Paper 2033-2014

# Creating Journal Ready Tables with Special Characters Using ODS LaTeX

**Steven Feder, Federal Reserve Board of Governors, Washington, DC**

## ABSTRACT

LaTeX is a free document creation package often used to create journal articles. It provides the capability to create very specific formatting and to write a wide variety of formulas. Using ODS, SAS® can write documents to a LaTeX file, which can then be compiled through LaTeX into PDF files. In this article we briefly review the basic syntax and options to produce these files. Then we look at how to create a new tagset to make changes to the standard ODS LaTeX templates to create the non-gridded table appearance typically seen in journal articles. We also explore how to write special characters and equations not otherwise available through ODS LaTeX.

## ODS LATEX BASICS

There are four LaTeX variations:

```
ods tagsets.latex;
ods tagsets.colorlatex;
ods tagsets.tablesonlylatex;
ods tagsets.simplelatex;
```

For the full syntax reference, see http://support.sas.com/rnd/base/ods/odsmarkup/latex.html. We will focus on modifying the LATEX tagset, but the modification process would apply to the others as well.

## TABLE GRID FIX

With any of the ODS LATEX variations, the table grid can be a problem. The default is a full grid, with all of the internal horizontal and vertical lines. Consider a simple example:

```
data t;
length v1 $20;
input v1 $ v2;
datalines;
f=ma 21
f=mba 24
;

run;

ods tagsets.tablesonlylatex file="t1.tex";
proc print data=t;
run;
ods tagsets.tablesonlylatex close;
```

This produces fully gridded output. (See Appendix 1 for the the LaTeX output compiled into a PDF file, which was appended to the PDF file saved from Word to produce this article.)

The style setting RULES=NONE should remove the grid lines, as it does in other ODS destinations such as PDF and RTF. Unfortunately, setting the style for rules using code such as:

```
proc print data=t style(table)={rules=none};
run;
```

does not remove the grid lines in ODS Latex output. It is possible, however, to produce the non-gridded output by creating a new tagset eliminating the unwanted LaTeX grid line codes. See the full code of the SAS tagset fix in Appendix 2 and the compiled output in Appendix 3. (The output is a PDF file compiled from the LaTeX code file SAS produces.) This tagset fix was

provided by SAS Support, with some minor tweaking by this author. The resulting table is now in the ungridded format required by many journals and ready to be included into a document for publication.

The text of the uncompiled LaTeX file produced by this grid fix helps explain how LaTeX and SAS work together. Ignoring a long section of definitions and other LaTeX business, the actual SAS table output does not begin until very near the bottom of the file, enclosed in the longtable codes:

```
\begin{longtable}{rlr}\hline
   Obs &    v1 &    v2\\
\endhead
   1 &    f=ma &    21\\
   2 &    sqrt(abd) &    22\\
\hline \end{longtable}
```

The text “rlr” inside the braces on the first line defines the columns without vertical grid lines. The text “\\” at the end of each data line produces a new line without horizontal grid lines. In the code below, adding the “|” character separating the aligned columns in the longtable braces adds vertical grid lines. Adding the text “hline” to each line produces the horizontal grid lines. When each data line ends in a horizontal separator, there is no need to add an additional horizontal separator before the end of the table.

```
\begin{longtable}{|r|l|r|}\hline
   Obs &    v1 &    v2\\\hline
\endhead
   1 &    f=ma &    21\\\hline
   2 &    sqrt(abd) &    22\\\hline
\end{longtable}
```

Changes to a few events in the ODS LaTeX tagset code produce the modifications above. The events COLSPEC_ENTRY and COLSPECS define the columns. To add the vertical grid lines in the LaTeX code, add the vertical bar to these ODS LATEX TAGSET events:

```
define event colspec_entry;
   put just /if ^cmp( just, "d");
   put "r" /if cmp( just, "d");
end;
define event colspecs;
   start:
      put "{";
   finish:
      put "}\hline" NL;
end;
```

with this as the new code:

```
/* add the vertical lines generally */
define event colspec_entry;
   put just "|" /if ^cmp( just, "d");
   put "r|" /if cmp( just, "d");
end;
/* add the far left table border*/
define event colspecs;
   start:
      put "{|";
   finish:
      put "}\hline" NL;
end;
```

To add the horizontal grid lines, add HLINE to the ROW event to produce the separator after each row. The TABLE event, however, already by default produces a horizontal line as the border at the bottom of the table, so remove the HLINES from the TABLE event, so as not to print two horizontal lines. This code:

```
define event row;
   finish:
      put "\\" NL;
end;

/* addif putting in horizontal lines for each row, do not end table with line */
define event table;
   start:
      put NL;
      put NL;
      put "\begin{longtable}";
   finish:
      put "\hline \end{longtable}" NL;
      put NL;
end;
```

becomes this:

```
/* add horizontal lines after each row*/
define event row;
   finish:
      put "\\\hline" NL;
end;

define event table;
   start:
      put NL;
      put NL;
      put "\begin{longtable}";
   finish:
      put "\end{longtable}" NL;
      put NL;
end;
```

The small code segments above do not represent the complexity of the LaTeX tagset. Even understanding just the full code of the tagset fix in Appendix 2 would be well outside of the focus of this paper. The code above, however, does illustrate how to develop a modification to the LaTeX tagset by first identifying the LaTeX codes that produce the desired result, and then perusing and test modifying the tagset to determine what events control those codes.

## INCORPORATING A LATEX TABLE INTO A LATEX DOCUMENT

One way to bring ODS LATEX output into a document is to cut and paste the generated LONGTABLE code into a larger document, resulting in a simple, complete document such as this:

```
\documentclass[]{article}
\RequirePackage{longtable}
%opening
\title{}
\author{}
\begin{document}
\maketitle
\begin{abstract}
\end{abstract}
\section{}
Here is line of text that precedes the table.
\begin{longtable}{|r|l|r|}\hline
   Obs &    v1 &    v2\\\hline
```

```
\endhead
   1 &    f=ma &    21\\\hline
   2 &    sqrt(abd) &    22\\\hline
\end{longtable}
\end{document}
```

This exercise in creating a simple document illustrates how little of what ODS LATEX writes is necessary to any specific document.

## WORD VS. LATEX

Microsoft Word can produce special characters and equations using the Microsoft Equation Editor within Word (but Word cannot save the equations to a PDF). It is outside the scope of this paper to compare the Word and LaTeX capabilities and results. We will assume that the user has already made the decision for whatever the reason to use LaTeX. We will look only at how to print special characters and equations into SAS output to the LaTeX destination. It should be noted, however, that printing special characters and equations into SAS output to the RTF destination, and thereby to Word, would present related issues. We also assume that the user needs the features and exacting print requirements we discuss below. We recommend that the casual user consider the options carefully before deciding to invest in learning to use LaTeX.

## SPECIAL CHARACTERS

It is relatively easy to insert special characters into SAS output. The easiest way available on Windows is to use the Windows accessory Character Map. To start Character Map:

Click **Start→All Programs→Accessories→System Tools→Character Map**.

Then, select the desired characer and copy and paste the character directly into data or code.

It is, however, only slightly more effort to use the four digit unicode to print the character. The unicodes are also available from Character Map. Put the code in quotes, followed by the letter 'x':

```
data t1;
  v1='00B3'x;
  put v1=;
  text='The value is X'||v1;
  put text=;
run;
```

The special characters available are of course limited to those with a unicode. The set of unicode characters does cover many if not all of the characters users typically need. There are obvious exceptions, though. There are unicodes for "squared," i.e., "X²" and "cubed," i.e., "X³." but there is no unicode for any power higher than three. Unicode provides symbols for only the fractions with 2, 3, 4, and 8 as the denominator. And Unicode does not provide a way to produce the extended lines and so forth that can be seen in equations.

## EQUATIONS

It is not feasible to produce more complex characters or equations directly in SAS. For example, the square root sign in unicode,"√," can not be encoded with special characters to enclose its argument. And properly scaled and precisely written numerator lines and so forth are impossible or very difficult.

LaTeX, however, can produce all of these with very fine control. There is a difficulty, however, in coding into SAS the codes that need to reach LaTeX. LaTeX can produce the special characters, and SAS can produce LaTeX tables, but SAS cannot directly produce tables with LaTeX special characters, because there is no way to pass the LaTeX special characters into the tables. First, the lack of a unicode means it is not possible to code the special character directly into a SAS program or data. It would seem possible to write the LaTeX code for the special characters into the document as text which LaTeX would interpret into special characters, but ODS LATEX interprets the parts of the LaTeX code as text characters and specifically codes them so that they will *not* be interpreted as special characters. Perhaps this is a deficiency which also calls for an eventual enhancement of the LaTeX tagset, but until then we must work around this.

## WORK AROUND TO PRODUCE LATEX SPECIAL CHARACTERS IN SAS

To produce LaTeX codes embedded in SAS tables, we will first embed a dummy code to produce the square root sign in our data:

```
data t;
length v1 $20;
input v1 $ v2;
datalines;
f=ma 21
sqrt(abd) 22
;
run;
```

The text “sqrt” will later be used to find and enclose in a square root sign the text within the parentheses in the data. ODS will first write the table to a LaTeX file with the parentheses as above. Then we will read the table with simple data step code, translate the dummy parentheses code into the LaTeX square root code, and rewrite the LaTeX file, which is really only a flat file until compiled:

```
ods tagsets.test file="t1.tex";

proc print data=t;
run;

ods tagsets.test close;

data re1;
  infile "t1.tex" pad missover;
  file "t1_out.tex";
  length c1 $200;
  input c1 $ 1-200;
  p1=index(c1,'sqrt');
  if p1>0 then do;
    c2=substr(c1,(p1+5));
       p2=index(c2,')');
    c3=substr(c1,1,(p1-1))||'$\sqrt{'||scan(c2,1,')')    ||'}$'||substr(c2,(p2+1));
    put c3;
  end;
  else put c1;
run;
```

See the output in Appendix 4.

## CONCLUSION

Using ODS LATEX can extend the capabilities of SAS into areas of journal formatting and typesetting otherwise impossible. The basic syntax is simple and will be all most people need. For those who need to go further into LaTeX Math, a relatively painless work around allows printing of any LaTeX coded special character in a SAS generated table.

## ACKNOWLEDGMENTS

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.

## CONTACT INFORMATION

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

Steven Feder
Federal Reserve Board
20th & C Streeets
Washington, DC 20551
202-452-3144
Steven.h.feder@frb.gov

## *Appendix 1*

## *Table Compiled from LaTeX with Grid*

| Obs | v1 | v2 |
|---|---|---|
| **1** | f=ma | 21 |
| **2** | sqrt(abd) | 22 |

## *Appendix 2*

```
proc template;
   define tagset Tagsets.event1;
      define event colspec_entry;
         put just "|" /if ^cmp( just, "d");
         put "r|" /if cmp( just, "d");
      end;
      define event table;
         start:
            put NL;
            put NL;
            put "\begin{longtable}";
         finish:
            put "\hline \end{longtable}" NL;
            put NL;
      end;
      define event stacked_cell;
         start:
            put NL;
            put "\begin{tabular}";
            trigger alignment;
         finish:
            put "\end{tabular}" NL;
      end;
      define event colspecs;
         start:
            put "{";
         finish:
            put "}\hline" NL;
      end;
      define event colspec_entry;
         put just /if ^cmp( just, "d");
         put "r" /if cmp( just, "d");
      end;
      define event row;
         finish:
            put "\\" NL;
      end;
      define event header;
         start:
            trigger data;
         finish:
            trigger data;
      end;
      define event data;
         start:
            put VALUE /if cmp( $sascaption, "true");
            break /if cmp( $sascaption, "true");
            put %nrstr(" & ") /if ^cmp( COLSTART, "1");
            put "   ";
            unset $colspan;
            set $colspan colspan;
            do /if exists( $colspan) | exists ( $cell_align );
               put "\multicolumn{";
               put colspan /if $colspan;
               put "1" /if ^$colspan;
               put "}{";
               put "|" /if ^$instacked;
               put just;
               put "|" /if ^$instacked;
               put "}{";
            done;
            put tranwrd(VALUE,"-","$-$") /if contains( HTMLCLASS, "data");
```

```
            put VALUE /if ^contains( HTMLCLASS, "data");
         finish:
            break /if cmp( $sascaption, "true");
            put "}" /if exists( $colspan) | exists ( $cell_align );
      end;
      define event rowspanfillsep;
         put %nrstr(" & ");
      end;
      define event rowspancolspanfill;
         put "   ~";
      end;
      define event image;
         put "\includegraphics{";
         put BASENAME /if ^exists( NOBASE);
         put URL;
         put "}" NL;
      end;
      parent = tagsets.latex;
   end;
run;
```

## *Appendix 3*

## *Table Compiled from LaTeX Without Grid*

| Obs | v1 | v2 |
|---|---|---|
| 1 | f=ma | 21 |
| 2 | sqrt(abd) | 22 |

## *Appendix 4*

## *Table Compiled from LaTeX with Special Character*

| Obs | v1 | v2 |
|---|---|---|
| 1 | f=ma | 21 |
| 2 | $\sqrt{abd}$ | 22 |