

## Tag Clouds - A list of tokens, sized by relative frequency

Richard A. DeVenezia, Independent Consultant, Remsen, NY

### Abstract

A tag cloud is a list of tokens, wherein the text size of a token varies according to how frequently the token has appeared within a collection of texts or text streams. Tag clouds have been found useful in various Internet search engines and some blogging frameworks.

This paper will demonstrate how to render a tag cloud into the ODS PDF destination and use it as a navigation aid at the top of a document. Code was submitted in SAS® Foundation version 9.2.

Keywords: Tag Clouds, Tokens, ODS, PDF

### Data

Token data can be pulled from a number of sources. In some scenarios a parsing process will scan numerous text or narrative objects for meaning rich tokens. In other scenarios the tokens will be retrieved directly from a log, such as an Apache web server log, or from the output of a log parser, such as AWStats.

It is outside the scope of this paper to discuss parsing. Instead, sample data will be dynamically generated to support these suppositions:

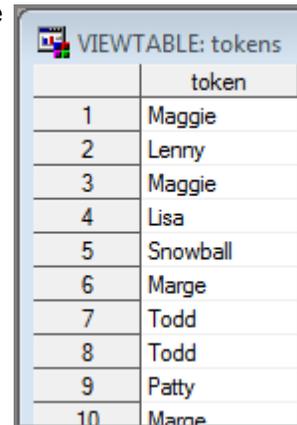
- A web poll occurred, asking participants to vote for their favorite Simpson's character. 20 characters were listed for selection.
- 2,000 votes were collected.
- There is a normal distribution in the votes.
- Each character has a distinct graph (unrelated to the act of voting, perhaps a biorhythm curve)

Thus, the sample data is simply a list of 2,000 token values.

### Data preparation

The first step is to compute the frequency of each token.

```
proc freq noprint data=tokens;  
  table token / out=tokenfreqs;  
run;
```



	token
1	Maggie
2	Lenny
3	Maggie
4	Lisa
5	Snowball
6	Marge
7	Todd
8	Todd
9	Patty
10	Marge

The values in the PERCENT column in the output table *tokenfreqs* will range from 0 to 100. This is a good basis for computing font sizes to be used when showing a token value in the Tag Cloud.

	token	COUNT	PERCENT
1	Bamey	121	6.05
2	Bart	174	8.70
3	Homer	138	6.90
4	Krusty	35	1.75
5	Lenny	36	1.80
6	Lisa	197	9.85
7	Maggie	190	9.50
8	Marge	175	8.75

You would not want to use PERCENT directly as a font size because of the possible extremes. Very low values would be nigh invisible, and the large ones overpowering.

The second step is to map the PERCENT values to a reasonable range of point sizes. I have chosen a low of 8pt for the minimum percent and a high of 32pt for the maximum percent. The mapping is linear. Other mappings such as weighted, logarithmic, or exponential are also possible.

During preparation the token values are also compressed into a tag value that can be used as a name. The compression keeps only digits, letters and underscores.

```
proc sql;
  create table token_freqs as
  select
    *
    , ( (percent) - min(percent) )
    / ( max(percent) - min(percent) )
    * ( &highPt - &lowPt )
    + &lowPt as size
    , compress(token, 'kno') as tag
  from tokenfreqs
  ;
quit;
```

	token	COUNT	PERCENT	size	tag
1	Bamey	121	6.05	22.1	Bamey
2	Bart	174	8.70	29.0	Bart
3	Homer	138	6.90	24.3	Homer
4	Krusty	35	1.75	10.9	Krusty
5	Lenny	36	1.80	11.0	Lenny
6	Lisa	197	9.85	32.0	Lisa
7	Maggie	190	9.50	31.1	Maggie
8	Marge	175	8.75	29.1	Marge
9	Mel	16	0.80	8.4	Mel
10	Moe	129	6.45	23.1	Moe
11	Mr. Burns	74	3.70	16.0	MrBurns

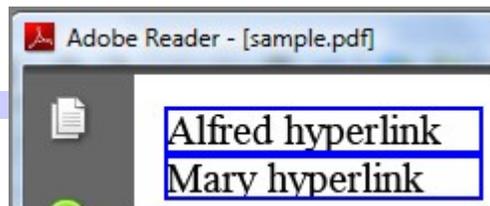
## Within PDF Links

Usage Note 12660 demonstrates a technique for creating links within a document. A link involves a trigger spot (text) and a destination (anchor). The note show that the trigger can be created using an ODS PDF TEXT statement that incorporates the URL in-line styling directive:

```
ods pdf text="^S={URL='#destination_name'}Trigger text";
```

The anchor can be created using an ODS PDF ANCHOR statement:

```
ods pdf anchor="destination_name";
```



The caret (^) in the text= value is the ODS ESCAPECHAR. This character tells the ODS processor to escape normal processing and handle any directives specified within the text value.

## Blue Border

As shown, and according to Usage Note 24182, a trigger region is rendered with a border having a color as defined by the LINKCOLOR attribute of the Document class in the active style template. In the **default** template this color is a blue hue, ultimately inherited from color name value pair 'fgB2'=cx0066AA. This color can be set to white so as to be essentially invisible. In-line styling directives are one way to set the LINKCOLOR value:

```
ods pdf text="^S={URL='#destination_name' LINKCOLOR=white}Trigger text";
```

The borders are only shown when using the Adobe Reader application, and never appear in printed output.

## Sizing link text

The font size of the link text can also be changed via in-line styling with the FONT\_SIZE option.

```
ods pdf text=
"^S={URL='#destination_name'
  LINKCOLOR=white
  FONT_SIZE=18pt
}Trigger text";
```

## Tag cloud strategy

A one cell table containing snippets of in-line styled text to set destinations and text sizes would qualify as a tag cloud. The ODSOUT Component Object is a versatile agent for generating the cell from the scaled frequency counts. ODSOUT is experimental.

## Pattern

A string is to be constructed that follows this pattern:

```
^S={URL='#tag1' FONT_SIZE=size1pt LINKCOLOR=white}token1
...
^S={URL='#tagN' FONT_SIZE=sizenpt LINKCOLOR=white}tokenN
```

This string can become quite long, but in general will not exceed the maximum DATA Step string length of 32,767. The following code shows the links being accumulated as control loops around a SET reading the tokens and their scaled frequencies. The linkcolor is set to RED for debugging purposes:

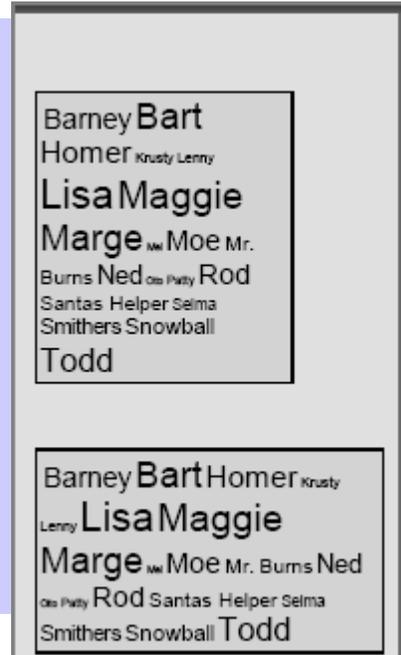
```
data _null_;
  length render $32760;
  do until (last_tag);
    set token_freqs end=last_tag;
    link = catt
      ( " ^S={URL='#" , tag , "' font_size=" , size , 'pt linkcolor=RED}'
      , token
      , '^S={}'
      );
  ;
end;
```

```
render = catt (render, link);
end;
```

The ODSOUT object is used to create a table with one cell containing all the links. The ODS PDF layout engine will wrap words to fit the size of the region. The tag cloud is rendered using different width and heights to demonstrate the layout handling.

```
declare odsout cloud();
cloud.layout_absolute();
cloud.region
( x:"4.2in", y:"0in"
, width:"3in", height:"4in"
);
cloud.table_start();
cloud.row_start();
cloud.format_cell(text:render);
cloud.row_end();
cloud.table_end();

cloud.region
( x:"4.2in", y:"4in"
, width:"4in", height:"3in"
);
cloud.table_start();
cloud.row_start();
cloud.format_cell(text:render);
cloud.row_end();
cloud.table_end();
cloud.layout_end();
```



## Uh-oh!

The token values were rendered at the right sizes, and automatically layed out. However, there is no red border. This means that the ODS PDF processor did not create links as desired. It turns out that at present, link generation in PDFs will only occur at the ODS cell level. It can not occur at an arbitrary span level.

## Work around

One work around is to create a table with one token per row. This table does not meet the desired single cell criteria of a tag cloud, however, it is a navigation aid with frequency hinting.

```
cloud.layout_absolute();
cloud.region(x:"0in",y:"0",width:"1.9in",height:"10in");
cloud.table_start();
do _n_ = 1 by 1 until (last_token);
set token_freqs end=last_token;
cloud.row_start();
cloud.format_cell
( text:token
, overrides:
catt(' FONT_SIZE=',size,'pt')
|| catt(" URL='#",tag,"")
```



```

||      ' LINKCOLOR=red' );
cloud.row_end();
end;
cloud.table_end();
cloud.layout_end();

```

## Content generation

The content linked to in the sample code is created by SAS/GRAPH. The simplest way to create multiple pages of one-page-per-plot content is to use a BY statement:

```

goptions reset=all htext=11pt hby=0;

symbol1 v=dot;

axis1 minor=none label=none;
axis2 minor=none label=none order=-1 to 1 by .5;

title1 h=20pt " ";
title2 h=14pt "#byval1 (votes=#byval3)";

ods pdf startpage=yes; * one page per plot;

proc gplot data=chart_info;
  by token tag count;
  plot y * x / vaxis=axis2 haxis=axis1;
run;
quit;

```

The code will create content with associated destination anchors named IDX1 through IDX20. The anchor names do not match the tag values used during link creation (URL=#destination), so the links do not work. The content generation has to be changed to create anchor names that match the computed tag values. At present, these anchor names can not be created using a plot option such as /ANCHOR="#byval2". The code would be quite succinct if it were. Instead, a macro is written to run the plot code for each by group:

```

%macro plot_token (value=, tag=);
  ods pdf anchor="&tag"; * Create appropriately named destination;

  title1 h=20pt " ";
  title2 h=14pt "#byval1 (votes=#byval3)";

  proc gplot data=chart_info;
    by token tag count;
    plot y * x / vaxis=axis2 haxis=axis1;
    where token="&value";
  run;
  quit;
%mend;

data _null_;
  set token_freqs;
  statement = cats('%nrstr(%plot_token(value=', token, ', tag=', tag, '));');
  call execute (trim(statement));
run;

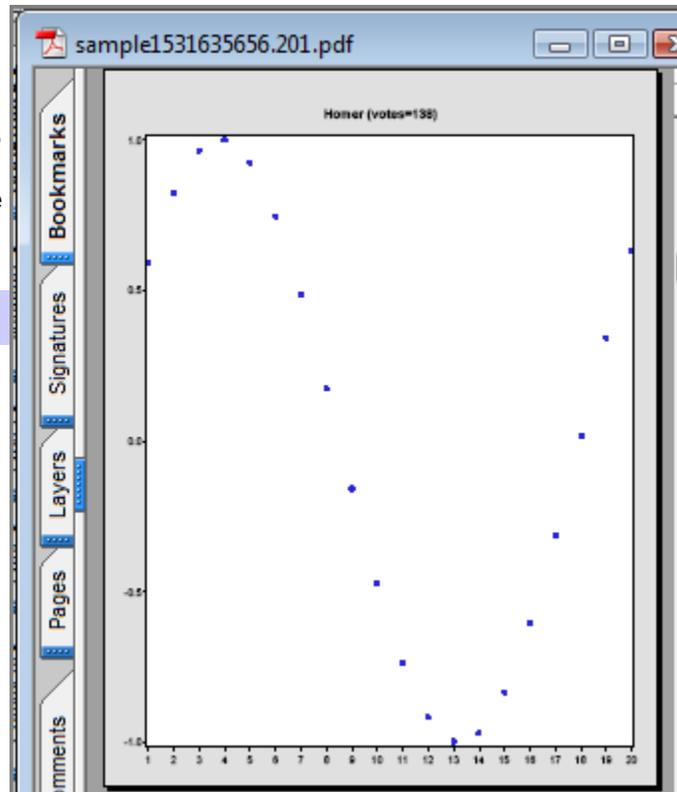
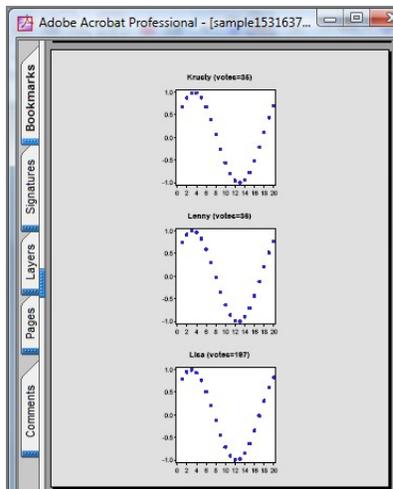
```

The code generates 20 pages of plots, with destinations named so that the link in the cloud table will work.

## Fewer pages

20 pages of output is fine and good, however it may be too many for the intended audience. One simple way to get more content per page is to output smaller plots so that there can be more than one plot per page. The following settings do just that:

```
goptions hsize=3in vsize=3.5in;  
ods pdf startpage=NO;
```



## Uh-oh #2

When STARTPAGE=NO the ODS system creates a PDF that has link boxes that do not always go to the intended page! This issue is pretty much a deal breaker for making a SAS based tag cloud table in the PDF destination.

## Unexplored

Proc DOCUMENT has many features for dealing with ODS output in a generic or abstract manner. SAS/GRAPH has a PDFC device driver for creating documents. Neither of these were examined as the basis for tag cloud generation.

## Conclusion

Implementing a new design idea takes patience and often leads to dead-ends.

A tag cloud is a useful navigation aid for reaching high frequency content. Implementing a SAS based tag cloud generators for the PDF destination is not yet ready for prime-time. When the

ODS PDF engine is updated to handle creation of link texts in an arbitrary context, tag cloud creation as discussed in this paper will be possible.

## Contact Information

Richard A. DeVenezia  
9949 East Steuben Road  
Remsen, NY 13438  
(315) 831-8802  
<http://www.devenezia.com/contact.php>

Richard is an independent consultant who has worked extensively with SAS products for over fifteen years. He has presented at previous SUGI, NESUG and SESUG conferences. Richard is interested in learning and applying new technologies. He is a SAS-L Hall of Famer and remains an active contributor to SAS-L.

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.

This document was produced using OpenOffice.org Writer.