ruled.tex

TEX macros for making ruled tables

Eric Myers

Department of Physics and Astronomy
Vassar College
Poughkeepsie, New York 12601 USA

and

Frank E. Paige

Physics Department
Brookhaven National Laboratory
Upton, New York 11973 USA

While `\halign` in Plain TEX is convenient for making simple tables, it is still difficult to make tables with horizontal and vertical rules. Indeed, typesetting ruled tables in TEX has often been likened to programming a computer in assembler language. Even *The TEXbook* acknowledges that "people who know how to make ruled tables are generally known as TEX Masters."[1] The macros described below take a great deal of the difficulty out of making ruled tables, so that anybody can become a TEX Master, or at least a master of ruled tables.

The basic idea behind making a ruled table is similar to `\halign`, and the syntax is the same as the Princeton table macros known as `TABLES.TEX` by Ray Cowan,[2] although the `ruled.tex` macros are quite different internally and run about 5 times faster. You begin the table material with `\ruledtable` and end it with `\endruledtable`. The table material is a list of items to appear in the rows and columns of the tables. Just as with `\halign` you can separate items in different columns with `&` and end a row with `\cr`. The `&` alignment character separates columns without putting a vertical rule between the columns, while `\cr` ends a line and puts a thin horizontal rule under it. To get a vertical rule between columns replace the `&` with a '`|`' (the vertical line), and to end a line without a horizontal rule under it replace the `\cr` with `\crnorule`. One important thing to remember is to end the last line in the table with `\endruledtable` instead of `\cr`!

As a first example, here is a simple ruled table,

Year	World Population
8000 B.C.	5000000
50 B.C.	200000000
1650 A.D.	500000000
1850 A.D.	1000000000
1945 A.D.	2300000000
1980 A.D.	4400000000

and here are the instructions that created it,

```
\ruledtable
Year      |    World Population \cr
8000 B.C. |   ~~~5000000        \crnorule
~~50 B.C. |   ~200000000        \crnorule
1650 A.D. |   ~500000000        \crnorule
1850 A.D. |   1000000000        \crnorule
1945 A.D. |   2300000000        \crnorule
1980 A.D. |   4400000000
\endruledtable
```

This example also appears in *The TEXbook*, where it is typeset using `\halign`. Compare how the two are constructed and see which way you think is easier.

By default each item in the table is centered in its column. Inside a table the tie character "`~`" takes up the space of exactly one digit (it is the same as saying `\phantom{0}`), and this has been used to line up the digits of the entries in both columns. It is also possible to change the definition of the `\TableItem` macro so that each entry in each column is left or right justified. This is described a little bit later.

Something to note about this example table is that the outer rules are thicker than the others. You can also get a thick vertical rule between columns by using "`\|`" in place of "`|`", and you can get a thick horizontal rule below a row by replacing the `\cr` with `\crthick`. The widths of thick and thin rules are controlled by the dimensions `\thicksize` and `\thinsize`. If you don't want thick rules at all simply say `\thicksize=\thinsize`, which makes the thick rules as thin as the thin rules.

Some synonyms: `\CR` is the same as `\crthick`, `\nr` is the same as `\crnorule`, and `\crrule` is the same as `\cr` (actually it's the other way around!). `\tab` is the same as `&` — it separates columns without a vertical rule. `\vb` is the same as `|`, it separates columns with a thin vertical bar, while `\Vb` is the same as `\|` and separates the columns with a thick vertical bar. You can also use `\dbl` to separate columns with a double vertical rule.

As in Plain TEX you can use `\span` in place of `&` (or the other column separators) to join two columns together into one, and you can use `\omit` as the very first token in a column to tell it to omit the fancy column layout (the centering or other spacing). You can also use `\multispan` to `\span` and `\omit` several columns, just as in an `\halign`. The only difference is that the number of columns you tell it to `\multispan` is the number

Table 1: Another example of a ruled table.

AT&T Common Stock		
Year	Price	Dividend
1971	41-54	$2.60
2	41-54	$2.70
3	46-55	$2.87
4	40-53	$3.24
5	45-52	$3.40
6	51-59	.95

of columns of the table it spans — in a ruled table constructed with `\halign` you would count the columns *and* any vertical rules.

Table 1 is another sample table, again taken from *The TEXbook*. Something important to notice is that the last ruled table example didn't have a table number or a caption, while this one does. That's because the last example was created using just `\ruledtable`, while this example is inside of `\table`. The point is that `\table` and `\ruledtable` do separate things and can be used either separately or together; `\table` creates the space for the table and gives it a caption and a number, while `\ruledtable` actually constructs the body of the ruled table. The instructions that created Table 1 are:

```
\table{example-II}
\caption{Another example of a ruled table.}
\singlespaced
\ruledtable
\multispan3\hfil AT\&T Common Stock \hfil\CR
Year \dbl Price | Dividend \cr
1971 \dbl 41-54 | \$2.60   \cr
~~~2 \dbl 41-54 | \$2.70   \cr
~~~3 \dbl 46-55 | \$2.87   \cr
~~~4 \dbl 40-53 | \$3.24   \cr
~~~5 \dbl 45-52 | \$3.40   \cr
~~~6 \dbl 51-59 |  ~~.95
\endruledtable
\endtable
```

Sometimes it is desirable to have rules across only some of the columns of a table. In this case put `\crule` in the columns that are to have rules, `\cskip` in the columns where no rules are desired, and end the line with `\crpart`, which signals that the line contains a partial rule. The column headings in Table 2 are an example of partial rules in a table. That table was created with the commands,[3]

Table 2: Parameters used in the MS-Xα calculations for the chlorosilane molecules, H_xSiCl_{4-x}; x=0–4.

Molecule	Region	Radius (au)	α_{HF}	l_{max}	
				initial state	final state
$SiCl_4$	outersphere	6.21	0.7238	4	7
	Si	2.20	0.7275	2	3
	Cl	2.40	0.7233	2	3
$SiHCl_3$	outersphere	6.21	0.7260	4	7
	Si	2.16	0.7275	2	3
	Cl	2.40	0.7233	2	3
	H	1.20	0.7772	1	2
SiH_2Cl_2	outersphere	6.25	0.7295	4	7
	Si	2.20	0.7275	2	3
	Cl	2.40	0.7233	2	3
	H	1.20	0.7772	1	2
SiH_3Cl	outersphere	6.30	0.7360	4	7
	Si	2.20	0.7275	2	3
	Cl	2.40	0.7233	2	3
SiH_4	outersphere	4.10	0.7524	4	7
	Si	2.04	0.7275	2	3
	H	1.20	0.7772	1	2

```
\table{SiHCl}
\caption{...}
\singlespaced\tenpoint

\ruledtable
\vctr{Molecule} | \vctr{Region} | Radius |
  \vctr{$\alpha _{\rm HF}$}|\multispan{2} $l_{\rm max}$
\crnorule
\cskip | \cskip | \cskip |\cskip|    \crule     |  \crule \crpart
       |        | (au)   |      | initial state | final state \CR

SiCl$_4$      | outersphere | 6.21 | 0.7238 | 4 | 7 \crnorule
              | Si          | 2.20 | 0.7275 | 2 | 3 \crnorule
              | Cl          | 2.40 | 0.7233 | 2 | 3 ...

\endruledtable
\endtable
```

Ruled tables are normally centered on the page, but you can turn off the centering with `\noncenteredtables`. Then the table becomes just another piece to add to a horizontal list. One thing this lets you do is have two tables side by side on the page, like so:

Item ABC	Item DEF
Item GHI	Item JKL

Data 111	Data 222
Data 333	Data 444
Data 555	Data 666

This pair of tables was produced by saying:

```
\noncenteredtables
\medskip
\line{
\ruledtable
Item ABC | Item DEF \cr
Item GHI | Item JKL \endruledtable
\hfil
\ruledtable
Data 111 | Data 222 \cr
Data 333 | Data 444 \cr
Data 555 | Data 666 \endruledtable}
\smallskip
```

Saying `\centeredtables` turns table centering back on. Tables are then centered across the page, and a table is then added to a *vertical* list.

Usually `\ruledtable` makes tables as wide as their natural width, but it is possible to make them wider by setting the dimension `\tablewidth` to the width desired before saying `\ruledtable`. We can make the same table in Table 1 but with the width set to 10cm:

AT&T Common Stock		
Year	Price	Dividend
1971	41-54	$2.60
2	41-54	$2.70
3	46-55	$2.87
4	40-53	$3.24
5	45-52	$3.40
6	51-59	.95*

* (first quarter only)

The careful reader will notice that the version of Table 1 in *The TEXbook* had a footnote to the last entry, and we have added that here. To do so we had to put the footnote in an `\hbox` and stack it under the table, so we had to turn off table centering. The table is therefore a bit more complicated to make, but not much:

```
\noncenteredtables
\tablewidth=10cm
\line{\hfil\vbox{\singlespaced
\ruledtable
\multispan3\hfill AT\&T Common Stock \hfill\CR
Year \dbl Price | Dividend \cr
1971 \dbl 41-54 | \$2.60   \cr
~~~2 \dbl 41-54 | \$2.70   \cr
~~~3 \dbl 46-55 | \$2.87   \cr
~~~4 \dbl 40-53 | \$3.24   \cr
~~~5 \dbl 45-52 | \$3.40   \cr
~~~6 \dbl 51-59 |  ~~.95\rlap*
\endruledtable
\hbox{* (first quarter only)}
\vss}\hfil}
```

If we had not wanted to add the footnote, then the table would have been typed the same way as Table 1 except for setting the `\tablewidth`.

Important note: `\tablewidth` is turned “off” after the table is made, so the next table will have its natural width unless you explicitly set `\tablewidth` again!

`\ruledtable` normally inserts a `\quad` of space around each item. If you have a very wide table, especially one with many columns, you may wish to reduce this. `\TightTables` make this spacing a single space, the minimum acceptable. `\LooseTables` restores the default spacing. You can define this space to be something else by re-defining `\tablespace`, for example as

```
\def\tablespace{\ \ }
```

If the table is still too wide with `\TightTables`, you will probably have to reformat it or print it in landscape mode.

The careful reader may also have noticed that in the examples in Table 1 and Table 2 we said `\singlespaced`. That’s because the spacing of the rows of a table can be controlled just like the spacing of the text with `\singlespaced`, `\doublespaced`, etc. The space between the rows of a table is set by a strut called `\tstrut`, which holds the rows apart. Its default definition is

```
\def\tstrut{\vrule height 3.1ex depth 1.2ex width 0pt}
```

You can change the spacing between rows by changing the definition of this strut. One of the things `\singlespaced` does is define the `\tstrut` to be 0.5ex higher and deeper than a parenthesis. `\doublespaced`, `\triplespaced`, and `\widenspacing` all increase the height and depth of the strut in proportion to the increase in the baseline spacing.

If you put something in a table which is larger than the strut, then it will touch the rules above or below. You can fix this by inserting your own strut into that row. More simply, you can say `\bigitem{`⟨stuff⟩`}`, and “stuff” will be positioned correctly with a nice amount of space above and below.

Table 3a: Basic table macros for use with `\ruledtable`.

<table>
<tr><th><i>Macro name</i></th><th><i>Description</i></th></tr>
<tr><td><code>\ruledtable</code></td><td>Starts a ruled table.</td></tr>
<tr><td><code>|</code> (vertical bar)
or <code>\vb</code></td><td>Separates one column from the next, with a a thin vertical rule between them.</td></tr>
<tr><td><code>&</code> or <code>\tab</code></td><td>Separates one column from the next, but with no rule between them.</td></tr>
<tr><td><code>\cr</code></td><td>Ends the current row, and starts the next one. The completed row will be separated from the next with a thin horizontal rule.</td></tr>
<tr><td><code>\crthick</code>
or <code>\CR</code></td><td>Similar to <code>\cr</code>, but the rows will be separated with a thick horizontal rule.</td></tr>
<tr><td><code>\crnorule</code>
or <code>\nr</code></td><td>Similar to <code>\cr</code>, but the rows will not be separated by a rule.</td></tr>
<tr><td><code>\|</code></td><td>Same as <code>|</code>, but with a thick vertical rule.</td></tr>
<tr><td><code>\dbl</code></td><td>Same as <code>|</code>, but with two thin vertical rules.</td></tr>
<tr><td><code>\endruledtable</code></td><td>Ends the ruled table.</td></tr>
</table>

Continued...

You can also decide whether or not you want diagnostic messages printed when a table is created. Saying `\tableinfotrue` causes a message to be printed on your terminal every time a ruled table is created. The message tells you how many rows and columns were in the table. This is useful information, because it can tell you immediately if you made a mistake in the table or left out a row or column. `\tableinfotrue` is the default, but you can turn off these messages by saying `\tableinfofalse`. In any case, you will be warned if a row has a different number of columns from the previous one.

The tables we have demonstrated so far all contain numbers, but sometimes it is desirable to have an entry in a table which is a paragraph of text. For that purpose, you can use `\para{`⟨text⟩`}`. The width of the paragraph is determined by the dimension `\parasize`, which defaults to 4 inches.

Tables 3a and 3b on the next two pages summarize the most important commands for making ruled tables.

Table 3b: Additional macros for use with \ruledtable.

Continued...

Macro name	*Description*
\multispan{n}	Makes the next entry span the next n columns, where n is an integer, $n > 0$. See other notes on \multispan below.
\omit	This TEX primitive causes the normal template for its entry to be omitted, allowing the user to do something else with this entry.
\para{⟨text⟩}	Formats *text* into a neat little paragraph like this one. The width of the paragraph produced is determined by the dimension \parasize. The default is 4 inches.
\parasize=⟨dimen⟩	Sets the width of paragraphs for the \para macro.
\TableItem{⟨stuff⟩}	Used in the standard template, this macro centers its argument in the column. The user can redefine it for special effects.
\tablespace	The spacing around each item for \TableItem.
\LooseTables \TightTables	Set \tablespace to \quad and \ respectively.
\TrailingSpaces \NoTrailingSpaces	Use trailing spaces and glue items in table entries (default) or ignore them. See text.
\bigitem{⟨stuff⟩}	Use this for something that would normally not fit in one line of a table to make the spacing between rows large enough for this item.
\vctr{⟨stuff⟩}	Centers the item *vertically* so that it can appear between two rows.
\centeredtables	Turns table centering on. Each table will be centered left-to-right on the page. This is the default.
\noncenteredtables	Turns table centering off. Each table is returned as an \hbox, and it is up to the user to place it as desired.
\tableinfotrue \tableinfofalse	Turns on or off the diagnostic message telling you how many rows and columns were found in the table.
\thicksize=⟨dimen⟩	This dimension specifies the thickness of the thick rules in the table. The default size is 1.5 points.
\thinsize=⟨dimen⟩	This dimension specifies the thickness of thin rules in the table. The default size is 0.8 points.
\tablewidth=⟨dimen⟩	Specifies how wide to make the next table. If not specified, the table is made to its natural width. This value is reset following the construction of each table.

Finally we can discuss how you can change how each column of a table is constructed. The macros which build the table take each entry and pass it to a macro called `\TableItem`. The default definition of `\TableItem` is:

```
\let\TableItem=\@TableItem       % default is centered
\def\@TableItem#1{%              % centers item in ruled table
   \hfil\tablespace              % left glue
   #1\killspace                  % item, no space after
   \tablespace\hfil              % right glue
    }%
```

The `\hfil` glue on both sides of the argument causes the item to be centered in the column, while `\tablespace` (which defaults to `\quad`) insures that there will be at least some amount of space between the item and the vertical rules on each side. Because of the way that TEX handles macro arguments, if you type a table with spaces between the text and the next column separator, an extra `\space` will be inserted into the table. If you first say `\NoTrailingSpaces`, then `\killspace` in the above macro is defined to remove any trailing glue items. `\hfill` is redefined by the table macros so that if it is the last token in an entry, then that entry is properly left justified, but to insert any other trailing glue, you must say

⟨text⟩⟨glue⟩`\null`

The default is `\TrailingSpaces`, which does incorporate any trailing spaces into the table but which also treats any trailing glue items normally. You may wish to change this default in your `TXSpatch.tex` file.

You can change the definition of `\TableItem` to suit your own purpose. The simplest modification is to omit the `\hfil` on the left or right so that each entry is right or left justified. This is so commonly used that saying `\RightJustifyTables` or `\LeftJustifyTables` makes the appropriate change for you. Saying `\NoJustifyTables` restores `\TableItem` to its original centering function.

Since `\TableItem` is used to create every column of the table, you cannot use it to modify a single column. To justify just one column you can simply put `\hfill` to the left or right of each item in the column, since `\hfill` is infinitely more stretchy than `\hfil`. (The behavior of glue items other than `\hfill` depends on whether `\NoTrailingSpaces` has been set.) With a little more work it is possible to design special or unusual tables by redefining the "preamble" of the table. This is possible because `\ruledtable` uses `\halign` with a preamble called `\TablePreamble`. The default definition of `\TablePreamble` is:

```
\def\TablePreamble{%                     % \ruledtable preamble
   \TableItem{####}%                     % the first item
   \plaintab\plaintab                    % && means repeat this
   \TableItem{####}%                     % the subsequent items
   \plaincr}%                            % end of preamble
```

The syntax for `\TablePreamble` is like an `\halign` preamble in Plain TEX but with `&` replaced by `\plaintab`, `\cr` replaced by `\plaincr`, and `#` replaced by `####`. If you have read this far and want to try changing `\TablePreable` then you should read the source code and the comments that go with it in the file `TXSruled.tex.`

As a final note we point out that it is also possible for you to create your own column separators with the macro `\nextcolumn`. This macro takes a single argument which is used to separate one column of the table from another. Thus for example, the `\dbl` macro for creating a double vertical rule between columns is defined as:

```
\def\dbl{%
   \nextcolumn{\vrule width\thinsize
   \hskip\thinsize\vrule width\thinsize}}
```

References

1. D.E. Knuth, *The TEXbook* (Addison Wesley, 1986), page 245.
2. Ray F. Cowan, *Making Tables with Macros* (unpublished).
3. J.D. Bozek, Ph.D. thesis (unpublished), Department of Chemistry, University of Western Ontario, London, Ontario CANADA.