

Justification of reviewer's names right ~	II:3 62 II:3 63	T ϵ Xarcane Class T ϵ Xarcane Class	Seating charts	III:1 39	R. Beaman
Letters	II:2 A-32-35	M. Díaz	Spanish	II:2 A-12	M. Díaz
Letterhead	II:2 A-33	M. Díaz	Strings		
Line numbering	III:1 43	T ϵ Xarcane Class	testing for ~ equivalence	II:3 61	L. Price
Lists	II:1 59, 72-72 II:1 98-110 II:2 A-15	A. Keller L. Price M. Díaz	testing for the null ~	II:1 60, 77 II:2 51-51	A. Keller M. Spivak
Margins	II:2 A-19	M. Díaz	Syntax charts	II:3 39-56	M. Plass
Matrices	II:2 A-30	M. Díaz	Table of Contents	II:1 60, 62, 86 II:1 111-118 II:2 A-27-28 II:3 24	A. Keller L. Price M. Díaz B. Beeton
Memos	II:2 A-32-35	M. Díaz	Tables	II:2 A-25-27 III:2 38	M. Díaz Problems column
Miscellaneous			paragraphs in ~		
automatic printing of macro names avoiding "Argument of (control sequence) has an extra }."	II:3 60-61	L. Price	Testing		
conditional evaluation of macros	II:2 50	M. Spivak	integral values	II:1 119-120	P. Milligan, L. Price
input-dependent macro redefinition	II:2 50	M. Spivak	math-style (display, script or scriptscript)	II:2 46	B. McKay
input within \if	II:3 59-60	L. Price	for string equivalence	II:3 61	L. Price
single tokens, identifying	II:2 52	M. Spivak	for the null string	II:1 60, 77 II:2 51-52	A. Keller M. Spivak
Multiplication	II:2 47	B. McKay	Theorems	II:2 A-31-32	M. Díaz
Nonfill			Top, baseline set to ~ of box	II:1 60, 77	A. Keller
macros	II:1 59-60, 74-76 II:2 A-16-18, 36	A. Keller M. Díaz	TUGboat submissions	II:1 53-54 II:3 25	B. Beeton B. Beeton
program (SAIL)	II:1 87-93	L. Price, P. Milligan	Underlining	II:1 59, 73 II:2 A-13	A. Keller M. Díaz
program (Pascal)	II:1 94-97	L. Price, P. Milligan	Uppercase letters		
program errata (SAIL and Pascal)	II:2 43-44		large ~ at beginning of paragraph	II:1 60, 78 II:2 A-16	A. Keller M. Díaz
Notes			Roman numerals	II:1 120-121	P. Milligan, L. Price
output to the writer on a separate file	II:1 60, 76, 85	A. Keller	Verbatim		
printed at end of document	II:2 A-25	M. Díaz	mode	II:1 59-60, 74-76 II:2 A-16-18, 36	A. Keller M. Díaz
Null string, testing for	II:1 60, 77 II:2 51-52	A. Keller M. Spivak	program (SAIL)	II:1 87-93	L. Price, P. Milligan
Numbering, page	II:1 57, 70-71	A. Keller	program (Pascal)	II:1 94-97	L. Price, P. Milligan
line	III:1 43	T ϵ Xarcane Class	Vertical text	II:3 64	T ϵ Xarcane Class
Output routines	II:1 57-58, 60-62, 71, 82-85 II:2 A-18, 40 III:2 38	A. Keller M. Díaz B. Beeton			
Overfining	II:2 A-13	M. Díaz			
Page numbering	II:1 57, 70-71 II:2 A-18, 23	A. Keller M. Díaz			
Paragraphs					
beginning with large capital letters	II:1 60, 78 II:2 A-16	A. Keller M. Díaz			
in tables	III:2 38	Problems column			
indented	II:1 58, 72 II:2 A-13-15	A. Keller M. Díaz			
numbered, see Lists					
Parentheses, assorted sizes	II:2 A-11	M. Díaz			
Pictures, plotting	II:2 48-49	B. McKay			
Point, declaring font families of a particular ~ size	II:1 58-57, 65-66 II:2 A-11	A. Keller M. Díaz			
Proofs	II:2 A-31-32	M. Díaz			
Push-down stacks	III:2 34-36	L. Lamport			
Recursion	II:2 46-48 II:2 53	B. McKay M. Spivak			
References	II:2 A-25	M. Díaz			
Registration marks	III:2 30	B. Beeton			
Roman numerals, uppercase	II:1 120-121	P. Milligan, L. Price			

* * * * *

MULTI-COLUMN OUTPUT FORMAT

Barbara N. Beeton
American Mathematical Society

At the AMS, we are still using the old SAIL version of T ϵ X, which is severely limited in memory capacity. Several of our publications are formatted with very small type in multiple columns; one such publication, the *Combined Membership List* of the Society and two other mathematical organizations, can require over 15,000 6-point characters on a single printed page.

To avoid overloading memory (both memsize and varsize are susceptible), we take advantage of the fact that, to T ϵ X, each column is a "\page". Instead of saving all columns on a page until the final column is complete, each column is shipped out to the .DVI file as soon as it is ready. The several columns which comprise a true page are then "pasted up" by the output driver software, using instructions stored in an "option" file or interactively by responding to a "format spec" request.

There is another advantage to this technique has: Our publication-quality output device, an Alphatype CRS, sets type one baseline at a time, across the full page width for each baseline. Mechanically, a lens (which transmits the type image from a CRT screen to photographic paper) rides along a worm gear for the required distance, then returns. For most applications, type is set in both forward and reverse directions, but in some cases (because of alignment problems), type can be produced in only one direction. If a page contains 4 columns, say, but the baselines from column to column are not evenly aligned (as in the TUG membership list), driving the lens across the full page width could cause the distance traveled to be over 300% greater than necessary, with a corresponding increase in the length of time required to complete a page. Since the Alphatype is a slow machine (wall-clock time can be over 5 minutes for particularly dense pages), the saving is significant.

Initialisation and defaults

The output routine requires two "page counters": `\count0` keeps track of columns (or "`\pages`" in the TeX sense), and `\count9` is used to record the printing page number. Depending on how the output driver keeps track of where it is in a .dvi file, one or the other of these can be used to restart a job in the middle, or to print only selected pages.

Registration marks may be output to delimit the trim area; "T" marks, centered top and bottom, or drawn corners may be chosen. Vertical rules may be drawn between columns. Running heads and folios are accommodated, as are top and bottom matter on the first page; these "full-width" elements are output only on the last segment of a page, while registration marks are generated on each segment, assuring that all true pages have the same number of columns for the sake of the output device driver. Multiple sections, each with its own "first page", may be strung together in the same TeX run.

Page width is calculated dynamically. Column and intercolumn widths are specified (in the input) as an integral number of points. If `\leaders` are to be used, the column and intercolumn widths must be multiples of the leader width, in order to assure correct alignment.

All parameters are initialized; if none are reset, output will be two-column pages of TUGboat dimensions, that is, suitable for printing on 8.5×11 inch paper, with 1-inch side margins and .75-inch margins top and bottom. All running head and folio strings are initially empty, and only those required for a particular job need be reset.

Parameters are of 4 types:

n = integer
 d = dimension (e.g. 12in)
 x = single letter or text string
 c = control sequence

This header file (called `multcol.hdr`) may be used with `AMS-TEX`. If it is, the user must specify

`\useamstex`

and input the header files in the following order:

```
\input 0-0at.mac
\input multcol.hdr
\input 0-0at.fnl
```

Page dimensions (in points) are set as follows:

Page width:

```
\setcolmax{n}  number of columns
\setcolwd{n}   width of one column
\setintercol{n} width of gap between columns
\resetpagewd
```

Page length:

```
\settoplgt{n}  height of first-page header
\setbotlgt{n}  height of first-page footer
\settheadlgt{n} height of running head box
\settfootlgt{n} height of folio box
\setcollgt{n}  height of full-page column
\resetfpagelgt or \resetpagelgt
```

Page and column numbers (initialized to 1) are set by:

```
\setpageno{n}  printing page number
\setspoolno{n} column number for spooler
```

To establish type and placement of trim marks:

```
\settrimtype{x} C, T, or U
\settrimlgt{d}  default = 11in
\settrimwd{d}   default = \pagewd
\setheadmargin{d}
```

Type C gives top and bottom corners at the trim boundaries, T (default) gives "T" marks, and U gives upper corners only. The page contents are centered horizontally within the trim width (the present version does not permit different treatment of left- and right-hand pages, e.g.), and vertically within the trim length unless a different head margin has been specified.

A vertical rule will be drawn between columns if

`\userule{T}`

is specified, or suppressed (default) if F.

To define running heads (all text strings default to null):

```
\setrunners{
  \firstrun x \\  running head on first page:
                  T = yes or F = no (default)
  \rheadfont c \\ font; default is 'current' font
  \outside x \\
  \inside x \\
  \midhead x \\
  \leftmid x \\
  \rightmid x \\
  \runskip d \\
}
```

end of running head items

`\outside` and `\inside` specify the running head segments which appear at the outer margin (left on even pages, right on odd) and toward the spine, respectively. If centered header text is to be the same on left (even) and right (odd) pages, `\midhead` is used; otherwise `\leftmid` and `\rightmid` give the different segments. `\runskip` gives the distance between the baseline of the running head and the top of the page body; default = 10pt. `\setrunners{...}` is cumulative: different portions may be initiated at different times, as convenient; no portion returns to the default value automatically, but must be reset.

A folio, or page footer, may be defined by giving the full description:

```
\setfolio{x}
```

This is not implemented as elaborately as the running head, mainly because folios are not as common as running heads in AMS publications.

Top and bottom matter for the first page are specified by:

```
setfirsthead{x}
setfirstfoot{x}
```

These items are set within `\vboxes` of heights specified by `\settoplgt` and `\setbotlgt`.

The following marks are made available for each completed page:

```
\topterm   \firstmark at top of first column
\lastterm  \botmark from last completed column
```

At the end of a section (bottom of last data column, just before `\eject`) a message or special routine may be inserted:

```
\def \endjobmsg{x}
```

A common use of this feature is a `\send` to establish the starting page number for a subsequent section. In any event, the following message is sent to the terminal (and to the `.err` file) at the end of the job:

```
\send0{data ends on page \curpage, column \xcol}
```

(this requires that file 0 not be `\opened`).

Macro definitions

The following "utility" macros are required:

```
% avoid vertical glue when making up pages:
\def \basezero{\baselineskip Opt\lineskip Opt}
% pseudo-counters:
% structure: \xcount{name}{value}
\def \setxcount#1#2{\setcount#1#2}
\def \xdef#1{\count#1}
\def \advxcount#1{\setcount#1#1}
\def \advcount#1{\xdef#1{\count#1}}
\def \chgxcnt#1#2{\setcount#1#1}
\def \advcount#1#2{\advcount#1#2}
```

% registration marks:

```
% "T" marks centered on top and bottom trim edges
\def \topregister{\vbox to Opt{\vss
  \hbox to \trimwd{\hfil
    \vrule height 24 pt width 0.2pt\hfil}
  \hbox to \trimwd{\hfil
    \vrule height 0.2pt width 0.5in\hfil}}}
\def \tbotregister{\vbox to Opt{
  \hbox to \trimwd{\hfil
    \vrule height 0.2pt width 0.5in\hfil}
  \hbox to \trimwd{\hfil
    \vrule height 24 pt width 0.2pt\hfil}
  \vss}}
```

% corners at limits of trim area

```
\def \ctopregister{\vbox to Opt{
  \hbox to \pagewd{\hss\hbox to \trimwd
    {\vrule depth .5in width 0.2pt
     \vrule depth 0.2pt width .5in
     \hfil
     \vrule depth 0.2pt width .5in
     \vrule depth .5in width 0.2pt}\hss}
  \vss}}
\def \cbotregister{\vbox to Opt{\vss
  \hbox to \pagewd{\hss\hbox to \trimwd
    {\vrule height .5in width 0.2pt
     \vrule height 0.2pt width .5in
     \hfil
     \vrule height 0.2pt width .5in
     \vrule height .5in width 0.2pt}\hss}}}
\def \ctopregister{\vbox to Opt{\vss
  \hbox to \pagewd{\hss\hbox to \trimwd
    {\vrule height .5in width 0.2pt
     \vrule height 0.2pt width .5in
     \hfil
     \vrule height 0.2pt width .5in
     \vrule height .5in width 0.2pt}\hss}}}
\def \cbotregister{\vbox to Opt{\vss
  \hbox to \pagewd{\hss\hbox to \trimwd
    {\vrule height .5in width 0.2pt
     \vrule height 0.2pt width .5in
     \hfil
     \vrule height 0.2pt width .5in
     \vrule height .5in width 0.2pt}\hss}}}
```

AMS-TeX "protects" certain control sequences, e.g. `\page` as `\page1`, and disables the "basic" sequence. The following permits an orderly transition to the AMS-TeX conventions:

```
\def \isamstex{B}
\def \useamstex{\gdef\isamstex{A}
\gdef\normaloutput{/outa}}
```

Initialization comprises a large number of control sequence pairs, of the following structure:

```
\def \colmax{2}
\def \setcolmax #1{\gdef\colmax{#1}}
```

The following conform to this structure, with defaults as shown:

<code>\setcolmax</code>	<code>\colmax</code>	2
<code>\setcolwd</code>	<code>\xcolwd</code>	225
<code>\setintercol</code>	<code>\intercol</code>	18
<code>\settoplgt</code>	<code>\xtoplgt</code>	0
<code>\setbotlgt</code>	<code>\xbotlgt</code>	0
<code>\setrheadlgt</code>	<code>\xrheadlgt</code>	24
<code>\settfootlgt</code>	<code>\xrfootlgt</code>	<code>\xrheadlgt</code>
<code>\setcollgt</code>	<code>\xcollgt</code>	648
<code>\settrimlgt</code>	<code>\xtrimlgt</code>	11in
<code>\settrimwd</code>	<code>\xtrimwd</code>	<code>\pagewd</code>
<code>\userule</code>	<code>\xrrule</code>	F
<code>\setfolio</code>	<code>\folio</code>	0

Some of the initialization macros are more elaborate:

```

\def \headmarginw{F}
\def \headmarginlgt{}
\def \setheadmargin #1{\gdef\headmarginw{T}
\gdef\headmarginlgt{#1}}

\def \topregister{\ttopregister}
\def \botregister{\tbotregister}
\def \settrimtype #1{
\if T#1{\gdef\topregister{\ttopregister}
\gdef\botregister{\tbotregister}}
\else{\if C#1{\gdef\topregister{\ctopregister}
\gdef\botregister{\cbotregister}}
\else{\if U#1{\gdef\topregister{\ctopregister}
\gdef\botregister{}}
\else{\send0{invalid trim type; T marks will be used}}}}}}

\def \firstrunner{F}
\def \firstfolio{F}
\def \rhfont{}
\def \outrunner{}
\def \inrunner{}
\def \leftmidrunner{}
\def \rightmidrunner{}
\def \runskiplgt{10pt}
\def \setrunners #1{
\def\firststrum#1\\\{\gdef\firstrunner{#1}}
\def\rheadfont#1\\\{\gdef\rhfont{#1}}
\def\outside#1\\\{\gdef\outrunner{#1}}
\def\inside#1\\\{\gdef\inrunner{#1}}
\def\midhead#1\\\{\gdef\leftmidrunner{#1}
\gdef\rightmidrunner{#1}}
\def\leftmid#1\\\{\gdef\leftmidrunner{#1}}
\def\rightmid#1\\\{\gdef\rightmidrunner{#1}}
\def\runskip#1\\\{\gdef\runskiplgt{#1}}
#1}

\def \firsthead{}
\def \setfirsthead #1{\gdef\firsthead{
\vbox to \xtoplgt pt{
\if \xcol\colmax{#1}
\else{}}}}

\def \firstfoot{}
\def \setfirstfoot #1{\gdef\firstfoot{
\vbox to \xbotlgt pt{
\if \xcol\colmax{#1}
\else{}}}}

\def \setcurpage{\ifpos9{\xdef\curpage{\count9}}
\else{\setcount7 -\count9
\xdef\curpage{-\count7}}}
adjust for roman numerals

\def \pageno{1}
\def \resetcurpage{\setcount9\pageno \setcurpage}
\def \setpageno #1{\gdef\pageno{#1}\resetcurpage}
\resetcurpage

\setcount0 1
\def \setspoolno #1{\setcount0 #1 }

\topbaseline Opt
align tops of multiple columns rather than baselines
to accommodate type of different sizes

```

Page dimensions are calculated using counter arithmetic:

```

\def\resetpagelgt{
  \setcount2 \xcolgt
  \advcount2 by \xrheadlgt
  \advcount2 by 2
  \advcount2 by \xrfootlgt
  \xdef \pagelgt{\count2 pt}
  \xdef \rheadlgt{\xrheadlgt pt}
  \xdef \rfootlgt{\xrfootlgt pt}}

\def \resetfpagelgt{\resetpagelgt
  \xdef \toplgt{\xtoplgt pt}
  \xdef \botlgt{\xbotlgt pt}
  \setcount1 \xcollgt
  \advcount1 by -\xtoplgt
  \advcount1 by -\xbotlgt
  \vsize \count1 pt
  \gdef \fpage{T}}
\resetfpagelgt

\def \howwide{\setcount8\xcol \setcount3 0 \sowide}
\def \sowide{\advcount8 by -1
  \advcount3 by \xcolwd
  \ifpos8{\advcount3 by \intercol
    \sowide}
  \else{\xdef \thiswide{\count3 pt}}}}

\def \resetpagewd{
  \xdef \colwd{\xcolwd pt}
  \hsize \colwd
  \xdef \xcol{\colmax}
  \howwide
  \xdef \pagewd{\thiswide}
  \xdef \xcol{1}}
\resetpagewd

```

ordinary page
length of full-page column
add length of running head
include \maxdepth
add length of folio
full-page length

first page of a section
length of first page top matter
length of first page bottom matter

length of column on first page

for each column, add column width
add intercol for all but last column
keep going, up to number of columns
in current "page"

column measure

Running heads and folios are pieced together from input segments for use in the output routine:

```

\def \runner{\hbox to \pagewd{\rhfont
  \spose{\hbox to \pagewd
    {\ifeven9{\hfil\leftmidrunner\hfil}
      \else{\hfil\rightmidrunner\hfil}}}\!
  \ifeven9{\unskip\outrunner\hfill\null\inrunner\unskip}
  \else{\unskip\inrunner\hfill\null\outrunner\unskip}}}

\def \runhead{\vbox to \rheadlgt{\vss
  \if \xcol\colmax{
    \if T\fpage{\if F\firstrunner{}
      \else{\runner}}
    \else{\runner}}
  \else{}
  \vskip\runskiplgt}}

\def \runfoot{\vbox to \rfootlgt{\vss
  \if \xcol\colmax{\folio}
  \else{}}}}

```

A couple more utility definitions for special circumstances:

```

\def \ruler{\if T\xrule{\hbox to \intercol pt{\hfil\vrule\null
  \vbox to size{\hbox to Opt{} \vfil}\hfil}}
  \else{}}

\def \markit{}

```

to permit tricky code of the sort used to
insert continuation entries at top of next column

At last! The actual output routine!

```

\def \midpage{\hbox to \trimwd{\hfil
  \vbox to \trimlgt{\basezero
    \topregister
    \if T\headmarginw{\vskip\headmarginlgt}
    \else{\vfill}
    \markit\par
    \vbox to \pagelgt{\basezero
      \runhead
      \if T\page{\firsthead}
      \else{}
      \gdef\lastterm{\botmark}
      \howwide          calculate width of current "page"
      \vbox to size{\hbox to \thiswide{\hfil
        \if 1\col{}
        \else{\ruler}!!
        \if A\isamster{\page!}
        \else{\page}}
      \if T\page{\firstfoot}
      \else{}
      \runfoot }
    \vfill
    \botregister}\hfil}
\advcount0 }          number each output segment uniquely

\def \xcolstart{\if \xcol\colmax{\output{\outa}}
  \else{\output{\outb}} }

\def \xcolend{\if \xcol\colmax{\advcount9 \xdef\curpage{\count9}
  \vsize\collgt pt
  \gdef\xcol{1}
  \gdef\page{F}}
  \else{\advxcount\xcol} }

\output{\outa}
\def \outa{\xcolstart
  \xdef\topterm{\firstmark}
  \midpage
  \xcolend}
\def \outb{\xcolstart
  \midpage
  \xcolend}

And finally, various macros to fill out incomplete columns, terminate sections neatly, and finish up a job,
reporting to the user where it ended. In \newcol, intended to permit manual balancing of the last page of a
section, the penalty is necessary to overcome possible large negative penalties at other points in the column,
especially when the column is broken very close to maximum length.
\def \newcol{\par\penalty -900\vfil\ejct}

\def \nullcol{\hbox to \colwd{\null}\ejct}          empty column to fill page

\def \blankit{
  \if T\pass{\xdef\pass{F}\vfill\reportlastcol\ejct\blankit}
  \else{\if 1\xcol{}
    \else{\nullcol\blankit}}}

\def \endsection{\gdef\pass{T}\blankit}

\def \endjobmsg{}          allow special messages, sending starting page for
                           next section, etc.

\def \reportlastcol{\send0{data ends on page \curpage, column \xcol}\endjobmsg}

```