Presentation Quality Bulleted Lists Using ODS in SAS® 9.2

Karl M. Kilgore, PhD, Cetus Group, LLC, Timonium, MD

ABSTRACT

Business reports frequently include bulleted lists of items: summary conclusions from a larger detailed report, a list of next steps, listings of “Other” responses to coded response items, etc. Even when the bulk of an analytical report is produced by reporting tools in SAS®, the bulleted lists and surrounding explanatory text are typically produced using another tool (e.g., Microsoft Word) and then combined with the quantitative report after the fact. This paper describes a simple yet powerful method for incorporating publication quality bulleted lists directly into SAS output using Output Delivery System (ODS) styles. Specifically, this method: 1) indents variable text after a bullet symbol and wraps multiple lines of text within a list item to the indentation point; 2) allows for an arbitrary number of indentation levels, using either consistent bullet symbols or symbols which vary by indentation level; 3) has the capability for blocks of normal text (i.e., without bullets) for, e.g., an introductory or closing paragraph; 4) does not interfere with or override the styles of the parent element. This method is implemented as a simple COMPUTE block in the REPORT procedure. It is specifically intended for the ODS printer family destination, but minor modifications to the basic method allow its extension to the markup family and RTF destinations as well.

INTRODUCTION

Business reports frequently include bulleted lists of items: summary conclusions from a larger detailed report, a list of next steps, listings of “Other” responses to coded response items, etc. Even when the bulk of an analytical report is produced by reporting tools in SAS®, the bulleted lists and surrounding explanatory text are typically produced using another tool (e.g., Microsoft Word) and then combined with the quantitative report after the fact. This paper describes a simple yet powerful method for incorporating publication quality bulleted lists directly into SAS output.

THE ANATOMY OF A BULLETED LIST

The Chicago Manual of Style (CMOS) considers bulleted lists to be one type of the more general category of “vertical lists”, the other type being numbered lists. Stylistically, CMOS recommends that “a vertical list is best introduced by a complete grammatical sentence, followed by a colon. . . If items run over a line, the second and subsequent lines are usually indented” (CMOS, 2003). Conventionally, bulleted lists are frequently indented from the left margin, and secondary levels of bullets have an additional indent to visually underscore their subordinate nature. Figure 1 summarizes the generic characteristics of a bulleted list and will also provide us with a target document which we hope to implement in SAS.
Ideally, we want our bulleted list program to do the following:

- Display a bullet symbol as the first character in each list item and “out-dent” (i.e., indent to the left) the first line of text in the list
- Indent variable text after the bullet symbol and wrap multiple lines of text within a list item to the indentation point
- Allow for an arbitrary number of indentation levels, which:
  - Either have consistent bullet symbols throughout the list or symbols which vary by indentation level, and
  - Display text wrapping on the second and successive lines which is appropriate for the indentation level
- Permit blocks of normal text (i.e., without bullets) for, e.g., an introductory or closing sentence or paragraph

Finally, it is highly desirable that the bulleted lists produced by our module co-exist with any pre-existing formatting and not interfere with or override styles of the parent element.

Figure 1. A Sample Bulleted List

Because the focus of this paper is on printed reports (including PDF), the method presented here focuses on the SAS Output Delivery System (ODS) printer family destinations. Some brief comments on extending this method to the RTF and HTML destinations are provided near the end.

CONSTRUCTING A BULLETED LIST

Over ten years ago, Schellenberger (2000) presented a method which used the REPORT procedure for outputting near-publication quality text to the ODS printer destinations, using SAS version 8.2. Further ODS refinements in subsequent versions have added enhancements to font handling and line spacing, and we will build on his groundbreaking work here to produce our bulleted lists.

Step 1: Getting the text of the list elements and other necessary data into a SAS data set. The input to Proc Report is a simple data set with a minimum of two fields, called here “IndentLevel” (a numeric field containing the indentation level of the associated list element) and “ListContent” (a character field of sufficient length to hold the longest list item. In the current version of our bulleted list method, IndentLevel 0 is reserved for un-bulleted and un-indented text. IndentLevel 1 and higher are used to insert a bullet and increasing amounts of indentation. The first few lines of code are shown below. (The Appendix contains a listing of the entire program.)

```sas
DATA bullet;
  length ListContent $400;
  IndentLevel = 0;
  ListContent = "Ideally, we want our bulleted list program " ||
    "to do the following:";
  output;
  IndentLevel = 1;
  ListContent = "Display a bullet symbol as the first character " ||
    "in each list item and ""out-dent"" (i.e., indent to the " " ||
    "left) the first line of text in the list";
  output;
```

Note that one of the advantages of this method is that data (the text which is the content of the bulleted list) is separated from formatting (the instructions that construct the list). It is not necessary to add any embedded codes or
other special instructions in the data itself. The only pre-processing requirements are that each list item be contained in its own logical record, and that each record included a field containing the desired indentation level.

**Step 2: Setting the Formatting for the Bulleted List.** Three parameters are required: the symbol to use for the bullet, how much to indent the text of the list item itself, and how much to outdent the first line of each list element. In our target list (see Figure 1), there are two levels of bullets (not counting the special case of IndentLevel 0, used here for the opening sentence and closing paragraph), so we need two instances of each of the three parameters. These are implemented as macro variables and are displayed below.

```plaintext
%let Bullet1 = ^{style [fontfamily=symbol]·};
%let Bullet2 = ^{style [fontfamily=symbol]o};
%let MarginLeft1 = 0.3in;
%let MarginLeft2 = 0.6in;
%let Outdent1 = -1.1em;
%let Outdent2 = -1.1em;
```

The “Bullet” parameter contains an ODS escape character (here “^”), the text of an inline style to temporarily change to our desired font, and the specific bullet character itself. Note that here, we chose to use different bullets for the two different indentation levels. If you wanted to use the same bullet, you would just set the two bullet parameters to the same value.

The “MarginLeft” parameter is the distance to shift the text of the list item to the right of the default left margin. You can vary this to suit your own taste.

The “Outdent” parameter specifies the amount to shift the first line (the line containing the bullet character) to the left of the rest of the text. Because the horizontal space that the bullet character and the white space that follows it are not necessarily known in advance, this parameter can only be determined by trial and error. In our experience, we find it most convenient to specify this parameter in “em” units, the standard typesetting measurement unit for width, and that a value between 1 and 2 typically provides acceptable results. For this example, we could use the same value for both Outdent parameters, but that may vary, depending on the width of the bullet characters you select.

You need one set of three parameters for each indentation level. In theory, you can have as many levels as you want, but in practice, more than two or three becomes confusing to the reader and you run the risk of your MarginLeft parameter exceeding your right hand margin.

**Step 3: Generating the Bulleted List with Proc Report.** The final step is to generate the bulleted list itself. The program below assumes that the ODS environment (margins, file destination, overall style template, etc.) has been specified and the ODS escape character has been set to “^”.

```plaintext
PROC REPORT data = bullet nowindows noheader
   style(report) = [rules=none frame=void cellspacing=0]
   style(column) = [cellwidth=4.5in]
;
columns IndentLevel ListContent;
define ListContent / display;
define IndentLevel / display noprint;

COMPUTE ListContent;
   if IndentLevel=1 then call define(_row_,'style',
      'style={marginleft=&MarginLeft1 textindent=&Outdent1
      pretext="&Bullet1  "}');
   else if IndentLevel=2 then call define(_row_,'style',
      'style={marginleft=&MarginLeft2 textindent=&Outdent2
      pretext="&Bullet2  "}');
endcomp;
title;
run;
quit;
```

The Proc REPORT statement specifies that column headers be turned off, and includes a style override for the entire report which turns off all rules and the frame around the report. It also sets the cellspacing style attribute to 0, which allows for better control of the spacing between list elements. (This spacing can be adjusted using the margintop
and/or marginbottom style attributes. To avoid over-complicating this example, however, we have chosen simply to use the defaults from the parent style.)

In addition to those required elements of the Proc Report statement, we have used another style override to set the column width. This was done simply to emulate the width of our target list in Figure 1. In practice, this style override would be set to match the desired width of the list, or simply omitted altogether, which would produce a list which used the width of the printed page.

The next two statements simply define the columns. Note that the IndentLevel variable is used for formatting only, and should not be displayed. So we set its definition to “display noprint”.

The COMPUTE block is where the important work is done. You need one “if” condition for each level of bullets. (Here, again for simplicity’s sake, we just let IndentLevel 0 use the default style.) The marginleft style attribute shifts all the text to the right. The textindent attribute here takes on a negative value, to shift the first line back to the left, producing the out-dented bullet beginning each list item. Finally, the pretext attribute prepends the appropriate bullet character and adds some white space for aesthetic reasons. Here, we simply use two spaces, but that is a matter of taste.

We should make one additional comment regarding our use of the “textindent” style attribute. The SAS 9.2 ODS Manual indicates that the value of this attribute “specifies the number of spaces that the first line of output will be indented.” In fact, you can use any of the standard dimensions: cm, em, ex, in, mm or pt.

The results of running the program, using the default style template for the PDF destination is shown in Figure 2. It can be seen that we have been quite successful at closely matching the layout of our target.

<table>
<thead>
<tr>
<th>Ideally, we want our bulleted list program to do the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Display a bullet symbol as the first character in each list item and &quot;out-dent&quot; (i.e., indent to the left) the first line of text in the list</td>
</tr>
<tr>
<td>• Indent variable text after the bullet symbol and wrap multiple lines of text within a list item to the indentation point</td>
</tr>
<tr>
<td>• Allow for an arbitrary number of indentation levels, which:</td>
</tr>
<tr>
<td>• Either have consistent bullet symbols throughout the list or symbols which vary by indentation level, and</td>
</tr>
<tr>
<td>• Display text wrapping on the second and successive lines which is appropriate for the indentation level</td>
</tr>
<tr>
<td>• Permit blocks of normal text (i.e., without bullets) for, e.g., an introductory or closing sentence or paragraph</td>
</tr>
</tbody>
</table>

Finally, it is highly desirable that the bulleted lists produced by our module co-exist with any pre-existing formatting and not interfere with or override styles of the parent element.

**Figure 2. The Bulleted List Generated by our SAS Method**

Finally, as stated in the last sentence in our sample list, we want our method to respect as much as possible the style of the larger document in which our list is embedded. To assess our degree of success, we made a change to our original program, as shown below.

```ods pdf notoc file="c:\temp\BulletList3.pdf" style=gears;```

Whereas we originally used the default style template for PDF, here we specified a different style, with a colored background and a different, slightly larger font. Figure 3 shows the output. It can be seen that our method is quite robust to changes in the style attributes it inherits from the parent document. (To make full disclosure, we did, in fact...
tweak both the Outdent parameters by a few tenths of an em to improve the appearance of the line wrap.) The only
notable incompatibility known as of this writing concerns the use of the marginleft attribute within the Table style
element of the style template. It may be necessary to add the marginleft value from the style template to the
marginleft value in the COMPUTE block of this method to achieve entirely satisfactory results.

![Illustration](image)

**Ideally, we want our bulleted list program to do the following:**

- Display a bullet symbol as the first character in each list item and "out-dent" (i.e., indent to the left) the first line of text in the list
- Indent variable text after the bullet symbol and wrap multiple lines of text within a list item to the indentation point
- Allow for an arbitrary number of indentation levels, which:
  - Either have consistent bullet symbols throughout the list or symbols which vary by indentation level, and
  - Display text wrapping on the second and successive lines which is appropriate for the indentation level
- Permit blocks of normal text (i.e., without bullets) for, e.g., an introductory or closing sentence or paragraph

Finally, it is highly desirable that the bulleted lists produced by our module co-exist with any pre-existing formatting and not interfere with or override styles of the parent element.

Figure 3. The Same Bulleted List with only a Change to the Parent Style

**OTHER ODS OUTPUT DESTINATIONS**

Although this paper has focused on the ODS printer destinations, the method presented here does, in fact, work
without modification for the RTF destination as well. As for the HTML destination, this method is probably not needed,
as HTML has bulleted list capability built in, through the use of the `<ul>` (unordered list) and `<li>` (list item) tags.
However, should the need arise, two modifications to the method allow its extension to the HTML destination.

First, the specification of the bullet character must be coded in standard HTML. Recall that the information in our
"Bullet" parameter is essentially a SAS inline style contained in a macro variable, which becomes part of a "pretext=" attribute in the output. For the HTML destination, SAS outputs the pretext attribute value as is, without converting it to
standard HTML. So we must store the actual HTML in the "Bullet" parameter. The code below highlights this change:

```sas
%let Bullet1 = %nrstr(<span style="font-family: symbol;">·
&nbsp;&nbsp;</span>);
%let Bullet2 = %nrstr(<span style="font-family: symbol;">o
&nbsp;&nbsp;</span>);
```

The code within the parentheses is HTML for changing to the desired font, the character for the bullet, and the HTML
entity codes for 2 non-breakable spaces. Because the HTML non-breakable space entity begins with an ampersand,
the macro quoting function `%nrstr()` must be used to prevent the SAS macro processor from trying to interpret it as a
macro variable.

Second, the marginleft attribute, which we used to indent the text of the bullet, does not work the same within an
HTML table cell as in other destinations. But the same effect is can be accomplished by replacing the marginleft
attribute(s) in the COMPUTE block with paddingleft. The relevant changes are highlighted in the code below:

```sas
COMPUTE ListContent;
```
CONCLUSION

In SAS 9.2, one can use PROC REPORT and inline styles to embed near typeset quality bulleted lists into reports. The method presented here uses only a few statements in a COMPUTE block to gain flexible and aesthetically appealing bulleted lists which co-exist well with the parent style.

REFERENCES


http://support.sas.com/documentation/cdl/en/odsug/61723/HTML/default/a002972093.htm#a002604907

CONTACT INFORMATION

Karl M. Kilgore, PhD  
Vice President, Research & Outcomes  
Cetus Group, LLC  
9690 Deereco Rd., Suite 840  
Timonium, MD 21093  
kkilgore@cetusgroup.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 registered trademarks or trademarks of their respective companies.
APPENDIX

Complete listing of the program which produced Figure 2.

* Create the sample data set;

```sas
data bullet;
  length ListContent $400;
  IndentLevel = 0;
  ListContent = "Ideally, we want our bulleted list program to do the following:";
  output;
  IndentLevel = 1;
  ListContent = "Display a bullet symbol as the first character in each list item " ||
    "and ""out-dent"" (i.e., indent to the left) the first line of text in " ||
    "the list";  
  output;
  ListContent = "Indent variable text after the bullet symbol and wrap " ||
    "multiple lines of text within a list item to the indentation point";
  output;
  ListContent = "Allow for an arbitrary number of indentation levels, which:";
  output;
  IndentLevel = 2;
  ListContent = "Either have consistent bullet symbols throughout the list or " ||
    "symbols which vary by indentation level, and";
  output;
  ListContent = "Display text wrapping on the second and successive lines which " ||
    "is appropriate for the indentation level";
  output;
  IndentLevel = 1;
  ListContent = "Permit blocks of normal text (i.e., without bullets) for, " ||
    "e.g., an introductory " ||
    "or closing sentence or paragraph";
  output;
  IndentLevel = 0;
  ListContent = "Finally, it is highly desirable that the bulleted lists " ||
    "produced by our module co-exist with any pre-existing formatting " ||
    "and not interfere with or override styles of the parent element.";
  output;
run;
```

* Set the bulleted list parameters;

```sas
%let Bullet1 = ^{style [fontfamily=symbol]·};
%let Bullet2 = ^{style [fontfamily=symbol]o};
%let MarginLeft1 = 0.3in;
%let MarginLeft2 = 0.6in;
%let Outdent1 = -1.1em;
%let Outdent2 = -1.1em;
```

* Set up the ODS environment;

```sas
ods listing close;
ods escapechar='^';
options nodate nonumber;
ods pdf notoc file="c:\temp\BulletList1.pdf";
```

* Generate the report;

```sas
proc report data = bullet nowindows noheader
  style(report) = [rules=none frame=void cellspacing=0]
  style(column) = [cellwidth=4.5in]
;
columns IndentLevel ListContent;
define ListContent / display;
define IndentLevel / display noprint;

compute ListContent;
```

Posters
NESUG 2010

7
if IndentLevel=1 then call define(_row_, 'style',
    'style={marginleft=&MarginLeft1 textindent=&Outdent1 pretext="&Bullet1 "}');
else if IndentLevel=2 then call define(_row_, 'style',
    'style={marginleft=&MarginLeft2 textindent=&Outdent2 pretext="&Bullet2 "}');
endcomp;
title;
run;
quit;

* Clean up;
ods _all_ close;
ods listing;
options date number;