The Objective:
Develop Code to Easily Generate Bulleted Lists

Specifications: What do we want our module to do?

> 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.

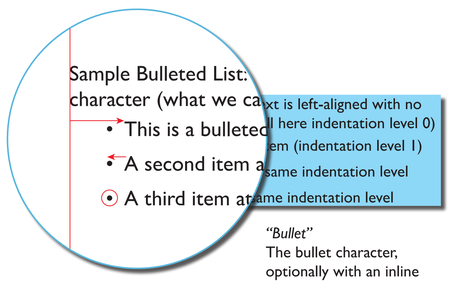
We will use the bulleted list above as our target output.

The Method:
Parameters, Data and Code

Step 1: 3 parameters, implemented as global macro variables

"MarginLeft"
How far from the left margin to the text on the list item, specified as a **positive** number of standard units (e.g. in)

"OutDent"
How far to shift the 1st line of the list item (the line with the bullet) to the left of the "marginleft" parameter, specified as a **negative** number of standard units ("em" works best")



"Bullet"
The bullet character, optionally with an inline style to select font family, color, etc.

If your list contains multiple indentation levels (our target list contains 2), there can be multiple instances of the 3 parameters, differentiated by appending a digit to the macro variable name, e.g., Bullet1, Bullet2.
See example code A

Step 2: Prepare your data — only 2 fields are required

The text of the bulleted list is in a data set, one list item per record, in a character field of sufficient length to hold the longest list item

Called "ListContent" in the sample program

A numeric field containing the indentation level of the associated list element

Called "IndentLevel" in the sample program

Indent level "0" is used for a list item with no bullet and no indentation

See example code B

Step 3: Display the list with Proc REPORT

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 vertical spacing between list elements.

There are additional statements to (optionally) set the column width and suppress the display of the IndentLevel field, which is used only for formatting.

See example code C

The COMPUTE block is where the important work is done.

- You need one "if" condition for each indentlevel of bullets.
- The marginleft style attribute shifts all the text to the right.
- The textindent attribute, which takes a negative value, shifts the first line to the left, producing the out-dented bullet beginning each list item.
- Finally, the pretext attribute prepends the appropriate bullet character and some white space, purely for aesthetics.

See example code D

The Results

The output from running the sample program using the default style template for the PDF designation. We closely match the layout of our target bulleted list.

> 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.

Our method respects style changes to the parent element. This output was created by the sample program using a single change, invoking a different style sheet:

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

> **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**
>
> **{ … Rest of example has been truncated … }**

We have shown that our method works with printer destinations. The following example simply changes the ODS destination from PDF to RTF, with the default style sheet. Again, the output matches our target list well.

> 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
>
> { … Rest of example has been truncated … }

As for HTML, this method is probably not needed, as HTML has bulleted list capability built in, through the use of the <ul> (unorderd list) and <li> (list item) tags. However, should the need arise, our method can be modified to allow extension to the HTML destination. (See code E in the Code Listing.) The example shows the output, using the "money" style sheet.

> **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**
>
> **{ … Rest of example has been truncated … }**

The Code:
Complete Listing of the Program

A

```
* Set the bulleted list parameters;
%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;
```

B

```
* Create the sample data set;
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 up the ODS environment;
ods listing close;
ods escapechar='^';
options nodate nonumber;
ods pdf notoc file="c:\temp\BulletList1.pdf";
```

C

```
* Generate the report;
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;
```

D

```
 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;
```

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

E

```
* Code changes for HTML destination;
%let Bullet1 = %nrstr(<span style=""font-family: symbol;"">·
    </span>);
%let Bullet2 = %nrstr(<span style=""font-family: symbol;""><o
    </span>);

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


CONTACT INFORMATION

Karl M. Kilgore, PhD
Vice President, Research & Outcomes
Cetus Group, LLC
9690 Deereco Rd., Suite 840
Timonium, MD 21093
kkilgore@cetusgroup.com