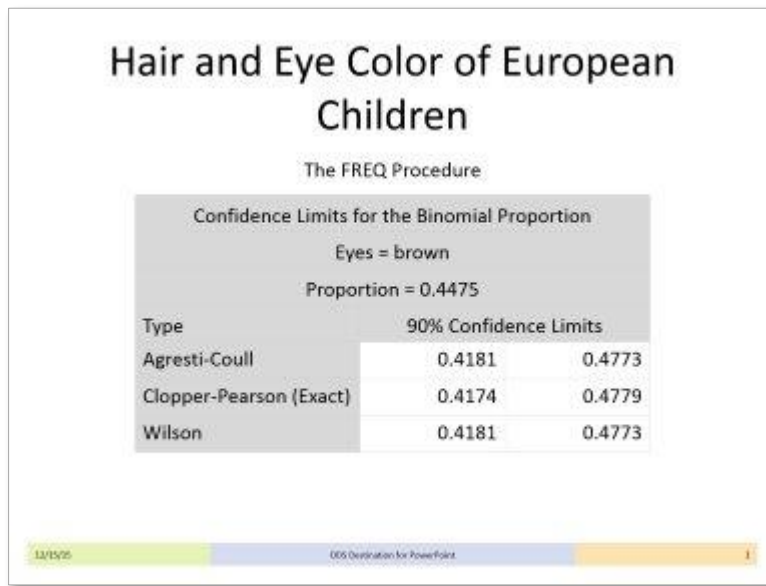
### SLIDE NUMBERS AND DATES

Slide numbers and dates are different from the other text on a slide. The slide number and date are rendered as a special text box whose content is automatically updated by PowerPoint each time the slide is displayed or printed. If you manually add a slide after creating a presentation the slides following the addition are renumbered to account for the new slide. Slides are always numbered starting with 1. This behavior cannot be changed. The destination for PowerPoint does not respect the PAGENO system option.

Similarly, the date will change to the current date each time you open or print the presentation. You can also use the Header and Footer dialog box in PowerPoint to select the date format for the presentation. You can override the automatic refreshing of the date by adding the SASDATE option to the ODS POWERPOINT statement. When the SASDATE option is used, the destination for PowerPoint adds the date and time you created the presentation to the slide as constant text.

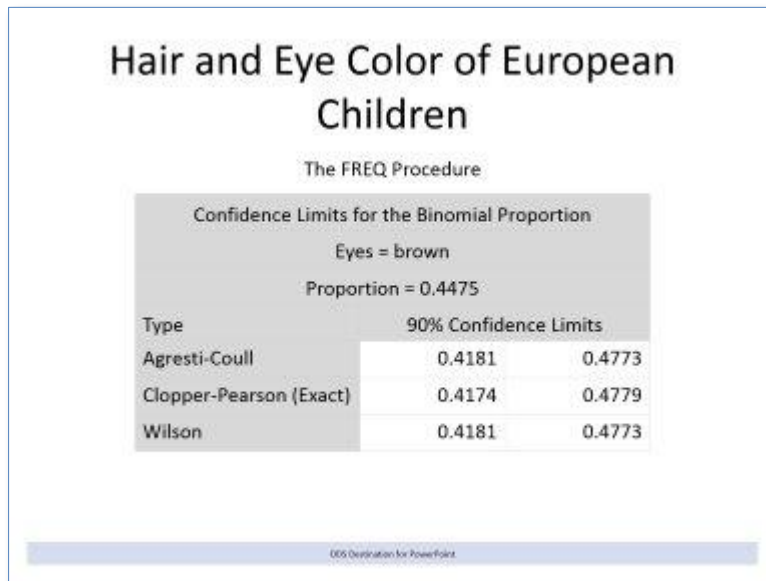
By default the text boxes that contain the slide number, footnote, and date are sized so that together they occupy the entire width of the slide. The slide number's text box occupies the right 25% of the width. The date's text box occupies the left 25% of the width. As mentioned earlier, if a footnote is present its text box occupies the middle 50% of the width. These proportions are important to keep in mind if you want to change the background color of one or more of these elements. For example, you can add a solid band of color to the bottom of your slides by using the same background color for the date, footnote, and slide number.

Figure 6 shows how the page width is shared between the three elements.



**Figure 6. The Slide Width Occupied by the Date, Footnote, and Slide Number**

When the slide number and date are not present, the text boxes that contain the footnotes occupy the full width of the slide, as shown in Figure 7.



**Figure 7. The Slide Width Occupied by the Footnotes When the Slide Number and Date Are Omitted**

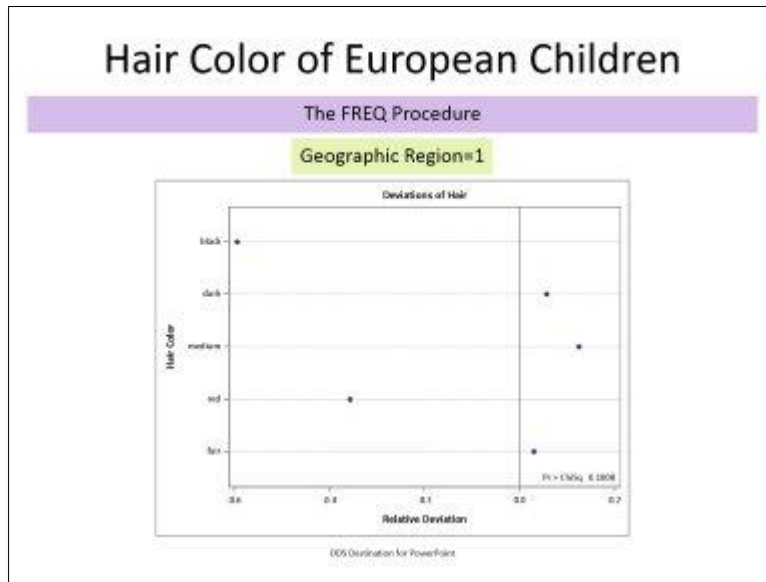
### YOUR CONTENT

After the titles, footnotes, date, and slide number have been added to the slide, the destination for PowerPoint adds your content. Your content can be one of four types of SAS output: tables, graphs, lists, and paragraphs. The destination for PowerPoint renders tables as PowerPoint tables, graphs as images, and lists and paragraphs as text boxes.

By default the destination for PowerPoint puts one table or one graph on each slide, and as many list items or paragraphs as will fit. Your content is horizontally centered on the slide. There is a 5 point (a

point is 1/72 of an inch or 0.035 centimeters) gap between the bottom of the last title and the top of the content.

Frequently, tables and graphs are accompanied by text such as the procedure title, a BY line, or table- or graph-specific notes. If present, those items are also separated from each other by 5 points. The text box that contains the procedure title occupies the full width of the slide. The text box that contains the BY line is only wide enough to contain the BY line.



**Figure 8. The Slide Width Occupied by the Procedure Title and BY Line**

While we're talking about spacing, text boxes and table cells have 5 points of padding on each side. You can change the ODS style to use a different amount of padding, but if you do so, keep in mind that PowerPoint has its own ideas about padding. If you try to use less than 5 points of padding PowerPoint might keep a minimum amount of padding around the text by wrapping the text in the cell.

It is good practice to ensure that your tables fit on one slide. However, as mentioned earlier if there is not enough space on the slide to contain all the rows and columns of a table, the destination for PowerPoint will print as much of the table as it can on the slide and then print the remainder of the table on subsequent slides. Too-big tables are rendered in row-major order. That is, the destination for PowerPoint determines how many rows can fit on a slide, and then renders all the columns for that many rows. Then it renders the next set of rows, repeating until all the rows in the table have been rendered. If the table is too wide to fit on a slide, the destination for PowerPoint determines how many columns can fit on a slide and creates as many slides as necessary to hold all the columns for the current set of rows.

Graphs must always fit on one slide. If a graph is too big to fit, and there is already other content (such as a paragraph created by PROC ODSTEXT) on the slide, then the destination for PowerPoint will start a new slide for the graph. If the graph is too big to fit and it is the only thing on the slide, the destination for PowerPoint will shrink the graph up to about 85% of its original size in order to make it fit. If the graph still does not fit, the destination for PowerPoint will render the slide anyway. The graph will overflow the slide boundaries. In that case, you can change the graph's size to make it fit.

## PANIC MODE

As you might imagine, there can be times when certain page elements are simply too large to fit on a slide. If this happens, the destination for PowerPoint will discard the page elements that don't fit. If there are too many titles to fit on a page, the destination for PowerPoint discards the titles. Similarly, if there are too many footnotes to fit in the space left after adding the titles, the destination for PowerPoint discards the footnotes. You will see warnings in the log when this happens.



In extreme cases the amount of unused space on a slide can be too small to fit even part of a table. In these cases the destination for PowerPoint enters “panic mode.” In panic mode, the destination for PowerPoint makes no attempt to produce an attractive or even usable slide. Instead, it discards page elements as necessary to get at least one table cell on each slide. You can change your program so that the table will fit.

In panic mode, the destination for PowerPoint discards page elements in the following order:

1. the procedure title.
2. the BY line.
3. the table headers and footers.

You will see warnings in the log when the destination for PowerPoint goes into panic mode. If, after discarding these elements, there is still not enough room to fit one row, the destination for PowerPoint renders one row (or part of one row) per slide anyway. The table row will overflow the slide boundaries.

It is not possible for the destination for PowerPoint to render part of an individual table cell on one slide and the remainder of the cell on the next slide. If a single table cell is too big to fit on a slide, the destination for PowerPoint renders the entire cell on the slide anyway.

## THE POWERPOINT STYLE TEMPLATES

For this part of the paper it helps to understand how style templates control the appearance of your output. For in-depth information about style templates, see *SAS® 9.4 Output Delivery System: Procedures Guide, Second Edition*, or *PROC TEMPLATE Made Easy: A Guide for SAS® Users*, by Kevin Smith. The PROC TEMPLATE code that defines the PowerPointLight and PowerPointDark style templates is in Appendix 2: The PowerPoint Style Templates.

The visual appearance of the output of SAS procedures is controlled by a style template created by the TEMPLATE procedure. There are two standard ODS style templates for the destination for PowerPoint: PowerPointLight and PowerPointDark. These style templates are specifically designed to be used when creating PowerPoint documents. You should not use other ODS style templates as they will not produce attractive slides. The PowerPoint style templates are designed to make your content the focus of your presentation, so the slides they create are monochromatic and use only one font.

You can also create your own style template to use with the destination for PowerPoint. Your style template must use either PowerPointLight or PowerPointDark as its parent template. The reason for this restriction is that PowerPointLight and PowerPointDark use the SCHEME statement in the template definition. The SCHEME statement is only used in PowerPoint style templates, and all PowerPoint style templates must have a SCHEME statement. The SCHEME statement allows the destination for PowerPoint to create a PowerPoint theme for the presentation.

Every PowerPoint presentation is based on a theme. A theme is a predefined combination of colors, fonts, and effects. When you create a presentation using PowerPoint you usually start by choosing a theme. The theme might be one of the themes that are supplied by Microsoft as part of PowerPoint, a theme that was created for use at your business, or a theme you create yourself. When you create a presentation using the destination for PowerPoint the theme is created from the ODS style template. Therefore, style templates used with the destination for PowerPoint must define the style elements required by a PowerPoint theme. These style elements are defined by the SCHEME statement.

The theme style elements include two fonts and twelve colors. (PowerPoint themes also define effects for which there are no ODS style template equivalents and therefore cannot be defined in an ODS style template.) The two fonts are called “heading font,” which is used for titles and footnotes, and “body font,” which is used for the other text in your slides. The colors include two pairs of light and dark colors named “light1,” “dark1,” “light2,” and “dark2.” By convention light1 and dark1 are always white and black. There are six accent colors named “accent1” through “accent6.” The accent colors are selected so that they can be seen over the light and dark colors. Finally, the theme defines a color for hyperlinks named “hyperlink” and a second color for followed (that is, clicked) hyperlinks named “followed\_hyperlink.”

Inside the PowerPoint document, theme fonts and colors are referenced by their names instead of their values. This is very similar to the way macro symbols are used in SAS programs. When you change the value assigned to a font or color name in the theme, the font or color changes throughout the presentation. When you create a presentation using PowerPoint, you change its theme style elements using the controls in the Design tab. When you create a presentation using the destination for PowerPoint, you change the theme style elements using the SCHEME statement.

## THE SCHEME STATEMENT

In the ODS style templates for PowerPoint, the SCHEME statement is responsible for defining values for the theme style elements. Here's the SCHEME statement in the PowerPointLight style template:

```
scheme 'ODS Light' /
    followed_hyperlink = cx7A40A6
    hyperlink = cx00709f
    accent6 = cx5E6C10
    accent5 = cx6590CD
    accent4 = cxF8B93C
    accent3 = cx9D2E14
    accent2 = cx90B328
    accent1 = cx3D5AAE
    light2 = cxd8d8d8
    dark2 = cx646464
    light1 = cffffff
    dark1 = cx000000
    body_font = ("Calibri, <MTsans-serif>, <sans-serif>", 20pt)
    heading_font = ("Calibri, <MTsans-serif>, <sans-serif>", 42pt);
```

The name of the theme is "ODS Light." A theme name can be just about any quoted string. Following the name is a series of assignments of values to color and font symbolic names. Color values can be any one of the color names supported by SAS/GRAPH. Font values use the font definition syntax used by the FONT style attribute.

To use a different theme font or color in your presentation, change the value assigned to the font or color symbolic name:

1. use PROC TEMPLATE to define a new style template.
2. specify either PowerPointLight or PowerPointDark as the parent of your template.
3. add a SCHEME statement to your template definition. In the SCHEME statement assign the new value to the symbolic name of the theme style element you're changing. You do not have to specify any theme style elements that you are not changing.
4. specify the name of your new style template in the STYLE option of your ODS POWERPOINT statement.

This example creates a new style template called SegoeLight. This style template replaces the default Calibri font with the Segoe UI Light font. Then, the example creates a "before" slide using PowerPointLight and an "after" slide using SegoeLight:

```
proc template;
    define style styles.segoelight;
        parent=styles.powerpointlight;
        scheme "Segoe Light" /
            heading_font = ("Segoe UI Light, <MTsans-serif>, <sans-serif>",42pt)
            body_font = ("Segoe UI Light, <MTsans-serif>, <sans-serif>",20pt);
    end;
run;

title 'Patterns of Diabetes';
footnote 'ODS Destination for PowerPoint';
```

```

/* Use PowerPointLight */
ods select InputSummary;
ods powerpoint file="Figure_9a.pptx";
proc gam data=diabetes;
    model logCP = spline(Age) spline(BaseDeficit);
run;
ods powerpoint close;

/* Use SegoeLight */
ods select InputSummary;
ods powerpoint file="Figure_9b.pptx" style=segoelight;
proc gam data=diabetes;
    model logCP = spline(Age) spline(BaseDeficit);
run;
ods powerpoint close;

```

Here's the result:

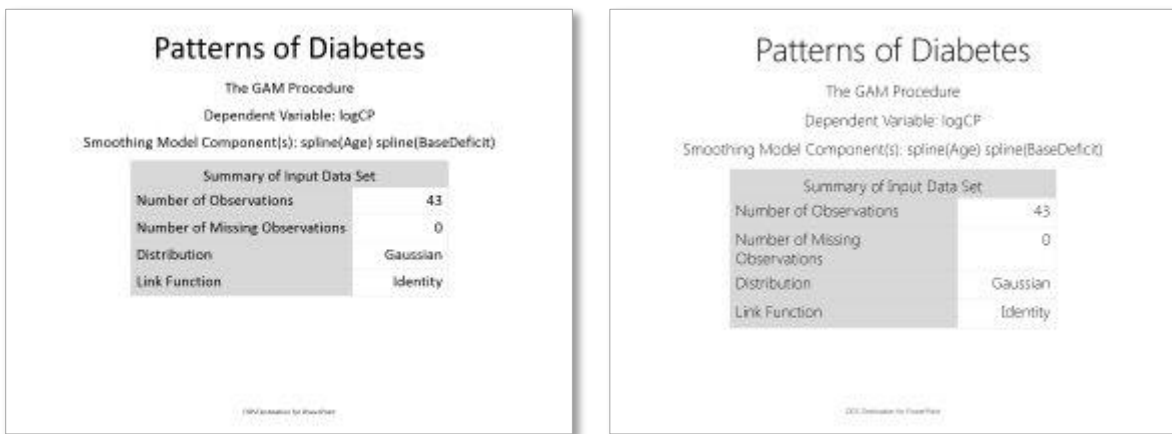


Figure 9. The Effect of Changing the SCHEME Statement

## THE LIGHT\_ON\_DARK STATEMENT

There is one other PROC TEMPLATE statement that is unique to the PowerPoint style templates. The LIGHT\_ON\_DARK statement should be used in any custom style template that has a dark colored background and light colored fonts. If the parent of your custom style template is PowerPointDark, then the LIGHT\_ON\_DARK statement is already in effect and you do not need to add it again. Otherwise, you can add it immediately following the DEFINE STYLE statement.

## POWERPOINT TEMPLATE STYLE ELEMENTS

In addition to the SCHEME statement, the style templates for the destination for PowerPoint are largely made up of a series of CLASS statements. (See the PowerPointLight and PowerPointDark style template definition in the appendix to this paper.) Each CLASS statement defines a set of style attribute specifications. Classes are usually associated with a specific type of text or a specific graph element. For example, the BodyDate class is associated with the date text box.

Just as style templates can have parent templates, classes can be the parent of other classes. Parent classes define groups of style attributes that are inherited by their child classes. Because the parent style template of the PowerPoint style templates is Base.Template.Style, most of the parent-child relationships are defined in that template. For example, in Base.Template.Style the TitlesAndFooters class is defined to be the parent class of the SystemTitle and SystemFooter classes. TitlesAndFooters defines style attributes that are inherited by both SystemTitle and SystemFooter.

To help understand the classes in PowerPointLight and PowerPointDark, let's look at three examples.

## CHANGE THE BACKGROUND COLOR OF TITLES AND FOOTNOTES

As you might guess, the SystemTitle and SystemFooter classes specify the style attributes for titles and footnotes. Both of these classes have the TitlesAndFooters class as their parent. Here's the definition of these three classes in the PowerPointLight style template:

```
class TitlesAndFooters /
  backgroundcolor = _undef_
;

class SystemTitle /
  font = heading_font
;

class SystemFooter /
  font = heading_font
  fontweight = medium
  fontsize = 9pt
;
```

In the TitlesAndFooters class the backgroundcolor attribute value is `_undef_`. This value “undefines” or removes any background color that can be inherited from a parent class. The effect of removing the background color is to make the background of title and footer text boxes transparent so that the slide background appears behind them. The SystemTitle class simply assigns the theme symbolic name “heading\_font” as the title font. Given the preceding discussion about the SCHEME statement it should be obvious why. The size of “heading\_font” is defined in the SCHEME statement as 42 points. The SystemFooter class also assigns “heading\_font” to the font attribute, and then modifies the font definition to use a medium font weight and a font size of 9 points.

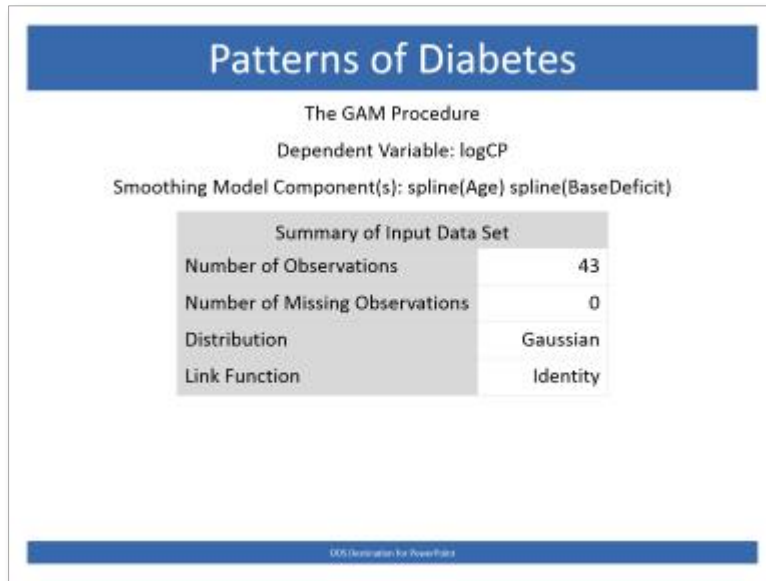
Suppose instead you wanted your titles and footnotes to have a dark blue background and white text. You can accomplish this by simply creating a new style template with a new definition of the SystemTitle and SystemFooter classes and then specifying your new style template in the STYLE option:

```
proc template;
  define style styles.mystyle;
    parent=styles.powerpointlight;
    class SystemTitle, SystemFooter /
      background = cx3869AD
      color = white;
    end;
run;

title 'Patterns of Diabetes';
footnote 'ODS Destination for PowerPoint';

ods select InputSummary;
ods powerpoint file="Figure_10.pptx" style=styles.mystyle;
proc gam data=diabetes;
  model logCP = spline(Age) spline(BaseDeficit);
run;
ods powerpoint close;
```

Because your custom style template has PowerPointLight as its parent, all other style attributes remain the same. Figure 10 shows the slide that this example creates:



**Figure 10. Change the Background Color of Titles and Footnotes**

### MOVE THE SLIDE NUMBER AND DATE TO THE TOP OF THE SLIDE

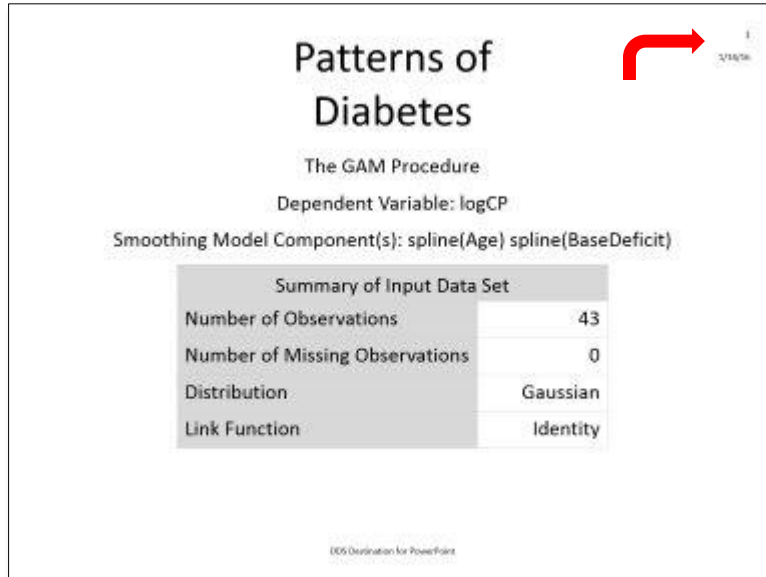
The style attributes for the date and slide number text fields are in the BodyDate and PageNo classes, respectively. Here's how these classes are defined in the PowerPoint style templates:

```
class BodyDate /
  fontweight = medium
  fontsize = 9pt
  just = left
  vjust = bottom
  width = 25%
  ;

class PageNo /
  fontsize = 9pt
  just = right
  vjust = bottom
  width = 25%
  ;
```

Here you can see that the position of each of these elements is described by the just and vjust style attributes. The BodyDate is in the bottom left. The PageNo is in the bottom right. Suppose you want to move these elements to the top right where they usually appear in other ODS destinations. As in the previous example, create a new style template. In it, modify the vjust and just attributes in the BodyDate and PageNo classes. The other style attributes remain the same:

```
proc template;
  define style styles.mystyle;
    parent=styles.powerpointlight;
    class BodyDate, PageNo /
      vjust = top
      just = right
    ;
  end;
  ...
ods powerpoint file="Figure_11.pptx" style=styles.mystyle;
```



**Figure 11. Move the Slide Number and Date**

There are two consequences of this change that might be unexpected. First, notice that the date appears *under* the slide number. This is because both classes specify that their text box appears in the top right but they can't both be in the same position. As you learned earlier, when the destination for PowerPoint renders a slide, it puts the slide number on the slide first, followed by the date. It puts the slide number in the top right corner, and then puts the date as close to the top right as it can, just below the date. The second thing to notice is that the title is now wrapped into two lines. Because the title has to share the slide width with the date and slide number, the title text box is now limited to 50% of the slide width.

### CHANGE THE BACKGROUND COLOR OF TABLE HEADERS

This last example shows how to change the color of the table header and row headers. The style attributes of these cells are in the Header and RowHeader classes. In turn, these classes inherit from the HeadersAndFooters class. Here's how these three classes are defined in the PowerPointLight style template:

```
class HeadersAndFooters /
  fontweight = medium
  backgroundcolor = light2
  bordercolor = cxCCCCCC
  borderwidth = 0.5pt
  borderstyle = solid
;

class Header /
;

class RowHeader /
;
```

Obviously the Header and RowHeader classes are just place-holders. They define no style attributes of their own. All their style attributes are inherited from HeadersAndFooters. Actually HeadersAndFooters is the parent of many other classes such as Footer and Row Footer. To limit this change to just the table headers and row headers, change the backgroundcolor style attribute in the Header and RowHeader classes. Here's the new style template:

```

proc template;
  define style styles.mystyle;
    parent=styles.powerpointlight;
    class Header, RowHeader /
      backgroundcolor = cxAEBBE2;
    ;
  end;
  ...
ods powerpoint file="Figure_12.pptx" style=styles.mystyle;

```

**Patterns of Diabetes**

The GAM Procedure  
Dependent Variable: logCP  
Smoothing Model Component(s): spline(Age) spline(BaseDeficit)

Summary of Input Data Set	
Number of Observations	43
Number of Missing Observations	0
Distribution	Gaussian
Link Function	Identity

ODS Destination for PowerPoint

**Figure 12. Change the Background Color of Table Headers**

## GRADIENTS

In addition to using a solid color or an image as the slide background you can use a gradient. A gradient is an image that changes color smoothly along a line or the radius of a circle. A well-chosen gradient background provides subtle shading, adding visual interest to your slides without being distracting. The destination for PowerPoint is the only ODS destination that supports gradients. Because gradients are images you can use a gradient definition in any class that supports the `BACKGROUNDIMAGE` style attribute. The destination for PowerPoint supports two types of gradients, *linear gradients* and *radial gradients*. Gradients are dynamically generated by PowerPoint from a gradient definition. Gradient definitions use a terse syntax that is modeled on the W3C working draft for CSS image values.

### LINEAR GRADIENTS

In a linear gradient, the colors change from one color to the next along lines perpendicular to the *gradient line*. The gradient line gives direction to the gradient. The definition of a linear gradient specifies the gradient line and at least two color-stops. A linear gradient definition starts with the string "linear-gradient" and looks like this:

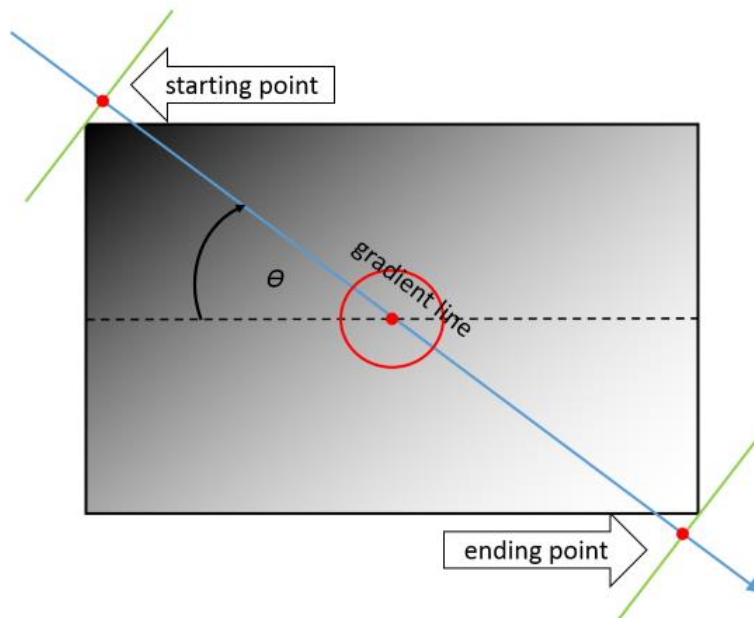
```
linear-gradient(<gradient line>, <color-stop>[, <color-stop>]+)
```

The first argument specifies the gradient line. You can specify the gradient line in two different ways. The first way is to specify an angle. An angle is a number followed by "deg". For example:

```
linear-gradient(45deg, ...)
```

Angles are measured clockwise. The angle 0deg points to the left and 90deg points up. When you specify the gradient line by an angle, the gradient line passes through the center of the gradient at the angle

specified. The ending point of the gradient line is the point on the gradient line where a line drawn perpendicular to the gradient line would intersect the corner of the gradient box in that direction. The starting point is determined identically, except in the opposite direction. See Figure 13.



**Figure 13. A Gradient Line Specified by an Angle**

The other way to specify the gradient line is by naming a side or corner. The side and corner names are “top”, “right”, “bottom”, “left”, “top right”, “top left”, “bottom right”, and “bottom left” (without the quotation marks). The gradient line begins at the named side or corner and extends to the opposite side or corner. For example, if the starting point is “top” then the gradient line extends to the bottom. Here’s an example of a linear gradient that starts in the top left corner and extends to the bottom right corner:

```
linear-gradient(top left, ...)
```

## RADIAL GRADIENTS

A radial gradient is defined by a *gradient ray* that starts in the center of the gradient and extends outward. Color-stops are positioned along the gradient ray starting in the center of the gradient toward the right. At each color-stop, the circle surrounding the center of the gradient has the specified color. Between color-stops the color is linearly interpolated between the colors of the adjacent color-stops. A radial gradient definition starts with the string “radial-gradient” and looks like this:

```
radial-gradient(<position>, <color-stop>[, <color-stop>]+)
```

*Position* defines the center of the gradient. There are two ways to specify the position. The first way is to specify “center” or one of the corners “top left”, “top right”, “bottom left”, or “bottom right”. The second way is to specify the position of the center as a pair of percentages. The first percentage is the distance from the top of the slide. The second percentage is the distance from the left of the slide. That is, “0% 0%” is the top left corner of the slide and “100% 100%” is the bottom right corner. “50% 50%” is the center. You can specify any positive integer percentage between 0% and 100%, but a pair of percentages that does not exactly describe the center or a corner will be adjusted to the nearest corner or to the center. Here are some examples of positions:



```
radial-gradient(top left, ...)
radial-gradient(center, ...)
radial-gradient(bottom right, ...)
radial-gradient(0% 0%, ...)
radial-gradient(50% 50%, ...)
radial-gradient(100% 100%, ...)
```

Radial gradients are always circular and sized to meet the corner of the slide farthest from the center of the gradient.

## COLOR-STOPS

Both linear gradient definitions and radial gradient definitions describe how the colors change by a series of at least two color-stops. A color-stop is a color name and optionally a position on the gradient line or gradient ray. Color-stops are arranged along the gradient line or gradient ray from the starting point to the ending point.

The color name can be an RGB value in the hexadecimal form #RRGGBB, or by an RGB value in functional form. If you use the hexadecimal form, the red, green, and blue color channels are specified by hexadecimal values between 00 and FF. For example, the color yellow is specified as #FFFF00. The functional form is specified as rgba(rr,gg,bb,aa), where the red, green, and blue channels can be specified either as decimal values between 0 and 255 or percentages between 0% and 100%. The fourth channel, called the alpha channel, can be specified either as a percentage between 0% and 100% or a decimal fraction between 0.0 and 1.0. Low values are more transparent, high values are more opaque. For example, rgba(100%,0%,0%,100%) is opaque red. The value rgba(100%,100%,0%,0.1) is very transparent yellow. A color name can also be one of the predefined SAS/GRAPH® colors such as red or blue, or a SAS/GRAPH® Color Naming System (CSN) color such as palevioletred. CSN colors must be specified without blank or underscore separators.

A color name can also be specified in a few other, less-frequently-used, formats. See Appendix 1: Other Color Value Specifications.

The position part of a color-stop is optional. If present, it is a percentage of the distance from the starting point on the gradient line or gradient ray to the ending point. The starting point is 0%. The ending point is 100%. Color-stops must always be specified in order from the starting point to the ending point. If the first color-stop does not have a position it is assigned 0%. If the last color-stop does not have a position it is assigned 100%. If a color-stop has a position that is less than the position of a color-stop before it, it is assigned a position equal to the largest position of any color-stop before it. If any color-stop still does not have a position, then each run of adjacent color-stops without positions is evenly spaced between the preceding and following color-stops with positions. Here are some examples of color-stops:

```
red
lightblue
blue 25%
#dde6cf
#d49e6c 30%
rgba(85%,84%,85%,50%) 50%
```

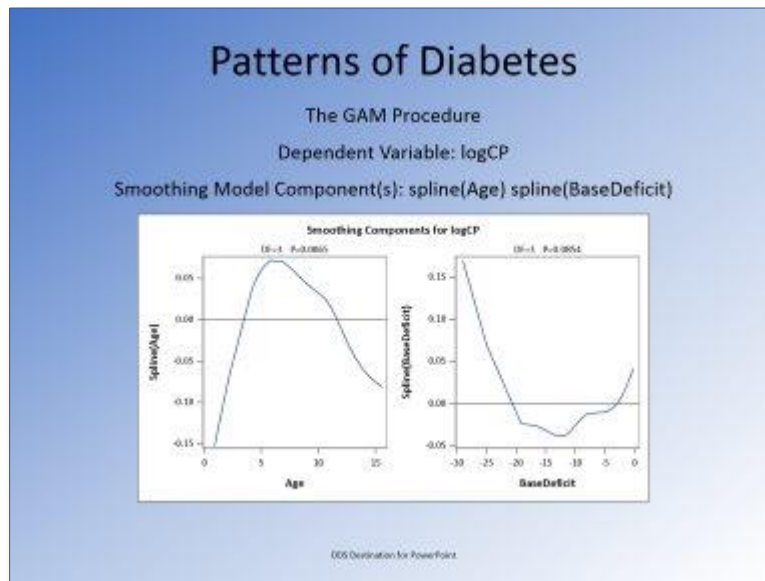
## EXAMPLES OF GRADIENTS

Here are three examples of using a gradient for the background in a presentation. Remember, a gradient is an image, so you can specify a gradient not just as a slide background, but anywhere you can specify an image.

The first example creates a linear gradient background. The gradient-line is specified by the corner "top left". Therefore, the starting point of the gradient-line is the top left corner and the ending point is the bottom right corner. The first color-stop is #4472C4. The definition does not specify a position for this color-stop. Because it's the first color-stop, the position defaults to 0%. The second color-stop is #B4C7E7. Again, the definition does not specify a position, so the position is 50%, half-way between the

first and third color-stop. The last color stop is white. Because it's the last color-stop the position defaults to 100%. Here's the code:

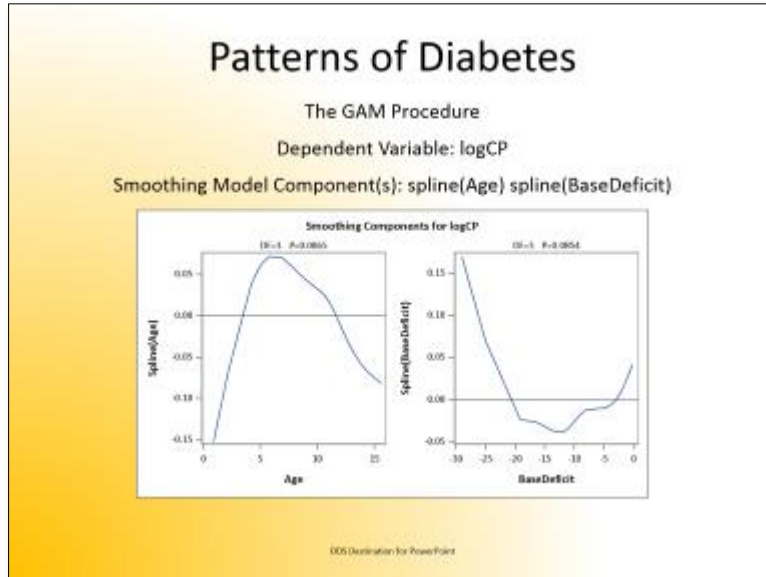
```
proc template;
  define style styles.mystyle;
    parent=styles.powerpointlight;
    class Body /
      backgroundimage =
        "linear-gradient(top left, #4472C4, #B4C7E7, white)";
    ;
  end;
  ...
ods powerpoint file="Figure_14.pptx" style=styles.mystyle;
```



**Figure 14. Using a Linear Gradient as the Slide Background**

The next example creates a radial gradient background. The center of the gradient and the starting point of the gradient ray is the bottom left corner of the slide. The ending point is the top right corner. Again, the definition does not specify a position for the first and last color-stops, so they default to 0% and 100% respectively. The middle color-stop is not half-way between the first and last color-stops, so the color-stop position must be explicitly specified as 62%:

```
proc template;
  define style styles.mystyle;
    parent=styles.powerpointlight;
    class Body /
      backgroundimage = "radial-gradient(bottom left,
        rgba(255, 192, 0, 1.0), white 62%, white)";
    ;
  end;
  ...
ods powerpoint file="Figure_15.pptx" style=styles.mystyle;
```



**Figure 15. Using a Radial Gradient as the Slide Background**

The last example creates a four color linear gradient background. Instead of specifying the gradient definition in the style template as in the previous examples, in this example it is the value of the BACKGROUNDIMAGE suboption in the OPTIONS option. The gradient-line goes from the top of the slide to the bottom. All the color names are in #RRGGBB format. The positions of the first and last color names default to 0% and 100%, respectively, while the second color-name is at 25% and the third at 75%. Here's the code:

```

title 'Patterns of Diabetes';
footnote 'ODS Destination for PowerPoint';

ods powerpoint file="Figure_16.pptx"
  options (backgroundimage="linear-gradient (top,
    #03D4A8, #21D6E0 25%, #0087E6 75%, #005CBF)");
ods select SmoothingComponentPlot;
proc gam data=diabetes;
  model logCP = spline(Age) spline(BaseDeficit);
run;
ods powerpoint close;

```

Here's how the slide looks:

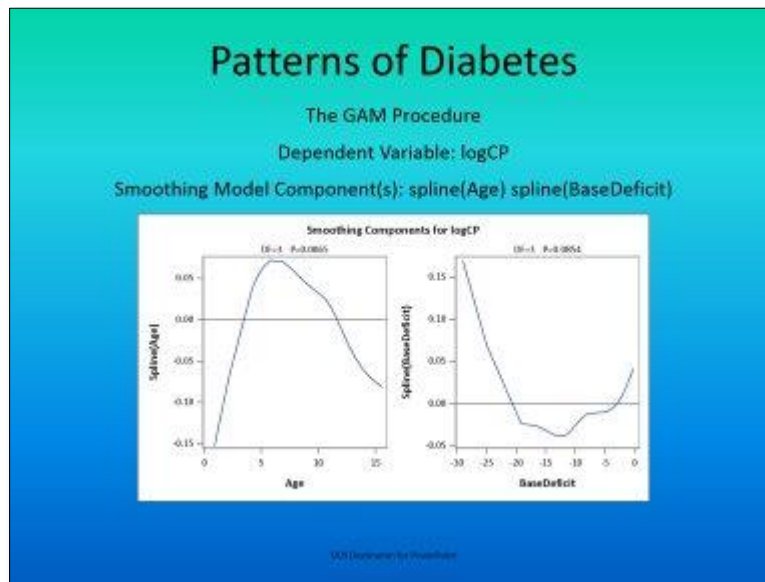


Figure 16. Using a Four-color Linear Gradient as the Slide Background

## CONCLUSION

The PowerPoint styles that SAS supplies, PowerPointLight and PowerPointDark, are very plain, monochromatic, and best suited for presentations where you want your SAS output to be the focus. They're also good starting points for your own customized PowerPoint slides. Understanding how the destination for PowerPoint renders a slide gives you insight into how changes to the style template can affect your presentation. This paper also explained many techniques for customizing your presentations, including how to add color, suppress or reposition slide elements, change slide backgrounds within the same presentation, add slide transitions, and create gradient backgrounds.

## REFERENCES

Hunter, Tim, 2013. "A First Look at the ODS Destination for PowerPoint." *Proceedings of the SAS® Global Forum 2013 Conference*, Cary, NC. SAS Institute Inc. Available at <https://support.sas.com/resources/papers/proceedings13/041-2013.pdf>.

"CSS Image Values and Replaced Content Module Level 3." W3C, 2011. Available at <http://www.w3.org/TR/2011/WD-css3-images-20110217/>.

"CSS3 Values and Units." W3C, 2006. Available at <http://www.w3.org/TR/2006/WD-css3-values-20060919/>.

"CSS Color Module Level 3." W3C, 2011. Available at <http://www.w3.org/TR/2011/REC-css3-color-20110607/>.

## ACKNOWLEDGMENTS

The ODS destination for PowerPoint could not have been created without the contributions of Wayne Hester, Senior Software Developer, and Nancy Goodling, Principal Development Tester.

## RECOMMENDED READING

- SAS® 9.4 Output Delivery System: User's Guide
- SAS® 9.4 Output Delivery System: Procedures Guide

- *SAS/GRAPH® 9.4: Reference, Fourth Edition*

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Tim Hunter  
100 SAS Campus Drive  
Cary, NC 27513  
SAS Institute Inc.  
[tim.hunter@sas.com](mailto:tim.hunter@sas.com)  
<http://www.sas.com>

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.

## APPENDIX 1: OTHER COLOR VALUE SPECIFICATIONS

In addition to the #RRGGBB and rgba() formats, a color value in a color-stop can be specified in one of the following formats:

- **#RGB**  
Three-digit RGB notation. A color value in the form #RGB is converted to #RRGGBB by duplicating each digit. For example, #A0B expands to #AA00BB.
- **hsl(hue, saturation, lightness)**  
Hue is an angle of the color circle between 0° and 360°. Hue starts with the primary color red at 0°, green at 120° and blue at 240°. Saturation and lightness are represented as percentages between 0% and 100%.
- **hsla(hue, saturation, lightness, alpha)**  
The same as hsl(), but with an alpha channel. As in the rgba() format, alpha can be either a percentage between 0% and 100% or a decimal fraction between 0.0 and 1.0.

These formats are available in PROC TEMPLATE only. They are not supported when a gradient is supplied by the BACKGROUNDIMAGE suboption of the OPTIONS statement.

## APPENDIX 2: THE POWERPOINT STYLE TEMPLATES

To view the current definition of the PowerPointLight and PowerPointDark style templates in the SAS windowing environment (also known as DMS), follow these steps:

1. In the Results window, select the Results folder. Right-click and select Templates to open the Templates window.
2. Double-click **Sashelp.Tmplmst** to view the contents of that directory.
3. Double-click **Styles** to view the contents of that directory.
4. Double-click either **PowerPointLight** or **PowerPointDark**.

The PowerPointLight and PowerPointDark style templates are largely identical except that the colors white and black are exchanged. In SAS 9.4M3 these style templates are created by the following macro:

```

%macro PowerPointStyle(name, foreground, background, hyperlink,
                      folHlink, secondary, hdrCellBorder);

%local gdata1 gdata2 gdata3 gdata4 gdata5 gdata6 bodyFontSize;
%let gdata1 = cx3D5AAE;
%let gdata2 = cx90B328;
%let gdata3 = cx9D2E14;
%let gdata4 = cxF8B93C;
%let gdata5 = cx6590CD;
%let gdata6 = cx5E6C10;
%let bodyFontSize = 20pt;

proc template;

define style styles.PowerPoint&name;
  %if &name = Dark %then %do;
    light_on_dark;
  %end;

  scheme "ODS &name" /
    heading_font = ("Calibri, <MTsans-serif>, <sans-serif>", 42pt)
    body_font = ("Calibri, <MTsans-serif>, <sans-serif>", &bodyFontSize)
    dark1 = &foreground
    light1 = &background
    dark2 = cx646464
    light2 = cxd8d8d8 /* light2 = cxE6E6E6 */
    accent1 = &gdata1
    accent2 = &gdata2
    accent3 = &gdata3
    accent4 = &gdata4
    accent5 = &gdata5
    accent6 = &gdata6
    hyperlink = &hyperlink
    followed_hyperlink = &folHlink
  ;

  class fonts /
    "BatchFixedFont" = ("<monospace>, <MTmonospace>, monospace", 8pt)
    "FixedFont" = ("<monospace>, <MTmonospace>, monospace",
&bodyFontSize)
    "FixedEmphasisFont" = ("<monospace>, <MTmonospace>, monospace",
&bodyFontSize, italic)
    "FixedStrongFont" = ("<monospace>, <MTmonospace>, monospace",
&bodyFontSize, bold)
  ;

  class text "Common text." /
    "Note Banner" = "Note:"
    "Warn Banner" = "Warning:"
    "Error Banner" = "Error:"
    "Fatal Banner" = "Fatal:"
  ;

  class GraphFonts / /* From Styles.Harvest */
    "GraphAnnoFont" = ("Calibri, <sans-serif>, <MTsans-serif>", 10pt)
    "GraphLabelFont" = ("Calibri, <sans-serif>, <MTsans-serif>", 10pt,
bold)

```

```

"GraphLabel2Font" = ("Calibri, <sans-serif>, <MTsans-serif>", 10pt)
"GraphValueFont" = ("Calibri, <sans-serif>, <MTsans-serif>", 9pt)
"GraphTitle1Font" = ("Calibri, <sans-serif>, <MTsans-serif>", 14pt,
    bold)
"GraphTitleFont" = ("Calibri, <sans-serif>, <MTsans-serif>", 11pt,
    bold)
"GraphFootnoteFont" = ("Calibri, <sans-serif>, <MTsans-serif>", 10pt,
    bold)
"GraphUnicodeFont" = ("<MTsans-serif-unicode>", 9pt)
"GraphDataFont" = ("Calibri, <sans-serif>, <MTsans-serif>", 7pt)
"NodeTitleFont" = ("Calibri, <MTsans-serif>", 9pt)
"NodeLabelFont" = ("Calibri, <MTsans-serif>", 9pt)
"NodeInputLabelFont" = ("Calibri, <MTsans-serif>", 9pt)
"NodeDetailFont" = ("Calibri, <MTsans-serif>", 7pt)
;

%GraphStyleCommon(&name, &gdata1, &gdata2, &gdata3, &gdata4, &gdata5,
    &gdata6);

class LayoutContainer /
    rules = none
    frame = void
    borderwidth = 0
    cellspacing = 36pt
    padding = 0
;

class LayoutRegion /
    padding = 0
;

class Container /
    background = light1
    color = dark1
    font = body_font
    padding = 5pt
;

class Output /
    padding = 0
;

class Document /
    linkcolor = hyperlink
    activelinkcolor = hyperlink
    visitedlinkcolor = followed_hyperlink
    protectspecialchars = auto
;

class Body /
;

class SysTitleAndFooterContainer /
    cellspacing = 0
    font = heading_font
;

```

```

class TitlesAndFooters /
  background = _undef_
  ;

class SystemTitle /
  font = heading_font
  ;

class SystemTitle2 /
  fontsize = 24pt;
  ;

class SystemTitle3 /
  fontsize = 20pt;
  ;

class SystemFooter /
  font = heading_font
  fontweight = medium
  fontsize = 9pt
  ;

style PresentationTitle from SystemTitle
  "First level title for title slide" /
  just = c
  fontsize = 44pt
  ;

style PresentationTitle2 from PresentationTitle
  "Second level title for title slide" /
  fontsize = 32pt
  color = cx898989
  ;

class Date /
  background = _undef_
  ;

class BodyDate /
  fontweight = medium
  fontsize = 9pt
  just = left
  vjust = bottom
  width = 25%
  ;

class SASDate from BodyDate /
  fontsize = 7pt
  ;

class PageNo /
  fontsize = 9pt
  just = right
  vjust = bottom
  width = 25%
  ;

```



```

class ProcTitle /
    font = body_font
    ;

class ProcTitleFixed /
    font = fonts("FixedFont")
    ;

class BylineContainer /
    cellspacing = 0
    ;

class Byline /
    font = body_font
    ;

class HeadersAndFooters /
    fontweight = medium
    background = &secondary
    bordercolor = &hdrCellBorder
    borderwidth = 0.5pt
    borderstyle = solid
    ;

class Header /
    ;

class RowHeader /
    ;

class Footer /
    ;

class RowFooter /
    ;

class HeaderFixed, RowHeaderFixed, FooterFixed, RowFooterFixed /
    font = fonts("FixedFont")
    ;

class HeaderEmphasis, RowHeaderEmphasis, FooterEmphasis,
    RowFooterEmphasis /
    fontstyle = italic
    ;

class HeaderEmphasisFixed, RowHeaderEmphasisFixed, FooterEmphasisFixed,
    RowFooterEmphasisFixed /
    font = fonts("FixedEmphasisFont")
    ;

class HeaderStrong, RowHeaderStrong, FooterStrong, RowFooterStrong /
    fontweight = bold
    ;

class HeaderStrongFixed, RowHeaderStrongFixed, FooterStrongFixed,
    RowFooterStrongFixed /
    font = fonts("FixedStrongFont")

```

```

;

class Caption /
  background = _undef_
  font = body_font
  fontweight = medium
  borderwidth = 0
;

class BeforeCaption /
;

class AfterCaption /
;

style Note from Output /
  background = a00000000
  padding = 5pt
;

class NoteContent /
;

class NoteContentFixed, WarnContentFixed, ErrorContentFixed,
  FatalContentFixed /
  font = fonts("FixedFont")
;

class NoteBanner /
  pretext = text("Note Banner")
  fontweight = bold
;

class WarnBanner /
  pretext = text("Warn Banner")
  fontweight = bold
;

class ErrorBanner /
  pretext = text("Error Banner")
  fontweight = bold
;

class FatalBanner /
  pretext = text("Fatal Banner")
  fontweight = bold
;

class Table /
  frame = box
  rules = all
  background = a00000000
  borderwidth = 0.5pt
  bordercolor = &secondary
  bordercollapse = separate
  borderstyle = solid
;

```

```

class Data /
    color = dark1
    background = _undef_
    bordercolor = &secondary
    borderwidth = 0.5pt
    borderstyle = solid
    ;

class DataFixed /
    font = fonts("FixedFont")
    ;

class DataEmphasis /
    fontstyle = italic
    ;

class DataEmphasisFixed /
    font = fonts("FixedEmphasisFont")
    ;

class DataStrong /
    fontweight = bold
    ;

class DataStrongFixed, HeaderStrongFixed /
    font = fonts("FixedStrongFont")
    ;

class LineContent /          ;

class UserText /
    just = left
    ;

class Batch /
    background = &secondary
    font = fonts("BatchFixedFont")
    paddingtop = 4pt
    paddingbottom = 4pt
    paddingleft = 5pt
    paddingright = 5pt
    cellspacing = 0
    ;

class List /
    MarginLeft = 27pt
    TextIndent = 27pt
    fontsize = 32pt
    padding = 1pt
    background = _undef_;
    ;

class ListItem /
    font = body_font
    liststyletype = disc
    ;

```

```
class ListItem2 /
    liststyletype = circle
    ;

class ListItem3 /
    liststyletype = square
    ;

class Paragraph /
    background = _undef_
    ;

end;

%mend;

%PowerPointStyle(Light, cx000000, cxffffff, cx00709f, cx7A40A6, light2,
    cxCCCCCC);
%PowerPointStyle(Dark, cxffffff, cx000000, cx26A0DA, cx9550C0, dark2,
    cx9b9b9b);
```