

Table Making with INRST \TeX

Michael J. Ferguson
INRS-Télécommunications
Montréal

The table commands described here were developed as an integral part of INRST \TeX . They have also been implemented as a standalone macro package, TABLES. \TeX , to run with the standard Plain macro package that comes with \TeX . This paper was produced using TABLES. \TeX with the standard TUGboat macros (TUGBOT.STY).

The major intent of the table commands is to make the input for tables appear on the editor screen in a reasonable manner. Although this package appears to avoid the basic \backslash halign commands and forms of \backslash cr, $\&$, and $\#$, their insertion in tables is not precluded by the structure of the table commands. To use these basic commands will result in some loss of readability but a gain in flexibility.

Below is shown a now classic table (also used in the *The \TeX book*) for demonstrating the table-making skills of a set of macros. The input which produced the table appears in Figure 1. This table is **not** identical to the one in the *The \TeX book*. It is, however, very close.

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)

This example illustrates the basic structure of a table. A \backslash begintable ... \backslash endtable pair delineates it, and the row format is defined inside the pair \backslash begintableformat ... \backslash endtableformat. The entirely optional *sample row*, immediately after \backslash endtableformat, forces all three columns to the same width, by taking the longest word in the table data, Dividends, and putting it in a *sample command* \backslash sa{.} in each of the three columns. The sample row itself is invisible in the final table. The ~ is a space that is exactly the width of a digit. This permits the data to be lined up in the desired way. The | are separators that insert vertical rules or lines and also allow spaces to be ignored after the data in a column. This makes the columns look

```

\centerline{
\begin{table}
\begin{tableformat}
\center " \center " \center
\end{tableformat}
\br{} \sa{Dividend} "\sa{Dividend} "\sa{Dividend} \er{} %sample line
\~
\br{\:|} \use{3} AT&T Common Stock \er{|}
\~
\br{\:|} Year | Price | Dividend \er{|}
\~
\br{\:|} 1971 | 41--54 | \$2.60 \er{|}
\~
\br{\:|} ~2 | 41--54 | ~2.70 \er{|}
\~
\br{\:|} ~3 | 46--55 | ~2.87 \er{|}
\~
\br{\:|} ~4 | 40--53 | ~3.24 \er{|}
\~
\br{\:|} ~5 | 45--52 | ~3.40 \er{|}
\~
\br{\:|} ~6 | 51--59 | ~.95\rlap* \er{|}
\~
\br{\:}\use{3} \left{* (first quarter only)}
\end{table}

```

Figure 1. Input for table: AT&T Common Stock

nicer and more readable on the input screen. Other details about this table will be apparent later.

The INRST \TeX table commands really use two different types of columns: *rule columns* for insertion (or omission) of vertical rules and *data columns* for the insertion of normal data entries. Generally it is assumed that every pair of data columns is separated by a rule column. In addition, a template for rule columns is provided automatically at the left and right hand sides. The table commands make the insertion or omission of a vertical rule or column as simple as placing a " or a | in the text.

The INRST \TeX table commands have been designed to be most useful when the following conventions are obeyed:

- A rule column is specified in the table format using a ". Successive data columns are separated by a rule column. The actual rule is inserted in the text with a | or omitted with a " rather than being put in the format.
- A rule column is appended to each end of the table format statement. In general this means that there is a rule column on both sides of the table. The actual insertion or not of the rule is very easy.
- Struts for maintaining row height are inserted in each row rather than in the table format.

The commands make it very easy to include struts, and insert or omit rules as needed. In fact it is no more difficult than inserting an & in the appropriate place. The advantage of following this discipline is that insertion of partial horizontal lines (rules) is quite easy, as is the exact vertical placement of items in different data columns.

The basic format for building a table is

```
\beginable
<special table forms>
\beginableformat
<table format>
\endableformat
<first row>
...
<last row>
\endable
```

Each row has the form

```
\br{<..>} <item> | <item> " <item>
... | <item> \er{<..>}
```

where | means to put in a vertical rule and " means to leave it out of the corresponding rule column. $\backslash\text{br}\{..\}$ and $\backslash\text{er}\{..\}$ signify the beginning and end of the row respectively; the optional arguments are intended for rules and/or struts if needed. For instance, $\backslash\text{br}\{\:|\}$ indicates that this line has a

standard strut \backslash : and a standard vrule. $\backslash\text{br}\{\: \text{height } 2\text{pt}\}$ indicates that this row has no strut but that the initial vrule has a height of 2pt. The strut could just as easily have been put in the $\backslash\text{er}\{\:|\}$ instead. The form $\backslash\text{br}\{\}$ or $\backslash\text{br}\{\:"\}$ indicates a row with no initial strut or vrule.

The table commands supply a second flavour of vrule that is user definable. This is specified by the command $\backslash|$, which may be used in exactly the same way as |. The *insides* of the $\backslash|$ are changed by redefining a command $\backslash\text{sprule}$, a *special rule*. The default definition is $\backslash\text{def}\backslash\text{sprule}\{\backslash\text{tvrule}\{2.5\backslash\text{tr}\}\}$ where $\backslash\text{tvrule}\{\langle\text{dimen}\rangle\}$ produces a special centered $\backslash\text{vrule}$ with width $\langle\text{dimen}\rangle$, and $\backslash\text{tr}$ is \TeX default rule thickness, .4pt.

An INRST \TeX table format is a template for the table, and corresponds to the normal *preamble* in an $\backslash\text{halign}$. In fact any valid $\backslash\text{halign}$ preamble may be used in the table format, as long as the # are replaced by ##. However, since a rule column is added to the beginning and end of the *format* list for beginning and terminating rules, a repeating field specification, which is indicated by an (extra) & in the format, should **not** start at a vrule indicator or the result will be two adjacent rules.

Simple column formats are indicated by the use of the commands $\backslash\text{left}$, $\backslash\text{center}$, $\backslash\text{right}$, $\backslash\text{math}$, and $\backslash\text{displaymath}$. The two $\backslash.. \text{math}$ forms are **never** used alone but rather in conjunction with the first three. These forms are **always** separated[†] by ", which allows for, but does not put in, a vertical rule. The simplest table format, and probably the most useful, is

```
\beginableformat
&\center
\endableformat
```

This is a repeating format, as indicated by the & and thus allows for any number of columns. The data in each column is centered.

Another simple table format is

```
\beginableformat
\left " \right " \math\center
\endableformat
```

This table would have three data columns: The first data column is ordinary text, left justified; the second is right justified text; and the last is centered mathematics. Each pair of data columns is separated by a rule column indicated by ". Rule

[†] In fact, any field of the form
 $\&\langle\text{optional stuff}\rangle\#\langle\text{optional stuff}\rangle\&$
 is acceptable.

columns will be added automatically to the left and right hand ends of the specification. The commands `\right`, `\left`, and `\center` insert the appropriate glues and spacing at both sides of the column. The width of this spacing `\tcs`, the *table column separation*, is a `<dimen>`. This may be modified if desired for all tables by putting it in an `\everytable` token list, or for a specific table by placing it in the *special table forms*.

The following is a repeating column specification:

```
\begin{tableformat}
\left " \right " &\math\center
\end{tableformat}
```

This is identical to the previous example except that the data column format and the implicit rule column added by the table commands, `\math\center`, is repeated indefinitely. For almost all tables, the only `&` that should appear in the table format is the one indicating the start of a repeating field.

Horizontal rules are specified by `\-` for the standard `\tr` width rule. An `\hrule` the full width of the table should appear all by itself on a line between an `\er` and the following `\br`, or immediately after the table format. Other `\noalign` commands may be inserted as desired. A partial

horizontal rule is specified with `\use`, a special version of `\multispan`, and the `\-`.

The table command places some restrictions on the automatic insertion of `\tabskip` glue. Hopefully these will be minor. The initial and final `\tabskip` glues are set to `Opt`. This should create no problems. A parameter `\midtabglue` may be modified. This is set just after the first rule column and is turned off at the last rule column. Finally, `\tablespread = {to <dimen>}` will force the table to be `<dimen>` wide, assuming there is enough `\tabskip` glue available.

The instructions in Figure 2 produce the following rather contrived table.

XYZABC					
XYZ			ABC		
X	Y	Z	A	B	C
372.466	493.7	45	124	489	280
372.40	493.7	45	124	489	280
372.	493.7	45	124	489	280
XY/A		832	abc	774	f
		qrr	aaa	799	

This example illustrates the power and flexibility of the table commands. Row *i* shows a *modified*

```
\centerline{
\begin{table}
\def\sprule{\tvrule{5\tr}}
\begin{tableformat}
&\center
\end{tableformat}
\br{\:|} \use{6}          XYZABC      \er{\mst{\:}{3pt}{3pt}\rlap{ \it ** i}}
\br{\:|} \use{3}  XYZ      | \use{3}  ABC      \er{\mst{\:}{2pt}{2pt}}
\br{\:|}      X  | Y  | Z  | A  | B  | C  \er{|}
\br{\:|} 372.466 | 493.7 | 45 | 124 | 489 | 280 \er{|}
\br{\:|} 372.40~ | 493.7 | 45 | 124 | 489 | 280 \er{|}
\br{|}      "   |   "   |   | \use{3}\- \er{|}
\br{\:|} 372.~~~ | 493.7 | 45 | 124 | 489 | 280 \er{\rlap{ \it ** ii}}
\br{\:|} \use{2}      | 832 | abc | 774 | $\int$\er{\mst{$\int$}{Opt}{3pt}}
\br{|} \use{2}\z{XY/A} | \use{4}\- \er{|}
\br{\:|} \use{2}      | qrr | aaa | 799 | \er{\rlap{ \it ** iii}}
\br{\:|}
\end{table}
```

Figure 2.

strut, `\mst`, with changed height and depth (the standard *strut* is 2.5ex high and .9ex deep). `\mst` is used again in row *iii* where some extra space is added at the bottom of the integral sign, but not at the top. `\mst` takes three arguments or parameters: The first is the character from which the *strut* is derived, the second is the additional space on the top of the character and the third is on the bottom. The first two rows of this table had additional space at both the top and bottom while the “integral” row was changed only at the bottom. Fine tuning tables in this way can improve their visual form immensely. Row *iii* demonstrates the use of a `\zb` or *zero-centered box* for putting items in the center of larger boxes. This row with the `\zb` has **no** struts. In addition, `\zb` boxes are centered and have zero height and depth. This means that they take up no vertical space, so that a partial horizontal line or rule may be inserted. Here the `\use{4}\-` tells the table commands to use 4 (data) columns for the horizontal rule.

Many other variations are possible.

Editor’s note: The `INRSTTEX` table commands will be made available through the `TEX` Users Group in both printed form and on magnetic media. Details can be obtained from the TUG office.

Queries

Editor’s note: This is a gentle request to readers who find they can answer any of the queries published in this column. Please send a copy of your answer to the TUGboat editor as well as to the author; all answers will be published in the next issue following their receipt.

WEB System Extensions

I’m working on D. E. Knuth’s “WEB System for Structured Documentation”. If you have any experiences with or suggestions for improving this software development and documentation tool, please let me know about them. I want to extend the WEB system so that it is possible to use other programming languages than Pascal. If anybody out there uses the WEB system with another text formatting

system than `TEX` or if you have made extensions to the system, I would also like to hear about that. I’m very interested in all your experiences, in what you use WEB for, and in your suggestions for improvement.

Helmut Becker
Rittershausstr. 4
D-5300 Bonn, West Germany
+49 228 211850
UNI15C@DBNRHRZ1.Bitnet

Editor’s note: The last issue of TUGboat, 7#1, contained two articles on this subject: R. M. Damerell, “Error detecting changes to Tangle”, pages 22–24, and Wolfgang Appelt and Karin Horn, “Multiple changefiles in WEB”, pages 20–21. At least one attempt to build a WEB for C is known to Helmut Jürgensen, who would also be interested in hearing of work in this area, for possible publication in future issues.

This query was originally disseminated via `TeXhax` and `UNIX-TeX`, and the following responses have been posted.

WEB (`TEX`, `LATEX` etc.) should all conform to ISO-646 (i.e. ASCII) in the following way: In ISO-646 the codes 64, 91–96 and 123–126 are reserved for “national or application oriented use”. Many languages use more letters than a–z, these letters are present on keyboards, and the codes above are used for their representation. Every application (e.g. WEB) should be designed so these codes can be used according to the national standards and the documentation should show how this is done.

Staffan Romberger
"Staffan_Romberger_NADA%QZCOM.MAILNET"
@MIT-MULTICS.ARPA

Drama Scripts

Can anyone tell me where I can find macros for formatting a play complete with line numbers and different fonts for characters, text and stage directions?

John Kennedy
Mathematical Physics
University College Dublin
Dublin 4, Ireland
JKENNEDY@IRLEARN.Bitnet