PhUSE EU Connect 2018

Paper DV06

Streaming Tables and Listings by PROC STREAM

Charumathy Sreeraman, Ephicacy Lifescience Analytics, Bangalore, India
Samundeeswari Raja, Ephicacy Lifescience Analytics, Bangalore, India

ABSTRACT

Volume of data collected during a clinical study is huge. Maintaining confidentiality, integrity and availability of such data is vital in the process of clinical trials. Safety and efficacy analysis of a drug is done by summarizing the key elements in such data as tables, listings and figures through statistical analysis. An interactive and dynamic data visualization of tables and listings of the safety reports enhances the understanding and uncomplicates reviewing/validation process. PROC STREAM enables streamlining of such visualizations in case of bulky safety reports. This procedure can be used to produce virtually any text-based content, including but not limited to HTML, CSV, RTF, XML and many more. This paper shall discuss about the quantification of safety reports using PROC STREAM procedure through simple coding with minimal effect on double programming.

INTRODUCTION

The regulatory authorities insist that complete and extensive documentation of the clinical data collected is must for the authorization of the drug into the market. The documentation to be submitted to the regulatory authorities must prove the quality, safety, and efficacy of the new drug. One of the most critical documents submitted for this facilitation is the Clinical Study Report (CSR), which represents the integrated full report of the efficacy and safety data for an individual study with a therapeutic or diagnostic agent. The CSR integrates the clinical and statistical descriptions, presentations, and analyses into a single integrated report, incorporating tables and figures.

One of the major components in CSR is the summary tables and listings. They display a mixture of discrete (categorical) and/or continuous variables in a style that could be referred to as “summary statistics in rows”. These tables and listings are used for both safety and efficacy data summaries.

Depending the volume of the data collected, these summary tables and listings can be very elaborate and will take up more energy for validation/review. In this paper, we propose a way to use the PROC STREAM procedure on these bulky reports, to convert them as an interactive and dynamic data visualization for enhanced understanding, and uncomplicated validating/reviewing of the data present.

PROCEDURE SYNTAX

PROC STREAM statement specifies an external file with the “`OUTFILE=`” keyword, as well as options. The `OUTFILE=fileref` specifies the file where all tokens are written. The arbitrary text is sandwiched inside a “BEGIN” and a four semi-colon ending “;;;;”. The `text` specifies the SAS statements or macros to use with PROC STREAM. There is no “RUN” or “QUIT”.

```
PROC STREAM OUTFILE= fileref <option(s)>;
BEGIN
<text-1>
<text-n>
;;;;
```

BASIC EXAMPLE

```
filename new 'mytable.htm1';
%macro make_table (nrows, ncols);
      <table>
      %do i=1 %to &nrows;
      <tr>
      %do j=1 %to &ncols;
      <td>&j</td>
```

```
      %end;
      </tr>
      %end;
      </table>
%mend;
proc stream outfile=new; begin
      %make_table (5,3)
;;;;
```

The output produced will be

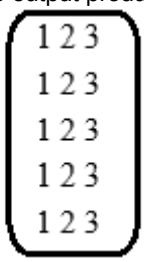

Figure 1. Sample Output

WHAT DOES THE STREAM PROCEDURE DO?

The STREAM procedure enables you to process an input stream that consists of arbitrary text that can contain SAS macro specifications. The macros are executed and expanded while the other text in the input stream is preserved. The text stream is not validated as SAS syntax. The output stream is sent to an external file that is referenced by a fileref and that can be defined to use any traditional SAS output destination.

SAS CODE TO CREATE THE HTML OUTPUT BY PROC STREAM

```
Filename _webout "/path/aesocptsev.html";

proc stream outfile=_webout;
BEGIN
%global pageTitle data hierarchy vars statistic
<html>
<head>
<title>pageTitle</title>
</head>
<body>
%let rc = %sysfunc(dosubl(%nrbquote(%raw)));
<table><tr><td>
</fieldset>
<legend>&pageTitle</legend>
%treeNodes(data = &data
          ,columns = &vars
          ,columnWidths = 100
          ,textWidth = 125
          ,divPrefix = R
          ,headers = Y
          ,expandedTo = 1
          ,tableAttributes = cellspacing="2" cellpadding="0"
          )
</fieldset>
</td></tr></table>
</body>
</html>
;;;;
```

The FILENAME statement assigns the path and name to the output which is to be produced by the PROC STREAM. The macro RC executes the SAS codes which creates the summary tables. The macro RAW which is in turn

assigned in macro RC, can be any of the summary tables or listings to which reporting will be done by PROC STREAM.

The macro TREENODES fetches the final dataset to be sent to PROC REPORT, utilizes it as the input dataset to create the nodes for drill-down HTML output by PROC STREAM. The macro includes basic HTML and Javascript. A basic understanding of these script by the reader is assumed.

CASE STUDIES

1. AE SUMMARY TABLE

AE summary tables are very important for physicians and pharmaceutical companies to assess the safety profile of the study drug. These tables usually display the number of adverse events, the number of patients in each treatment group in which the event occurred. Typically, AEs are grouped by System Organ Class, Preferred Terms and/or other variables of interest. In most clinical studies, we will have overall AE summary, summary tables of severity, outcome created. The figure 2 depicts the sample output of an AE summary table by PROC REPORT.

System Organ Class Preferred Term	Maximum/Severity	BP3304 (N =3) n(%)	BP3304 (N =3) Events	Placebo (N =3) n(%)	Placebo (N =3) Events	Overall (N =6) n(%)	Overall (N =6) Events
Any Treatment-Emergent Adverse Event	Mild	1 (33.3)		1 (33.3)		2 (33.3)	
	Moderate	2 (66.7)		2 (66.7)		4 (66.7)	
	Severe	0 (0.0)		0 (0.0)		0 (0.0)	
Gastrointestinal Disorders	Mild	1 (33.3)		0		1 (16.7)	
	Moderate	0 (0.0)		0		0 (0.0)	
	Severe	0 (0.0)		0		0 (0.0)	
Colitis Ulcerative	Mild	1 (33.3)		0		1 (16.7)	
	Moderate	0 (0.0)		0		0 (0.0)	
	Severe	0 (0.0)		0		0 (0.0)	
General Disorders And Administration Site Conditions	Mild	0 (0.0)		0		0 (0.0)	
	Moderate	1 (33.3)		0		1 (16.7)	
	Severe	0 (0.0)		0		0 (0.0)	
Chest Pain	Mild	0 (0.0)		0		0 (0.0)	
	Moderate	1 (33.3)		0		1 (16.7)	
	Severe	0 (0.0)		0		0 (0.0)	
Investigations	Mild	0		0 (0.0)		0 (0.0)	
	Moderate	0		1 (33.3)		1 (16.7)	
	Severe	0		0 (0.0)		0 (0.0)	
Blood Pressure Increased	Mild	0		0 (0.0)		0 (0.0)	
	Moderate	0		1 (33.3)		1 (16.7)	
	Severe	0		0 (0.0)		0 (0.0)	

Reference: Listing 16.2.7
Note: A treatment emergent adverse event (TEAE) will be any event that started on or after Day 1 up until the last dose date plus 14 days,inclusive. Treatment~
Subjects with more than one occurrence of a preferred term are counted only once.

Figure 2. AE Summary Table from PROC REPORT

Adverse Event By SOC and PT with Severity

	Severity/Intensity	A	B	T
⊞ Any Treatment-Emergent Adverse Event	Mild	1 (33.3)	1 (33.3)	2 (33.3)
⊞ Gastrointestinal Disorders	Mild	1 (33.3)	0	1 (16.7)
⊞ General Disorders And Administration Site Conditions	Mild	0 (0.0)	0	0 (0.0)
⊞ Investigations	Mild	0	0 (0.0)	0 (0.0)
⊞ Musculoskeletal And Connective Tissue Disorders	Mild	0	0 (0.0)	0 (0.0)
⊞ Skin And Subcutaneous Tissue Disorders	Mild	0	1 (33.3)	1 (16.7)
⊞ Vascular Disorders	Mild	0 (0.0)	0	0 (0.0)

Figure 3. AE Summary Table Output from PROC STREAM - With Collapsed Columns

Adverse Event By SOC and PT with Severity

	Severity/Intensity	A	B	T
⊞ Any Treatment-Emergent Adverse Event	Mild	1 (33.3)	1 (33.3)	2 (33.3)
⊞ Gastrointestinal Disorders	Mild	1 (33.3)	0	1 (16.7)
⊞ General Disorders And Administration Site Conditions	Mild	0 (0.0)	0	0 (0.0)
⊟ Investigations	Mild	0	0 (0.0)	0 (0.0)
	Moderate	0	1 (33.3)	1 (16.7)
	Severe	0	0 (0.0)	0 (0.0)
Blood Pressure Increased	Mild	0	0 (0.0)	0 (0.0)
	Moderate	0	1 (33.3)	1 (16.7)
	Severe	0	0 (0.0)	0 (0.0)
⊞ Musculoskeletal And Connective Tissue Disorders	Mild	0	0 (0.0)	0 (0.0)
⊞ Skin And Subcutaneous Tissue Disorders	Mild	0	1 (33.3)	1 (16.7)
⊞ Vascular Disorders	Mild	0 (0.0)	0	0 (0.0)

Figure 4. AE Summary Table with PROC STREAM - With expanded Columns

The figure 3 and figure 4 shows the AE summary tables with ease of accessing the required section without much tedious scrolling of the RTF output. These figures are the HTML output with interactive nodes for easy reviewing.

2. LABORATORY TABLE

The laboratory data is one of the most important parts of safety analysis in any drug discovery study. There are several summary tables commonly used to produce laboratory results. Descriptive statistics, clinically significant, shift tables and toxicity grade summaries are some of the summary tables that are commonly used. The shift table is one of the most frequently requested in a clinical study by statisticians or clinicians. The main purpose of shift tables in any clinical trial is to determine how the categorical result varies from baseline to post-dose.

The figure 5 shows the regular RTF output of the laboratory shift table , whereas figures 6 and 7 shows the interactive table with nodes created by PROC STREAM.

Treatment Variable [Normal range]	Visit	Timepoint	Baseline level	Low n (%)	Normal n (%)	High n (%)	Missing n (%)
EPHPO3 10 mg (N = 8)							
Sodium [136.0 - 145.0 mmol/L]	Day 1	2H	Low	0	0	0	0
			Normal	2 (25.0)	5 (62.5)	0	0
			High	0	1 (12.5)	0	0
	Day 2	24H	Low	0	0	0	0
			Normal	0	7 (87.5)	0	0
			High	0	1 (12.5)	0	0
	Day 4	72H	Low	0	0	0	0
			Normal	0	7 (87.5)	0	0
			High	0	1 (12.5)	0	0
	Follow-up		Low	0	0	0	0
			Normal	0	7 (87.5)	0	0
			High	0	1 (12.5)	0	0

Figure 5. LAB Shift Table Output from PROC REPORT

Lab Shift table

	low	normal	high	missed
⊞ EPHPO3 10 mg (N = 8)				
⊞ EPHPO3 20 mg (N = 6)				
⊞ EPHPO3 20 mg evening (N = 6)				
⊞ EPHPO3 20 mg fed (N = 6)				
⊞ EPHPO3 40 mg (N = 6)				
⊞ EPHPO3 80 mg (N = 6)				
⊞ EPHPO3 Placebo (N = 16)				
⊞ EPHPO3 Placebo evening (N = 3)				

Figure 6. LAB Shift Table Output from PROC STREAM – With Collapsed Columns

Lab Shift table

	low	normal	high	missed
⊞ EPHPO3 10 mg (N = 8)				
⊟ EPHPO3 20 mg (N = 6)				
Day 1				
Low	0	0	0	0
Normal	0	6 (100.0)	0	0
High	0	0	0	0
EPHPO3 20 mg (N = 6)				
Day 2				
Low	0	0	0	0
Normal	0	6 (100.0)	0	0
High	0	0	0	0
EPHPO3 20 mg (N = 6)				
Day 4				
Low	0	0	0	0
Normal	0	6 (100.0)	0	0
High	0	0	0	0
EPHPO3 20 mg (N = 6)				
Follow-up				
Low	0	0	0	0
Normal	0	6 (100.0)	0	0
High	0	0	0	0
⊞ EPHPO3 20 mg evening (N = 6)				
⊞ EPHPO3 20 mg fed (N = 6)				
⊞ EPHPO3 40 mg (N = 6)				
⊞ EPHPO3 80 mg (N = 6)				
⊞ EPHPO3 Placebo (N = 16)				
⊞ EPHPO3 Placebo evening (N = 3)				

Figure 7. LAB Shift Table Output from PROC STREAM

3. LABORATORY LISTING

The purpose of listing is to reciprocate the raw data collected in an orderly manner, to ensure the data integrity is achieved. It will be an elaborate document and will be difficult to manually review all the components. The below figures 8 and 9 are the standard and interactive outputs of the laboratory listings.

Subject info Variable [unit]	Visit	Timepoint	Date/time of measurement	Measured value	Flag	Investigator evaluation	Normal range	Change from baseline	Fasting status
1001/C1 (EPHPO3 10 mg)- 45 Years /M /WHITE/194 cm/86.0 kg									
Alanine Aminotransferase [U/L]	Screening		2011-05-12T07:16:00	29.7	NORMAL	Not allocable!	10.0 - 50.0		Y
	Day -1		2011-05-17T08:47:00	27.7	NORMAL	Not allocable!	10.0 - 50.0		Y
	Day 1	2H	2011-05-18T10:35:00	29.8	NORMAL	Not allocable!	10.0 - 50.0	2.1	Y
	Day 2	24H	2011-05-19T08:29:00	27.0	NORMAL	Not allocable!	10.0 - 50.0	-0.7	Y
	Day 4	72H	2011-05-21T08:30:00	27.0	NORMAL	Not allocable!	10.0 - 50.0	-0.7	Y
	Follow-up		2011-05-26T08:42:00	28.8	NORMAL	Not allocable!	10.0 - 50.0	1.1	Y
Albumin [g/L]	Screening		2011-05-12T07:16:00	48.1	NORMAL	Not allocable!	35.0 - 52.0		Y
	Day -1		2011-05-17T08:47:00	44.1	NORMAL	Not allocable!	35.0 - 52.0		Y
	Day 1	2H	2011-05-18T10:35:00	50.2	NORMAL	Not allocable!	35.0 - 52.0	6.1	Y
	Day 2	24H	2011-05-19T08:29:00	47.9	NORMAL	Not allocable!	35.0 - 52.0	3.8	Y
	Day 4	72H	2011-05-21T08:30:00	45.6	NORMAL	Not allocable!	35.0 - 52.0	1.5	Y
	Follow-up		2011-05-26T08:42:00	47.7	NORMAL	Not allocable!	35.0 - 52.0	3.6	Y

Figure 8. LAB Listing Output from PROC REPORT

Lab Listing of CHEMISTRY Category

	Planned Time Point Name	Date/Time of Specimen Collection	Result or Finding in Original Units	Reference Range Indicator	Abnormal clinical significance flag	Normal Range	Change from Baseline	Fasting Status
⊞ 1001/C1 (EPHPO3 10 mg)- 45 Years /M /WHITE/194 cm/86.0 kg							.	
⊞ 1002/C1 (EPHPO3 Placebo)- 42 Years /M /WHITE/169 cm/63.6 kg							.	
⊞ 1003/C1 (EPHPO3 10 mg)- 28 Years /M /WHITE/178 cm/74.7 kg							.	
⊞ 1004/C2 (EPHPO3 10 mg)- 43 Years /M /WHITE/179 cm/86.3 kg							.	
⊞ 1005/C2 (EPHPO3 10 mg)- 47 Years /M /WHITE/185 cm/75.0 kg							.	
⊞ 1006/C2 (EPHPO3 Placebo)- 47 Years /M /WHITE/174 cm/62.9 kg							.	

Figure 9. LAB Listing Output from PROC STREAM - With Collapsed Columns

Lab Listing of CHEMISTRY Category

	Planned Time Point Name	Date/Time of Specimen Collection	Result or Finding in Original Units	Reference Range Indicator	Abnormal clinical significance flag	Normal Range	Change from Baseline	Fasting Status
⊟ 1001/C1 (EPHPO3 10 mg)- 45 Years /M /WHITE/194 cm/86.0 kg							.	
Albumin [g/L]							.	
Screening		2011-05-12T07:16:00	48.1	NORMAL	Not allocable!	35.0 - 52.0	.	Y
Day -1		2011-05-17T08:47:00	44.1	NORMAL	Not allocable!	35.0 - 52.0	.	Y
Day 1	2H	2011-05-18T10:35:00	50.2	NORMAL	Not allocable!	35.0 - 52.0	6.1	Y
Day 2	24H	2011-05-19T08:29:00	47.9	NORMAL	Not allocable!	35.0 - 52.0	3.79999999999999	Y
Day 4	72H	2011-05-21T08:30:00	45.6	NORMAL	Not allocable!	35.0 - 52.0	1.5	Y
Follow-up		2011-05-26T08:42:00	47.7	NORMAL	Not allocable!	35.0 - 52.0	3.6	Y

Figure 10. LAB Listing Output from PROC STREAM

CONCLUSION

The main intention of the paper is to uncomplicate the validation/reviewing process by incorporating the PROC STREAM. 3-4-digit (100 – 10000) pages of report can reviewed in single digit page numbers(one page) by PROC STREAM. The validator/reviewer can use the PROC STREAM procedure in the place of PROC REPORT, with which we can compare the counts as well as check the cosmetic of the layout. The paper concentrates on creating a single step node.

REFERENCES

- Don Henderson, PROC STREAM and SAS® Server Pages: Generating Custom User Interfaces. Paper 1737-2014.
- Don Henderson, PROC STREAM and SAS® Server Pages: Generating Custom HTML Reports. Paper 1738-2014.
- Joseph Hinson, Proc STREAM: The Perfect Tool For Creating Patient Narratives. PharmaSUG 2015 - Paper AD03
- Joseph Hinson, The New STREAM Procedure as a Virtual Medical Writer. PharmaSUG 2015 - Paper AD03

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to the management of our organization – “Ephicacy” for the encouragement and support in helping us with all the necessary facilities to write this paper.

We would also like to thank members of the Ephicacy family for their encouragement, insightful comments and hard questions.

Our sincere thanks to our Director Mr. Tyagrajan Swaminathan and our India center head Mr. Siva Ramamoorthy for the unwavered support.

CONTACT INFORMATION

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

Author Name: Charumathy Sreeraman
Company: Ephicacy Lifescience Analytics
Address: 2nd Main Rd, Sarvobhogam Nagar, Arekere.
City / Postcode: Bengaluru, Karnataka 560076
Work Phone: 080 4146 3195/+91 9620209942
Email: charumathy.sreeraman@ephicacy.in
Web: http://www.ephicacy.com/

Author Name: Samundeeswari Raja
Company: Ephicacy Lifescience Analytics
Address: 2nd Main Rd, Sarvobhogam Nagar, Arekere.
City / Postcode: Bengaluru, Karnataka 560076
Work Phone: 080 4146 3195/+91 8903689288
Email: samundeeswari.raja@ephicacy.in
Web: http://www.ephicacy.com/

APPENDIX

MACRO TO CREATE THE NODES FOR THE HTML OUTPUT

```
%macro treeNodes
      (data = ,                    /* name of input data set that contains the tree */
       nestingLevel = nestingLevel,/* variable name that defines the level of the
branch */
       levelText = levelText, /* variable name that contains the text to label the
node */
       columns =,                  /* optional list of columns to display */
       columnWidths =,              /* width (in pixels) for the columns */
       textWidth = 200,             /* width (in pixels) for the levelText column */
       divPrefix = R,               /* a prefix to define unique ID values */
       headers = Y,                 /* display the column names/labels? blank = No */
       expandedTo = 1,   /* level the tree should be expanded to on initial display */
       tableAttributes = cellspacing="0" cellpadding="0" border = 0
                                    /* attributes for the HTML table tag */
      );

  <script language="javascript">
 function toggleDiv(divid) {
  if(document.getElementById(divid).style.display == 'none'){
  document.getElementById(divid).style.display = 'block';
  }
  else {
  document.getElementById(divid).style.display = 'none';
  }
  if(document.getElementById(divid+'_M').style.display == 'none') {
  document.getElementById(divid+'_M').style.display = 'block';
  document.getElementById(divid+'_P').style.display = 'none';
  }
  else {
  document.getElementById(divid+'_M').style.display = 'none';
  document.getElementById(divid+'_P').style.display = 'block';
  }
 }
 </script>

 %local dsid dsidNext nestingLevelid levelTextid l v nextl var nColumns i rows rc
rowNum lastWidth;

 &streamDelim newline;%let plusImage = %str(

MDAwAAAAAAAAAAAP///yH5BAEAAAsALAAAAAALAAsAAAQecMlJJbgY27oXmML0eaC4CShKrkvYja3JUmNmc1w
EADs=
);
 &streamDelim newline;%let minusImage = %str(

MDAwAAAAAAAAAAAP///yH5BAEAAAsALAAAAAALAAsAAAQbcMlJJbgY27oX4F5XfaFgmmE6ihRJrl6WgVUEADs
=
);
 %let imgWidth = 20;

 %let dsid = %sysfunc(open(&data));
 %let dsidNext = %sysfunc(open(&data));
 %let nestingLevelid = %sysfunc(varnum(&dsid,&nestingLevel));
 %let levelTextid = %sysfunc(varnum(&dsid,&levelText));

 /* determine maximum depth and number of rows */
 %let deep = 0;
 %let rows = 0;
 %do %while(%sysfunc(fetch(&dsid)) = 0);
    %let l = %sysfunc(getvarn(&dsid,&nestingLevelid));
    %if &deep < &l %then %let deep = &l;
    %let rows = %eval(&rows+1);
 %end;

 /* close and re-open to read from beginning */
 %let dsid = %sysfunc(close(&dsid));
```

```
 %let dsid = %sysfunc(open(&data));

 /* determine columns */
 %let i = 1;
 %let var = %scan(&columns,&i,%str( ));
 %let lastWidth = &textWidth;
 %do %while(%length(&var) gt 0);
     %local vnum&i vtype&i vfmt&i width&i vlabel&i;
     %let vnum&i = %sysfunc(varnum(&dsid,&var));
     %let vtype&i = %sysfunc(vartype(&dsid,&&vnum&i));
     %let vfmt&i = %sysfunc(varformat(&dsid,&&vnum&i));
     %let vlabel&i = %qsysfunc(varlabel(&dsid,&&vnum&i));
     %let vlabel&i = %qsysfunc(coalesceC(&&vlabel&i,&var));
     %let width&i = %scan(&columnWidths,&i,%str( ));
     %let width&i = %sysfunc(coalesceC(&&width&i,&lastWidth));
     %let lastWidth = &&width&i;
     %let i = %eval(&i+1);
     %let var = %scan(&columns,&i,%str( ));
 %end;
 %let nColumns = %eval(&i-1);

 %if %length(&headers) gt 0 and %length(&columns) gt 0 %then
 %do;  /* show headers */
    &streamDelim newline;<table &tableAttributes>
    <tr>
    %do i = 0 %to &deep;
       <td width="&imgWidth"> </td>
    %end;
    <td width="&textWidth"> </td>
    %do i = 1 %to &nColumns;
       <th width="&&width&i" align="right">&&vlabel&i</td>
    %end;
    </tr>
    </table>
 %end; /* show headers */

 %do %while(%sysfunc(fetch(&dsid)) = 0);
     %let rowNum = %eval(&rowNum+1);
     %let l = %sysfunc(getvarn(&dsid,&nestingLevelid));

 %if &l lt &expandedTo %then
 %do;  /* initially expanded */
    %let display = block;
    %let plus = none;
    %let minus = block;
 %end; /* initially expanded */
 %else
 %do;  /* initially collapsed */
    %let display = none;
    %let plus = block;
    %let minus = none;
 %end; /* initially collapsed */

     %let v = %sysfunc(getvarc(&dsid,&levelTextid));
     %let rc = %sysfunc(fetchobs(&dsidNext,&rowNum+1));
     %if &rc = 0 %then %let nextl = %sysfunc(getvarn(&dsidNext,&nestingLevelid));
     %else %let nextl = 0;
&streamDelim newline;<table &tableAttributes>
<tr>
     %do i = 0 %to &l-1;
<td width="&imgWidth"> </td>
     %end;
<td width="&imgWidth">
     %if &l lt &nextl %then
     %do;
<a href="javascript:;" onmousedown="toggleDiv('&divPrefix&rowNum');">
&streamDelim newline;<img id="&divPrefix&rowNum._M" src="&plusImage" border="0"
style="display:&plus;" alt="Expand">
&streamDelim newline;<img id="&divPrefix&rowNum._P" src="&minusImage" border="0"
style="display:&minus;" alt="Collapse">
```

```
&streamDelim newline;
     %end;
     %else %str( );
</td>
<td width="&textWidth">&v</td>
     %do i = &l+1 %to &deep;
<td width="&imgWidth"> </td>
     %end;
     %do i = 1 %to &nColumns;
<td width="&&width&i"
align="right">%sysfunc(getvar&&vtype&i(&dsid,&&vnum&i),&&vfmt&i)</td>
     %end;
</tr>
</table>
%if &l lt &expandedTo %then %let display = block;
%else %let display = none;
<div id="&divPrefix&rowNum" style="display:&display">
     %do i = &nextl %to &l;
</div>
     %end;

 %end;

 %let dsid = %sysfunc(close(&dsid));
 %let dsidNext = %sysfunc(close(&dsidNext));
%mend treeNodes;


%let pageTitle = ;
%let data =work.final;
%let class=;
%let vars=;
```