

Anatomy of a T_EX Macro Package

Arthur M. Keller

Stanford University

Abstract. A T_EX macro package is described that produces various formats of output. Parts of the macro package are based on the basic and book formats described in the T_EX manual as well as the format for the T_EX manual itself.

1 Introduction

This macro package was developed over approximately a year and a half. It started when I first began to use T_EX, and proceeded as I began to write macros that others found useful. Much of the development has been for a forthcoming book to be published by McGraw-Hill teaching PASCAL to students without a knowledge of computers.* Further, the work of others has been borrowed and adapted in preparing these macros.

It is assumed that the reader is familiar with T_EX. This documentation is intended to be read while perusing the code itself.

The macro package consists of a file called ARKTEX.TEX which should reside in the T_EX system files area. This file refers to other files which are loaded if needed. This package was designed on the SU-AI system and uses the SAIL character set as described in the T_EX manual. Others not using the SAIL character set may find it useful to change the \chcode's at the start of the file as well as some of the one character macros.

2 ARKTEX.TEX

The file ARKTEX.TEX is divided up into about twenty sections separated by horizontal rules. In the file, these are usually on separate pages (i.e., separated by control-L's).

2.1 Standard Basic Stuff

This section consists primarily of text that appears in BASIC.TEX. The \chcode's should be changed as necessary for your system.

2.2 Font Definitions and Related Macros

This section consists of macros for fonts of various fonts and sizes. The \chcode on the first line is to all © to be parsed correctly on this page because elsewhere © has \chcode of 13. The fonts on this page are primarily for eight, nine, and ten point typesetting. Other random fonts also exist. The \: macro has been redefined to save the font letter in \fontcode. This is used by the macros in "Definitions of Odd Characters" that produces output of different characters depending on which font is currently in use. Do not use the \usefont macro, but the \curfont macro may be used as an alternate to the \: macro.

The \loadfont macro is used to allow documents to use fonts that exist only on some systems to refer to these fonts symbolically. Ordinarily these fonts are not preloaded. However, a document attempting to use such a font for a particular output device for which the font exists may do so. For example, at Stanford, some of the fonts loaded by \loadfont exist only on the XGP or on the Dover but not on the Alphatype.

This work was supported in part by the T_EX project under Prof. Donald E. Knuth and Dr. Luis Trabb-Pardo, and by a National Science Foundation Graduate Fellowship.

Author's address: Computer Science Dept, Stanford University, Stanford, CA 94305. ARPANET address: ARK at SU-AI.

*The book will probably be titled *A First Course in Computer Programming Using PASCAL* and it will be available about January 1982.

The various sizes of type are referred to by the macros `\tenpoint`, `\ninepoint`, and `\eightpoint` to obtain ten, nine, and eight point type faces respectively. For each size there are the following fonts provided: roman (`\rm`), slanted roman (`\sl`), boldface roman (`\bf`), italic (`\it`), math italic (`\mi`), teletype (`\tt`), and symbol (`\sy`). In addition, small caps (`\sc`) is provided for ten point. Each size of type also includes definitions for the width of a digit (`\9`) as well as complete math mode information.

To start text in one of these sizes specify `\tenpoint`, `\ninepoint`, or `\eightpoint`. Customized macros for various point sizes may be constructed by redefining the macros `\usertenpoint`, `\userninepoint`, and `\usereightpoint`. However, if the most of the document is to be in that type size, say `\usertenpoint`, etc., instead at the start of the document. If you use `\startcode` and `\endcode`, `\fontsize` will be used to determine what font size to return to. This macro may be defined at any time prior to such usage, but it can be done easily by saying `\usertenpoint` if desired.

2.3 Definitions of Odd Characters

This section includes definitions that will allow characters such as `@` and `#` to be parsed correctly in any mode. Characters that have no other usage, such as `@` may appear without a preceding `\`, as they have a `\hcode` of 13. Characters that have other usages, such as `#`, must be preceded by a `\` if they are to appear as the character instead of being used for their standard purpose. However, because they are control sequences without an argument, you must put `\l` following them to avoid the space after them getting ignored.

Characters in teletype mode are fixed width characters. Therefore, the `\ttchar` macro takes a specified character and puts it in the desired size box so that the remainder of the line will line up.

The `\fontclassify` macro determines which mode or typestyle is being used and chooses the correct argument to emit. This allows `\#` to produce a `#` in the right font.

2.4 Redefinitions of One Character Macros

This section consists of redefinitions of one character macros so that they work in any mode. Users who redefine macros such as `\le` to print `≤` will also find the redefinitions useful. Macros starting with `\M` are defined to save the original definitions. Macros without an `M` are defined to work in or out of math mode. These original macros are redefined to match the new set. The exception is `\!` as it means different things in math mode than in non-math mode.

2.5 Make Some Math Things Work Anywhere

This group of macros works just like those in the previous section.

2.6 Page Numbering

The section on page numbering is rather complicated by the existence of macros to defer text. The pertinent macro for users is `\setpagecount` which sets the page number on the following page to the specified number. The page number on the current page is set to one less than that number. This mess is done because the author likes to put out an extra page describing what is going on whenever changing page numbers. In particular, output devices that do not put out header pages permit several users output to appear without intervening pages. Putting out your own separator page reduces the chance that your output will be misfiled. Lastly, the `\chapterbegin` macro in book format ejects the page first, so the author usually puts fixed garbage on the previous page. It's also a good place to put copyright notice if the file is going to be copyrighted. Most importantly, using `\settitle` of book format on the ejected page guarantees that the next page will have the correct headings.

2.7 `\output`, Style, and Format Routines

These macros are the heart of the claim of providing many formats of output. They fall into four

categories: overhead, output, style, and format routines. Overhead routines are used in many places and are obvious. These are `\normal`, `\resetsize`, and `\everyoutput`.

The output routines set the `\output` macro. Also, they should also set the macros `\normalhsize` and `\normalvsize`. See `OPLAIN.TEX` for the minimum required in an output routine.

Style routines set such things as paragraph spacing. See `OBLOCK.TEX` for the minimum required in an style routine.

Format routines are simply style and output routines in the same file or macro.

See the descriptions of the individual files for more information.

2.8 Footnotes

The footnote macro has gone through several generations. The latest one uses the `\botsep` to insert the horizontal bar. If your system does not yet support `\botsep`, you may have to be more clever about when to put in the bar and when to delete it. The author's previous method was to insert a bar if this was the first footnote on a page. The output routine would reset the first-footnote-on-the-page switch. However, this would occasionally fail in that the first footnote on the page would actually be generated before the output routine was called (e.g., if the paragraph is split on two pages). Then the `\firstfootnote` macro would be used which would hack the switches appropriately so that the next footnote would not get a bar. This involved setting a switch that the output routine cleared so that the output routine would not clear a second switch that indicated that a bar had already been output for that page. What a mess! Anyway, get a new version of TeX.

The macro `\footnote` provides automatically numbered footnotes. The numbers are started at 1—the macro pre-increments it.

There are three macros of characters for using for footnotes. These are `\upstar`, `\dagger`, and `\ddagger`.

2.9 Paragraphs

This section consists of macros for various hanging paragraphs. The `\hangbox` macro creates a box of width based on argument 1 containing argument 2. The remaining lines of the paragraph will be indented the same width. For example, `\hangbox to 30pt {foo}bar etc.`, will produce a paragraph containing bar etc., indented to 30 points with the first 30 points of the first line containing foo. The macros `\levelone`, `\leveltwo`, and `\levelthree` generate such hanging boxed paragraphs to 20 points, 40 points, and 60 points, respectively. However, the contents of the boxes are left justified in a twenty-point box that is right justified in the 40- or 60-point box. The following are uses of `\levelone`, `\leveltwo`, and `\levelthree`:

1. This is a short box followed by a long paragraph. Isn't it amazing to see what drivel can be published in the guise of an example. Put your ad here; to find out whether you can call 936-1212.
2. This is a medium box followed by a long paragraph. Isn't it amazing to see what drivel can be published in the guise of an example. For a good time call 767-8989.
3. This is a long box followed by a long paragraph. Isn't it amazing to see what drivel can be published in the guise of an example. For example, did you know that when the author finishes his Ph.D., he'll be looking for a teaching job? A reference to this article will pad his C.V.

The macros `\number` and `\nnumber` create indented paragraphs with boxes of 20 and 50 points respectively. That's right, `\number` is just like `\levelone`.

The `\indpar` macro takes the argument and creates an paragraph indented on both sides to 40 points. Normal paragraph indentation or paragraph separation must be done by you. A `\strut` has been inserted to get the correct line spacing between the paragraph and preceding and following text to handle risers and descenders properly. However, no `\parskip` glue is inserted and 1 point is inserted for the assumed `\lineskip`.

The `\hdr` macro creates a centered boldface heading consisting of its argument that makes a good section heading if you are not using book format.

2.10 List Definitions

Now that you know all about the paragraph macros, you might expect macros for doing numbered lists automatically. There are three levels of numbering. The first level uses `\list` followed by an argument which is the initial number for counting. Then `\item` is used to precede each item. You may use `\itemindent` to indent the same amount as `\item` for continuing the following paragraph, for example. The macro `\bitem` gives a centered bullet in a 20 point box starting the hanging paragraph.

The second level of counting is in roman numerals. Put the number you want to start counting from after the `\sublist`. Note that this number should be positive, so `-3` gives "iii." As you might expect, there are `\subitem` and `\subitemindent`.

The third level of counting is letters. Put the letter you want to start counting from after the `\subsublist`. And there are `\subsubitem` and `\subsubitemindent`.

2.11 Underlining and Boxes

This section consists of macros for doing various kinds of under- and overlining as well as lined boxes. The `\undertext` and `\overttext` macros underline and overline in horizontal mode. And `\leaderline` gives a leader of a rule.

The `\boxit` macro is from exercise 21.3 of the T_EX manual. However, `\sizeboxit` makes the box a specific size. Also, `\boxitnoglue` boxes the box without 3 points of space on all sides. If you want to put `\boxit`'s in a `\valign` or put straight text inside, use `\Boxit` or `\Boxitnoglue`, which reverse horizontal and vertical mode. To put corner L's around a box, use `\Lboxit`.

To demonstrate interactive system output, it is useful to display the user entered data underlined. To underline the second half of a line, say `\type prompt>underlined text`. Use `\ttype` the same way for indented dialogue.

2.12 Penalties

Aren't these obvious. They do save space in macros over their expansions.

2.13 `\nofill \endnofill`

This is the first of the verbatim mode set. To use it, say `\nofill` followed by the text, followed by `\endnofill`. Line breaks appear exactly where they do in the input text. Exactly as many spaces appear in the output as in the input. The code is listed verbatim without page breaks. To allow page breaks, say `\allowbreak`. A blank line is generated if there is no page break. To have no space generated if there is no page break, use `\allowbreaknoglue`.

Tabs are not allowed in verbatim mode. This is because it is not clear how many spaces to generate for a tab. If you think you know better, say `\def\tab{definition}`.

2.14 `\startcode \endcode`

The `\startcode` `\endcode` sequence produces verbatim code in `\displayfont`. See the previous section and the code for more details.

In `\startcode` mode, `\le` and `\ge` produce \leq and \geq , respectively. These revert back to their former meanings at the end.

2.15 Verbatim Mode Using `$$\halign$$`

Verbatim mode is just like `\startcode` mode except that the calling conventions are different and it may appear in an `\halign`. To use it precede the code with `\halign{` and follow the code with a right brace on its own line. To allow a page break, code `\breakhere%`. Note that the `%` is required.

The `\threecol` macro generates `\verbatim` mode except with three columns instead of one. To put a box around verbatim code, say `\Boxit{\verbatim{code}}`.

2.16 Notes

Notes are useful to provide descriptions of things that you want to fix. The description of the file `MNOTES.TEX` appears later.

2.17 Index Macros

An index package is described in Vol. 1, No. 1 of TUGboat.

2.18 Defer Mode

Defer mode is useful for specifying an entire page that is to appear as soon as possible. The description of the file `DEFER.TEX` appears later.

2.19 Table of Contents

This set of macros generates a table of contents compatible with book format. The description of the file `MTOFC.TEX` appears later.

2.20 Interesting Hacks

This section consists of interesting macro hacks that are useful for one and all.

The `\ifnull` macro determines if argument 1 is null. If so it expands argument 2; otherwise, argument 3 is expanded. To call say: `\ifnulltext\then{code}\else{code}`.

The `\bracex`, `\dnbrace`, and `\upbrace` macros are from page 103 of the TeX manual. On the other hand, `\blackslug` is from page 167.

The `\boxtop` macro sets the baseline at the top of the box. This is useful for lining up variable sized boxes at the top. For example, to line up `\hbox` par's in a `\halign`, use `\boxtop` around the `\hbox` par's.

The `\topspace` and `\magnify` macros is from the new version of `BASIC.TEX`.

There are two macros for playing with counters. Use `\setq` to set the control sequence which is the first argument to the counter number in the second argument. Use `\advcountq` to increment the counter in the control sequence. With these macros, you can save counters for what they are really needed for: setting up the correct numbers for output routines.

The `\done` macro goes at the end of the document.

The `\capitalpar` macro creates a paragraph like the "Gentle reader" at the start of the TeX manual.

2.21 Default Options

See `\startcode` and `\endcode` for a description of `\displayfont`. See section 1.6 for a description of page numbering.

2.22 Documentation

Every macro package should have some.

3 MBOOK.TEX

The book format macro package is the most developed of the macro formats. The `\bookoutput` routine handles `\titlepage` (it sets `\tpage` to T), as well as proper placement of the page numbers. The page

numbers will appear on the top of the page if `\pagenumberarea` is T, on the bottom if B; otherwise, no page numbers appear. It defaults to T.

The `\pagenumberregion` macro defines the format of the page heading (or footing). It defaults to the macro `\boxpagenumberregion`, which produces the format in the TeX manual and this document. The right and left headings are specified by `\titlemark{right}{left}`.

To get a page without a page heading, say `\titlepage`.

Chapters, sections, subsections, and diminished sections are all numbered automatically. To use, say `\chapterbegin`, `\sectionbegin`, `\subsectionbegin`, or `\dimsectionbegin`, all followed by the chapter or section name in braces. For an unnumbered chapter, such as an appendix or a table of contents, use `\specialbegin` followed by the name in braces. Use `\settitle` to set the left and right headings if you aren't using the other macros in this paragraph.

To get data for a table of contents, use `\inittofc{filename}`.

4 MACACM.TEX

This produces 25% oversized output for the ACM camera-ready copy specifications. At the start of the paper, code `\useacmformat`. Then define `\title` and `\authors`. Then define the title portion of the paper. Next, say `\endoftitle`. When done with the paper, say `\endofpaper\end`.

Note that `\defer` does not work with this format.

5 OPLAIN.TEX

This produces output unadorned with page numbers or anything.

6 OBASIC.TEX

This produces output with page numbers as in BASIC.TEX.

7 OWOODS.TEX

This produces output with page numbers on the bottom of the page with hyphens around the numbers as popularised by Don Woods at Stanford.

8 SBLOCK.TEX

This produces block style paragraphs with about 6 points of space between paragraphs.

9 SBASIC.TEX

This produces indented paragraphs as in BASIC.TEX.

10 MNOTES.TEX

This file contains macros to generate notes to the writer. Say `\initnotes` to create the file. Say `\sendnotes{text}` to output text. The `\putnotes` macro takes the notes and outputs them in the listing.

11 DEFER.TEX

Defer mode is used to produce a floating figure that takes one or more whole pages. Like `\topinsert` for floating figures, it is used in vertical mode. However, defer mode handles multiple page figures and will keep several figures in the order specified.

To use, say `\defer` followed by the figure followed by `\enddefer`.

Defer mode does not work with multicolumn formats.

12 MTOFC.TEX

The table-of-contents package produces a table of contents based on the data files produced if `\inittofc` is used in book format. To use, say `\begintofc` followed by `\chaptertofc`, `\sectiontofc`, etc., macros each followed by a title in braces and a page number terminated by a period. Use `\endtofc` at the end of the table of contents.

13 Acknowledgments

This paper could not have been written and published without TeX designed by Donald E. Knuth. He also provided some advice and encouragement, as did Luis Trabb-Pardo. Many of the macros were written with the help of Jim Boyce. Brent Hailpern designed the original defer mode. Max Diaz provided additional suggestions and several macros as well. Denny Brown suggested some of the macros and the idea of supporting many formats. My advisor, Gio Wiederhold, gave encouragement and enlightened criticism through his attempts to use these macros. The staff of the American Mathematical Society were very helpful and interested in getting this manuscript written and published.

14 The Macros

The source for the macro package follows.

Standard BASIC Stuff

```

\chcode '45+5      % % **** N.B. this must be first
\chcode '173+1     % {
\chcode '176+2     % }
\chcode '44+3      % $
\chcode '26+4      % @
\chcode '43+6      % #
\chcode '136+7     % †
\chcode '1+8       % Δ

```

```

% Shorthands for certain definitions
\def \trace{\chpar0+} \trace'1400345
\def \jpar{\chpar1+}
\def \hpen{\chpar2+}
\def \ragged{\chpar8+}

```

```

% centerings
\def \lft#1{\hfill #1\hfill }
\def \ctr#1{\hfill #1\hfill }
\def \rt#1{\hfill #1}
\def \top#1{\vfill #1\vfill }
\def \mid#1{\vfill #1\vfill }
\def \bot#1{\vfill #1}

```

```

\def \ljustline#1{\hbox to size{#1\hss}}
\def \ctrline#1{\hbox to size{\hss #1\hss}}
\def \rjustline#1{\hbox to size{\hss #1}}

```

```

\def \ldots{(. \condthinspace. \condthinspace.)}
\def \cdots{(\char '401\condthinspace\char '401\condthinspace\char '401)}
\def \ldotss{(. \condthinspace. \condthinspace. \condthinspace)}
\def \cdotss{\cdots \condthinspace}
\def \ldotssm{(\condthinspace. \condthinspace. \condthinspace. \condthinspace)}
\def \vdots{\vbox{\baselineskip 4pt \vskip 6pt \hbox{.}\hbox{.}\hbox{.}}}

```

```

\def \cpile #1{\vcenter {\halign {\hfill ## $\hfill \cr #1}}}
\def \lpile #1{\vcenter {\halign {$## $\hfill \cr #1}}}
\def \rpile #1{\vcenter {\halign {\hfill ## $\cr #1}}}

```

```

\def \null{\hbox {}}.

```

```

\def \spose #1{\hbox to 0pt{#1\hss}}

```

```

\def \log{\mathop{\char l\char o\char g}\limitswitch}
\def \lg{\mathop{\char l\char g}\limitswitch}
\def \ln{\mathop{\char l\char n}\limitswitch}
\def \lim{\mathop{\char l\char i\char m}}
\def \limsup{\mathop{\char l\char i\char m\char s\char u\char p}}
\def \liminf{\mathop{\char l\char i\char m\char i\char n\char f}}
\def \sin{\mathop{\char s\char i\char n}\limitswitch}
\def \cos{\mathop{\char c\char o\char s}\limitswitch}
\def \tan{\mathop{\char t\char a\char n}\limitswitch}
\def \cot{\mathop{\char c\char o\char t}\limitswitch}
\def \sec{\mathop{\char s\char e\char c}\limitswitch}
\def \csc{\mathop{\char c\char s\char c}\limitswitch}
\def \max{\mathop{\char m\char a\char x}}
\def \min{\mathop{\char m\char i\char n}}
\def \sup{\mathop{\char s\char u\char p}}
\def \inf{\mathop{\char i\char n\char f}}
\def \det{\mathop{\char d\char e\char t}}
\def \exp{\mathop{\char e\char x\char p}\limitswitch}
\def \Pr{\mathop{\char P\char r}}
\def \gcd{\mathop{\char g\char c\char d}}
\def \lcm{\mathop{\char l\char c\char m}}
\def \choose{\comb()}
\def \leftset{\mathopen{\{}}
\def \rightset{\mathclose{\}}}
\def \modop{<\, \mathbin{\char m\char o\char d}\penalty 900<\,}
\def \mod#1{\penalty 0; (\char m\char o\char d, \, #1)}
\def \eqv{\mathrel{\char'421 }}
\def \neqv{\mathrel{\not\eqv}}

```



```

\def\eqalign#1{\baselineskip15pt\lineskip3pt
  \vcenter{\halign{\hfill$\dispstyle{##}$@$\dispstyle{\null##}$\hfill
    \cr#1}}}
\def\eqalignno#1{\baselineskip15pt\lineskip3pt
  \vbox{\tabskip 0pt plus 1000pt minus 1000pt
    \halign to size{\hfill$\dispstyle{##}$\tabskip 0pt
      @$\dispstyle{\null##}$\hfill\tabskip 0pt plus 1000pt minus 1000pt
      @\hfill$ ##$\tabskip 0pt\cr#1}}}
\def\twoline#1#2#3{\vbox{\hbox to size{$\quad\dispstyle{#1}$\hfill}
  \vskip#2\hbox to size{\hfill$\dispstyle{#3}\quad$}}}
\def\chop to#1pt#2{\hbox{\lower#1pt\hbox{\lower100pt\hbox{\raise100pt
  \hbox{$\dispstyle{#2}$}}}\vskip-100pt}} % pretends that #2 is #1pt deep

```

• Font Definitions and Related Macros

```

\chcode'100+12 % allow @ on this page to be parsed correctly

% font definitions for 8, 9, and 10 point fonts and friends
\font @+cmathx
\font a+cmr10 \font b+cmr9 \font c+cmr8 \font d+cmr7 \font e+cmr6 \font f+cmr5
\font g+cmi10 \font h+cmi9 \font i+cmi8 \font j+cmi7 \font k+cmi6 \font l+cmi5
\font G+cmti10 \font H+cmti9 \font I+cmti8
\font m+cmsc10
\font n+cms10 \font o+cms9 \font p+cms8
\font q+cmb10 \font r+cmb9 \font s+cmb8
\font t+cmtt \font T+cmtt9 \font U+cmtt8
\font u+cmsy10 \font v+cmsy9 \font w+cmsy8 \font x+cmsy7 \font y+cmsy6 \font z+cmsy5

% font definitions for random desired fonts
\font ;+cmtitl
\font <+cmssb \font =+cmss12 \font >+cmss8 \font ?+cmsss8

% font request macros
\let \usefont=:
\def \curfont #1{\usefont #1\def\fontcode{#1}}
\let \: =\curfont

```

```

% font name macros
\def \loadfont#1#2#3{\font #1+#2 \gdef #3{\:#1}#3}
\def \big{\loadfont D{cmr12}{\big}}
\def \ms25{\loadfont A{ms25}{\ms25}}
\def \nons{\loadfont B{nons}{\nons}}
\def \peni11{\loadfont P{peni11}{\peni11}}
\def \stan70{\loadfont S{stan70}{\stan70}}
\def \biggfont{\loadfont C{cmr10 at 20pt}{\biggfont}}
\def \bigggfont{\loadfont E{cmr10 at 30pt}{\bigggfont}}
\def \cmrten{\:a}
\def \cmrnine{\:b}
\def \cmreight{\:c}
\def \cmrseven{\:d}
\def \cmrsix{\:e}
\def \cmrfive{\:f}
\def \cmiseven{\:j}
\def \cmisix{\:k}
\def \cmifive{\:l}
\def \cmscten{\:m}
\def \cmtitl{\:;}
\def \cmssb{\:<}
\def \cmss12{\:=}
\def \cmsseight{\:>}
\def \cmsseight{\:?}

% font family definitions
\def \tenpoint{\baselineskip 12pt
  \dispskip 12pt plus 3pt minus 9pt
  \dispaskip 0pt plus 3pt
  \dispbskip 7pt plus 3pt minus 4pt
  \def \strut{\lower 3.5pt
    \vbox to 12pt{}}% i.e., \lower 1pt+.25em\vbox to 2pt+1em{}
  \def \rm{\:a}
  \def \sl{\:n}
  \def \bf{\:q}
  \def \it{\:G}
  \def \mi{\:g}
  \def \tt{\:t}
  \def \sy{\:u}
  \def \sc{\:m}
  \def \biglp{\mathopen {\vcenter {\hbox {\:0\char '}}}}
  \def \bigrp{\mathclose{\vcenter {\hbox {\:0\char '1}}}}
  \def \9{\hskip 5pt}
  \mathrm adf
  \mathit gjl
  \mathsy uxz
  \rm \usertenpoint}
\def \usertenpoint{}
\def \usetenpoint{\gdef\fontsize{\tenpoint}\tenpoint}

```

```

\def \ninepoint{\baselineskip 11pt
  \dispskip 11pt plus 3pt minus 8pt
  \dispaskip 0pt plus 3pt
  \dispbskip 6pt plus 3pt minus 3pt
  \def \strut{\lower 3.25pt\vbox to 11pt{}}% see tenpoint for explanation
  \def \rm{\:b}
  \def \sl{\:o}
  \def \bf{\:r}
  \def \it{\:H}
  \def \mi{\:h}
  \def \tt{\:T}
  \def \sy{\:v}
  \def \biglp{\mathopen {\hbox{\:a{}}}}
  \def \bigrp{\mathclose{\hbox{\:a{}}}}
  \def \9{\hskip 4.625pt}
  \mathrm bef
  \mathit hkl
  \mathsy vyz
  \rm \userninepoint}
\def \userninepoint{}
\def \useninepoint{\gdef\fontsize{\ninepoint}\ninepoint}

\def \eightpoint{\baselineskip 9.5pt
  \dispskip 5pt plus 3pt minus 2pt
  \dispaskip 0pt plus 3pt
  \dispbskip 5pt plus 3pt minus 2pt
  \def \strut{\lower 2.75pt\vbox to 9.5pt{}}% see tenpoint for explanation
  \def \rm{\:c}
  \def \sl{\:p}
  \def \bf{\:s}
  \def \it{\:I}
  \def \mi{\:i}
  \def \tt{\:U}
  \def \sy{\:w}
  \def \biglp{\mathopen {\hbox {\:a{}}}}
  \def \bigrp{\mathclose{\hbox {\:a{}}}}
  \def \9{\hskip 4.25pt}
  \mathrm cef
  \mathit ikl
  \mathsy wyz
  \rm \usereightpoint}
\def \usereightpoint{}
\def \useeightpoint{\gdef\fontsize{\eightpoint}\eightpoint}

\mathex 0

% definitions of large parentheses
\def \biggpl{\mathopen{\vcenter{\hbox{\:0\char'22}}}}
\def \biggrp{\mathclose{\vcenter{\hbox{\:0\char'23}}}}
\def \bigggpl{\mathopen{\vcenter{\hbox{\:0\char'40}}}}
\def \bigggpr{\mathclose{\vcenter{\hbox{\:0\char'41}}}}

```

```
% definitions of glue
\def \qqquad{\quad\quad}
\def \xskip{\hskip 7pt plus 3pt minus 4pt}
\def \yskip{\penalty-50\vskip 3pt plus 3pt minus 2pt}
\def \yyskip{\goodbreak\vskip 6pt plus 6pt minus 4pt}
```

• Definitions of Odd Characters

```
\chcode'272+'3072 % this makes formulas like "$x:=x+1$" and "$f\?:X\to Y$" work
```

```
% ttchar puts the char into a \tt fixed width box
```

```
\def\ttchar#1{\save1\hbox{\ } \hbox to 1wd1{\hskip0pt plus1000pt minus1000pt
#1\hskip0pt plus1000pt minus1000pt}}
```

```
% fontclassify selects the right char based on what the current font is
```

```
\def\fontclassify#1#2#3{\ifmode{#1}
  \else{\if t\fontcode{#2}
  \else{\if T\fontcode{#2}
  \else{\if U\fontcode{#2}
  \else{(#3)}}}}}
```

```
%#1 is math, #2 is tt, #3 is others
```

```
% char macro definitions
```

```
\def\down{\fontclassify{\mathrel{\char'443}}{\ttchar{\sy\char'43}}{\sy\char'43}}
```

```

\def\alpha{\fontclassify{\char'213}{\ttchar{\mi\char'13}}{\mi\char'13}}
\chcode'2+13
\def\beta{\fontclassify{\char'214}{\ttchar{\mi\char'14}}{\mi\char'14}}
\chcode'3+13
\def\^{\fontclassify{\mathbin{\char'536}}{\ttchar{\if t\fontcode{\:z\char'136}
\else{\if T\fontcode{\:z\char'136}\else{\:z\char'136}}}{\sy\char'136}}
\chcode'4+13
\def\~{\fontclassify{\char'472}{\ttchar{\sy\char'72}}{\sy\char'72}}
\chcode'5+13
\def\epsilon{\fontclassify{\char'217}{\ttchar{\mi\char'17}}{\mi\char'17}}
\chcode'6+13
\def\pi{\fontclassify{\char'231}{\ttchar{\mi\char'31}}{\mi\char'31}}
\chcode'7+13
\def\lambda{\fontclassify{\char'225}{\ttchar{\mi\char'25}}{\mi\char'25}}
\chcode'10+13
\def\omega{\fontclassify{\char'461}{\char'25}{\sy\char'61}}
\chcode'16+13
\def\theta{\fontclassify{\char'245}{\ttchar{\mi\char'45}}{\mi\char'45}}
\chcode'17+13
\def\llcorner{\fontclassify{\mathrel{\char'432}}{\ttchar{\sy\char'32}}{\sy\char'32}}
\chcode'20+13
\def\lrcorner{\fontclassify{\mathrel{\char'433}}{\ttchar{\sy\char'33}}{\sy\char'33}}
\chcode'21+13
\def\lshch{\fontclassify{\mathbin{\char'534}}{\ttchar{\sy\char'134}}{\sy\char'134}}
\chcode'22+13
\def\lll{\fontclassify{\mathbin{\char'533}}{\ttchar{\sy\char'133}}{\sy\char'133}}
\chcode'23+13
\def\llll{\fontclassify{\char'470}{\ttchar{\sy\char'70}}{\sy\char'70}}
\chcode'24+13
\def\exists{\fontclassify{\char'471}{\ttchar{\sy\char'71}}{\sy\char'71}}
\chcode'25+13
\def\@{\fontclassify{\mathbin{\char'412}}{\char'26}{\sy\char'12}}

\def\wedge{\fontclassify{\mathrel{\char'444}}{\ttchar{\sy\char'44}}{\sy\char'44}}
\chcode'27+13
\def\_{\fontclassify{\char'465}{\char'32}{\sy\char'65}}
\chcode'30+13
\def\rarrow{\fontclassify{\mathrel{\char'441}}{\ttchar{\sy\char'41}}{\sy\char'41}}

\def\lshch{\fontclassify{\mathrel{\char'430}}{\ttchar{\sy\char'30}}{\sy\char'30}}
\chcode'32+13
\def\lshch{\fontclassify{\mathrel{\char'434}}{\ttchar{\sy\char'34}}{\sy\char'34}}
\chcode'33+13
\def\le{\fontclassify{\mathrel{\char'424}}{\hbox{\sponse{\char'32}<}}{\sy\char'24}}

\def\ge{\fontclassify{\mathrel{\char'425}}{\hbox{\sponse{\char'32}>}}{\sy\char'25}}

\def\equiv{\fontclassify{\mathrel{\char'421}}{\ttchar{\sy\char'21}}{\sy\char'21}}
\chcode'36+13
\def\ve{\fontclassify{\mathbin{\char'537}}{\ttchar{\:z\char'137}}{\sy\char'137}}
\chcode'37+13
\def\#\{\fontclassify{\char'561}{\char'43}{\sy\char'161}}

\def\$\{\fontclassify{\char'577}{\char'44}{\sy\char'177}}

```

```

\def\%{\char'45}

\def\@{\fontclassify{\char'574}{\char'100}{\sy\char'174}}
\chcode'100+13
\def\|\{\fontclassify{\mathbin{\char'404}}{\char'134}{\sy\char'404}}

\def\up{\fontclassify{\mathrel{\char'442}}{\char'136}{\sy\char'42}}

\def\larrow{\fontclassify{\mathrel{\char'440}}{\char'137}{\sy\char'40}}

\def\lbrace{\fontclassify{\mathopen{\char'546610}}{\char'173}{\sy\char'146}}

\def\orbar{\fontclassify{\char'552614}{\char'174}{\sy\char'152}}

\def\rbrace{\fontclassify{\mathclose{\char'547611}}{\char'176}{\sy\char'147}}

\def \uparrow{\up$}

\def \sharp{\#}

\def \seal{{\stan70 S}}

\let \space=\ % for defining \ to be \hbox{\space} in \tt

\def\sp{\tt\char'40}

```

• Redefinitions of One Character Macros

```

\let \space=\ % for defining \ to be \hbox{\space} in \tt

\let \Mthinspace=\,
\let \Mopospace=\>
\let \Mthickspace=\;
\let \Mcondthinspace=\>
\let \Mnegthinspace=\!
\let \Mignorespace=\!
\let \Mnegthickspace=\?
\let \Mnegospace=\<
\let \Mnegcondthinspace=\<

% new long names work anywhere
\def \thinspace{\ifmode{\Mthinspace}\else{\Mthinspace}}
\def \opospace{\ifmode{\Mopospace}\else{\Mopospace}}
\def \thickspace{\ifmode{\Mthickspace}\else{\Mthickspace}}
\def \condthinspace{\ifmode{\Mcondthinspace}\else{\Mcondthinspace}}
\let \negthinspace=\Mnegthinspace
\def \negthickspace{\ifmode{\Mnegthickspace}\else{\Mnegthickspace}}
\def \negospace{\ifmode{\Mnegospace}\else{\Mnegospace}}
\def \negcondthinspace{\ifmode{\Mnegcondthinspace}\else{\Mnegcondthinspace}}

```

```
% redefine old names to match new names
\let \,=\thinspace
\let \>=\opSPACE
\let \;=\thickSPACE
\let \_>=\condthinspace
\let \?=\negthickSPACE
\let \<=\negopSPACE
\let \_<=\negcondthinspace
```

• **Make Some Math Things Work Anywhere**

```
% save old definitions
\let \Msection=\section
\let \Mdag=\dag
\let \Mddag=\ddag
\let \MP=\P
\let \Mcopyright=\copyright
\let \Msterling=\sterling
\let \Mbullet=\bullet
\let \Mcirc=\circ

% let these work in any mode using old math mode definitions
\def \section{\ifmode{\Msection}\else{\Msection}}
\def \dag{\ifmode{\Mdag}\else{\Mdag}}
\def \ddag{\ifmode{\Mddag}\else{\Mddag}}
\def \P{\ifmode{\MP}\else{\MP}}
\def \copyright{\ifmode{\Mcopyright}\else{\Mcopyright}}
\def \sterling{\ifmode{\Msterling}\else{\Msterling}}
\def \bullet{\ifmode{\Mbullet}\else{\Mbullet}}
\def \circ{\ifmode{\Mcirc}\else{\Mcirc}}
% Note that \$ is defined with the odd characters and @ now does the right
% thing in any mode as does \@
```

• **Page Numbering**

```
% uses two flags:
%   \indefer mode is T when in defer mode
%   \deferredpage is T when there is a piece of a page being deferred

\def\advpagecount{\if T\indefer mode{\advpagecountone \setcount0\highestpagenumber}
\else{\if T\deferredpage{\setcount0\savedpagecount
\gdef\deferredpage{F}}
\else{\advpagecountone \setcount0\highestpagenumber}
}
}
\def\deferredpage{F}
\def\indefer mode{F}

\def\incpagecount{\gdef\advpagecountone{\advcountq{\highestpagenumber}}}
\def\decpagecount{\gdef\advpagecountone{\setcount9\highestpagenumber
\advcount9by-1
\setq{\highestpagenumber}9}}
```

```

\def\setpagecount#1{\setcount9 #1
  \ifpos9{\incpagecount\advcount9 by -2}
  \else{\decpagecount\advcount9 by 2}
  \setq{\highestpagenumber}9
}

```

• **“output, Style, Format Routines**

```

\def\normal{\resetsize \fontsize \parstyle}
\def\resetsize{\normalhsize \normalvsize}
\def\everyoutput{} % this is something that is in every output routine

% start of format descriptions
\def\usebookformat{\input mbook }

\def\usebasicformat{\usebasicstyle \usebasicoutput }

% ACM oversize format for Versatec (camera ready copy)
\def\useacmformat{\input macacm }
% To use, code \useacmformat at the start of the paper.
% Then define \title and \authors
% then define the title portion, followed by \endoftitle
% when you are all done \endofpaper\end

\def\useplainoutput{\input oplain }

\def\usebasicoutput{\input obasic }

\def\useWoodsoutput{\input owoods }

\def\useblockstyle{\input sblock }

\def\usebasicstyle{\input sbasic }

% look at \useplainoutput and \useblockstyle for the minimum needed
% in output and style routines
% Format routines are simply output and style together. Note that
% other related macros and definitions may be included also.

```

• **Footnotes**

```

% normal footnote
\def\footnote#1#2{#1\botinsert{\eightpoint\hbox par size{#1#2}}}

% numbered footnote
\def\nfootnote#1{\advcountq{\footnotenumbe}!\
  $†{\footnotenumbe}$!\
  \botinsert{\eightpoint\hbox par size{†{\footnotenumbe}$#1}}}
\def\footnotenumbe{0}

```



```
\botsep{\vskip15pt \hrule width5pc\vskip 3pt}
```

```
% footnote mark characters
```

```
\def\upstar{\lower 3pt \hbox{${\hbox{*}}$}}
```

```
\def\dagger{\lower 2pt \hbox{${\Mdag}$}}
```

```
\def\ddagger{\lower 2pt \hbox{${\Mddag}$}}
```

• Paragraphs

```
\def\hangbox to #1 #2{\par\hangindent #1\noindent
  \hbox to #1{#2}\!}
```

```
\def\levelone#1{\hangbox to 20pt {#1\hfill}}
```

```
\def\leveltwo#1{\hangbox to 40pt {\hbox to 20pt{\hfill}}#1\hfill}}
```

```
\def\levelthree#1{\hangbox to 60pt {\hbox to 40pt{\hfill}}#1\hfill}}
```

```
\def\number#1{\levelone{#1}}
```

```
\def\nnumber#1{\hangbox to 50pt {#1\hfill}}
```

```
\def\indpar#1{\par
```

```
  \save9\hbox to size{}
```

```
  \save9\hbox{\box9\hskip-40pt} % width minus 40pt
```

```
  \hsize 1wd9
```

```
  \vskip1pt
```

```
  \leveltwo{ }\strut#1\strut}\par\normalhsize
```

```
  \vskip1pt}
```

```
\def \hdr#1{\par\goodbreak\yyskip\ctrline{\bf #1}\posthdrskip}
```

```
\def \posthdrskip{\par\badbreak\vskip 5pt\badbreak}
```

```
\def \sectionskip{\par\excellentbreak\vskip 24pt plus 12pt minus 6pt}
```

• List Definitions

```
\def \list#1{\xdef{\listcounter{#1}}}
```

```
\def \item{\advcountq{\listcounter}
```

```
  \levelone{\listcounter.}}
```

```
\def \itemindent{\levelone{}}
```

```
\def \bitem{\levelone{\hfill\bullet}} % this centers the bullet. see \levelone
```

```
\def \sublist#1{\xdef{\sublistcounter{#1}}}
```

```
\def \subitem{\advcountq{\sublistcounter} % leaves count in \count9
```

```
  \setcount9 -\sublistcounter % we want roman numerals
```

```
  \leveltwo{\count9.}}
```

```
\def \subitemindent{\leveltwo{}}
```

```
\def \subsublist#1{\xdef{\subsublistcounter{#1}} % should be a letter
```

```
\def \subsubitem{\advcountq{\subsublistcounter}
  \levelthree{\char\subsublistcounter.}}
```

```
\def \subsubitemindent{\levelthree{}}
```

• Underlining and Boxes

```
\def \undertext #1{\underline{\hbox{#1}}}$} % underline in horizontal mode
```

```
\def \overtext #1{\overline{\hbox{#1}}}$} % overline in horizontal mode
```

```
\def \leaderline{\leaders\hrule\hfill}
```

```
\def \boxit#1{\vbox{\hrule\hbox{\vrule\hskip3pt
  \vbox{\vskip3pt#1\vskip3pt}\hskip3pt\vrule}\hrule}}
```

```
\def \sizeboxit to#1by#2 #3{\vbox{\hrule\hbox to #1{\rule\hss
  \vbox to #2{\vss#3\vss}\hss\vrule}\hrule}}
```

```
\def \boxitnoglue#1{\vbox{\hrule\hbox{\vrule
  \vbox{#1}\vrule}\hrule}}
```

% Boxit and Boxitnoglue are like boxit and boxitnoglue except that horizontal
% and vertical modes are reversed.

```
\def \Boxit#1{\hbox{\vrule\vbox{\hrule\vskip3pt
  \hbox{\hskip3pt#1\hskip3pt}\vskip3pt\hrule}\vrule}}
```

```
\def \Boxitnoglue#1{\hbox{\vrule\vbox{\hrule
  \hbox{#1}\hrule}\vrule}}
```

% Lboxit puts L's around box instead of rules

```
\def \Lboxit to #1 by #2 #3{\def \hsplitrule{\hbox to #1{\vbox{\hrule width .25in}
  \hfill
  \vbox{\hrule width .25in}}}}
```

```
\def \vsplitrule{\vbox to #2{\hbox{\vrule height .25in}\vfill
  \hbox{\vrule height .25in}}}
```

```
\vbox{\lineskip Opt
  \baselineskip Opt
  \hsplitrule
  \vbox to #2{\hbox to #1{\vsplitrule
    \hfill
    \vbox to #2{\vfill#3\vfill}
    \hfill
    \vsplitrule}}
  \hsplitrule
}}
```

```
\def \type #1>#2{\par\indpar{\displayfont #1\under{#2}}} % type a line (as in dialogue)
% the second argument is underlined, good for prompts
```

```
\def \ttype #1>#2{\par\noindent{\displayfont#1\under{#2}}\par}
% type a line (as in dialogue)
```

 • Penalties

```

\def\badbreak{\penalty1000}

\def\goodbreak{\penalty-100}

\def\excellentbreak{\penalty-1000}

```

• "nofill" "endnofill"

```

% To use, code:
% \nofill
% statements
% \endnofill
%
% The code is listed verbatim without any page breaks.
% To allow a page break, put \allowbreak on a line. If there
% is no break, a blank line is generated.
%
% Note that \fontsize must be defined to be your normal size of type, such
% as \tenpoint
%
% Font is not changed

% Use of tabs in verbatim mode will give an error message.

% Define \<cr> to be \CR when enabled
\chcode'15+12\def\
{\CR}\chcode'15+5 %

\def\nofill{\parskip Opt
\chcode'11+13           % define tab to give an error
\chcode'15+13           % define <return> to generate \cr
\chcode'40+13           % define space to generate \<space> (a real space)
\gdef\ {\hbox{\space}} % make space exactly one unshrinkable space
\gdef\CR{\par\badbreak\noindent\hbox{\!\,}}}}

\def\endnofill{\par\badbreak % force glue to this page
\vskip-11pt
\chcode"11+10           % define tab to be a space
\chcode'15+5           % define <return> be a end of line
\chcode'40+10           % define space to be a space
\let\ =\space          % make "\ " as normal
\normal}

\def\goodgele{\chcode'34+13 % ≤
\let \≤=\le
\chcode'35+13           % ≥
\let \≥=\ge
}

```

```

\def\normalgele{\chcode'34+12 % ≤
\let \le=\negcondthinspace
\chcode'35+12 % ≥
\let \ge=\condthinspace
}

```

• Verbatim Mode "startcode and "endcode

```

% To use, code:
% \startcode
% statements
% \endcode
%
% The code is listed verbatim without any page breaks.
% To allow a page break, put \allowbreak on a line. If there
% is no break, a blank line is generated.
%
% \startcode supplies 4 pt of glue
% \endcode supplies 5 pt of glue
% The code is printed in \displayfont mode
% To avoid glue, code \startcodenoglu or \endcodenoglu

% Note that \fontsize must be defined to be your normal size of type, such
% as \tenpoint

% Use of tabs in verbatim mode will give an error message.

% Define \<tab> to be \tab when enabled
\chcode'11+12\def\    {\tab}\chcode'11+10
% will cause an error message unless \tab is defined

\def\startcodenoglu{\par
\displayfont
\nofill
\goodgele
}

\def\endcodenoglu{\endnofill
\normalgele
\fontsize
}

\def\startcode{\par\excellentbreak\vskip 5pt plus 1pt minus 1pt\startcodenoglu}
\def\endcode{\endcodenoglu\excellentbreak\vskip 6pt plus 1pt minus 1pt}

\def\startoutput{\par\excellentbreak\vskip 5pt plus 1pt minus 1pt{\tenpoint
$\down\qqquad\down\qqquad\down\qqquad\down\qqquad\down\qqquad\down\qqquad\down$\par}
\vskip 6pt plus 1pt minus 1pt}

\def\allowbreaknoglu{\par\badbreak\vskip-11pt\excellentbreak}

\def\allowbreak{\allowbreaknoglu\vskip 11pt plus 1pt}

```

•• Verbatim Mode Using `ftft` "halignftft"

```

% To use, code the following:
% \verbatim{
% follow with code
% } terminates verbatim mode.
% Note that \verbatim stuff will not be broken across page boundaries.

% To allow a break, use \noalign{\excellentbreak}%
% or \breakhere%
% Note the absence of spaces in the above.
% Note that the % is necessary to avoid an extra line generated.

% Note that \fontsize must be defined to be your normal size of type, such
% as \tenpoint

% These macros rely upon the definitions of \<cr> and \<tab> on the previous page.
% Use of tabs in verbatim mode will give an error message.

\def\verbatim{\nofill
\gdef\CR{\cr\noalign{\badbreak}}
\goodgele
\verbatimgenerate}

\def\verbatimgenerate#1{{\displayfont$$\halign to size{##\hfill\cr#1}$$}
\endnofill
\normalgele
}

\def\breakhere{\noalign{\excellentbreak\vskip 11 pt}}

\def\threecol{\nofill
\gdef\CR{\cr\noalign{\badbreak}}
\goodgele
\threecolgenerate}

\def\threecolgenerate#1{{\displayfont\halign{##\hfill@##\hfill@##\hfill\cr#1}}%
\endnofill
\normalgele
}

```

• Notes

```

% \sendnotes creates a list of entries which will be output when
% \putnotes is used. This should be at the end of the manuscript.
% use \initnotes to initialize notes

\def \initnotes{\input mnotes }

```

• **Index Macros**

```
\def \initindex{\input mindex }
% see TUGboat (Vol. 1, No. 1) for an index package.
```

• **Defer Mode**

```
\def\defer{\input defer }
```

• **Table of Contents**

```
\def\beginofc{\input mtofc }
```

• **Interesting Hacks**

```
\def\ifnull#1\then#2\else#3{\def\jnk{#1?}\if?#1{#2}\else{#3}}
% to use \ifnull #1\then<true clause>\else<false clause>

\def\bracex{\leaders\hrule height 1.5pt \hfill}
\def\dnbrace{${\char'772$\bracex${\char'775
  \char'774$\bracex${\char'773$}
\def\upbrace{${\char'774$\bracex${\char'773
  \char'772$\bracex${\char'775$}

\def \TEX{\hbox{\rm T\hskip-.1667em\lower.424ex\hbox{E}\hskip-.125em X}}

\def\blackslug{\hbox{\hskip 1pt \vrule width 4pt height 6pt depth 1.5pt
  \hskip 1pt}}

\def\boxtop#1{\save9#1\lower 1ht9\box9}

\def\topspace{(\hrule height0pt)\vskip}
  % e.g. "\topspace 1in" puts an inch of space at the top of a page

\def\setq#1#2{\ifpos#2{\gdef#1{}}
  \else{\gdef#1{-} \setcount#2 -\count#2}
  \xdef#1{#1\count#2}
  \setcount#2#1} % notice how we restore \count#2

\def\advcountq#1{\setcount9#1
  \advcount9by1
  \setq{#1}9}

\def\magnify#1{\chpar12=#1} % operand is magnification times 1000

\def\done{\par\vfill\end}
```

```

% To put a big capital letter begining a paragraph; #1 = indent for (2 or 3)
% lines, #2 = letter, #3 = paragraph
\def\capitalpar#1#2#3{\save9\hbox par size{\ragged 1000000
  \if2#1{{1 \linebreak 2}} % find out how much
  \else{{1 \linebreak 2 \linebreak 3}} % to move up
  \vbox{\hbox{\biggfont #2}
  \vskip -1ht9
  \save9\hbox{\biggfont #2}
  \hbox par size{\hangindent 1.3wd9 for #1{ }#3}}
}

\def\ie{{\sl i.e.}}
\def\eg{{\sl e.g.}}

```

• Default Options

```

\def\displayfont{\ninepoint\tt}

\setpagecount{1}

```

• Documentation of Use of Counters and Boxes

```

% Counters and use
% 0 the page number to appear on current page. Valid only in \output,\send,\mark
% 1 unused
% 2 unused
% 3 unused
% 4 unused
% 5 unused
% 6 unused
% 7 unused
% 8 unused
% 9 work value, use this for temporary calculations in a macro

% Boxes
% 0 unused
% 1 used by defer output and macacm
% 2 used by defer output and macacm
% 3 unused
% 4 unused
% 5 unused
% 6 unused
% 7 unused
% 8 unused
% 9 for temp macro use: \boxtop

```

```

% Files for send
% 0  index
% 1  notes
% 2  tofc
% 3  unused
% 4  unused
% 5  unused
% 6  unused
% 7  unused
% 8  unused
% 9  unused

```

The following section consists of external files that are only loaded when needed. As described in the text, this saves on the amount of space needed by these macros in "TEX" itself.

MBOOK.TEX

```

% Book Format
\def \bookoutput{\vbox to 9truein
  {\baselineskip Opt\lineskipOpt % beginning of output routine, resets skips
  \advpagecount % use the correct page number in \send
  \everyoutput
  \if T\tpage % the next is used when tpage is "T" (title pages)
    {\gdef\tpage{F} % reset tpage
    \vskip .7truein % blank space in place of headlines
    \page} % insert the page contents, no page #
  \else{\if T\index{\indexoutput}
  \else{\if T\pagenumberarea{\pagenumberregion\vfill}\else{}
    \page % insert the page contents
    \if B\pagenumberarea{\vfill\pagenumberregion}\else{}}
  }}} % end \bookOutput routine

\def \pagenumberarea{T} % T for Top of page, B for Bottom, else for none

\def \bookstyle{\maxdepth 2pt
  \parindent 20pt
  \parskip Opt plus 1 pt
  \lineskip 1pt plus Opt
  \topskip 24pt plus 6pt minus 10pt
  \botskip 15pt plus 3pt minus 9pt
  \topbaseline Opt
}

```



```

% page number definitions

\def\boxpagenumberregion{\moveleft .125truein\vbox to .7truein{\hrule
% horizontal rule at top of page
\hbox to 6.75truein{\trule
% 20pt*(1+sqrt(5))/2=32.361pt
\ifeven0{\hbox to 32.361pt{\cmrten\hfill\count0\hfill\trule}
\hfill\cmss12\topmark\hfill}
\else{\hfill\cmss12\botmark\hfill
\hbox to 32.361pt{\cmrten\trule\hfill\count0\hfill}}
\trule}
\hrule}} % horizontal rule under the headline

\def\trule{\vrule height 13.5pt depth 6.5pt} % used at top of page

\def\titlemark#1#2{\mark{\ifeven0{#1}\else{#2}}}

\def\pagenumberregion{\boxpagenumberregion}

% "global variables"
\def\tpage{F}
\def\index{F}

\def\titlepage{\gdef\tpage{T}} % \titlepage sets tpage to T

% enable book format

\def\usebookformat{\gdef\standardoutput{\output{\bookoutput}}
\standardoutput
\gdef\parstyle{\bookstyle}
\gdef\normalhsizel{\hsizel 6.5truein}
\gdef\normalvsize{\vsize 8.3truein}
\normal
}

\usebookformat

% chapter section

\def\chapternumber{0}

\def\chapterbegin#1{\par
\gdef\footnotenuml{0}
\advcountq{\chapternumber}
\gdef\sectionnuml{0}
\xdef\wholesectionnuml{Chapter \chapternumber}
\titlemark{\wholesectionnuml}{\sectionname}
\vfill\eject
\gdef\sectionname{#1}
\titlemark{\wholesectionnuml}{#1}
{\noindent \cmss12 \wholesectionnuml \ \ #1}
\if T\writetofc{\send2{\chapterto{#1}\count0.}}\else{}
\posthdrskip}

```

```

\def\dosectionbegin#1{\par
  \titlemark{\wholesectionnumber}{\sectionname}
  \sectionskip
  \gdef\sectionname{#1}
  \titlemark{\wholesectionnumber}{#1}
  {\tenpoint \bf \noindent $\bullet$\ \wholesectionnumber\ \ #1}
  \posthdrskip}

\def\sectionbegin#1{\advcountq{\sectionnumber}
  \gdef\subsectionnumber{0}
  \xdef\wholesectionnumber{Section \chapternumber.\sectionnumber}
  \if T\writetofc{\send2{\sectiontofc{#1}\count0.}}\else{}
  \dosectionbegin{#1}}

\def\subsectionbegin#1{\advcountq{\subsectionnumber}
  \gdef\dimsectionnumber{0}
  \xdef\wholesectionnumber{Section
    \chapternumber.\sectionnumber.\subsectionnumber}
  \if T\writetofc{\send2{\subsectiontofc{#1}\count0.}}\else{}
  \dosectionbegin{#1}}

\def\dimsectionbegin#1{\advcountq{\dimsectionnumber}
  \xdef\wholesectionnumber{Section
    \chapternumber.\sectionnumber.\subsectionnumber.\dimsectionnumber}
  \if T\writetofc{\send2{\dimsectiontofc{#1}\count0.}}\else{}
  \dosectionbegin{#1}}

\def\specialbegin#1{\titlemark{#1}{\sectionname}
  \vfill\ejct
  \settitle{#1}
  {\noindent \cmss12 #1}
  \posthdrskip}

\def\settitle#1{\par\titlemark{#1}{#1}
  \gdef\wholesectionnumber{#1}
  \gdef\sectionname{#1}}

\def \wholesectionnumber{}
\def \sectionname{}

% automatic table of contents generation

\def\inittofc#1{\open2 #1
  \gdef\writetofc{T} % write tofc info

\def\writetofc{F}

```

MACACM.TEX

```

% ACM two column format for Versatec

\def\acmoutput{\everypage
\if T\tpage
  {\if T\column
    {\gdef\normalhsz{\hsz 4.25truein}
    \gdef\normalvsize{\vsz 8.9truein}
    \normalhsz\normalvsize
    \save1\page\gdef\column{L}
    }
  \else{\if L\column
    {\save2\page\gdef\column{R}}
    \else{\vbox to 11.9truein{\box1\vskip -1000pt plus 1000000pt
      \hbox to 9 truein{\box2\hfill\page}}
      \advcount 0
      \gdef\column{L}
      \gdef\tpage{F}
      \gdef\normalvsize{\vsz 11.5truein}
      \normalvsize
    }}}
\else{\if L\column
  {\save2\page\gdef\column{R}}
  \else {\vbox to 11.9 truein{\hbox to 9truein{\ninepoint\ifeven0
    {\rm\lastnames\hfill\sl\title}
    \else{\sl\title\hfill\rm\lastnames}}
    \vfill
    \hbox to 9 truein{\box2\hfill\page}}
    \advcount0
    \gdef\column{L}
  }}}

\def \acmstyle{\maxdepth 2pt
  \parindent 20pt
  \parskip 0pt plus 1 pt
  \lineskip 1pt plus 0pt
  \topskip 24pt plus 6pt minus 10pt
  \botskip 15pt plus 3pt minus 9pt
  \topbaseline 0pt
}

\def\endoftitle{\par\vfill\ejct}

\def\endofpaper{\par\vfill\if L\column{\ejct\hbox{} \vfill}\else{}}

% To use, code \useacmformat at the start of the paper.
% Then define \title and \authors
% then define the title portion, followed by \endoftitle
% when you are all done \endofpaper\end

% enable acm format

\def\standardoutput{\output{\acmoutput}}

```

```

\def\useacmformat{\standardoutput
  \gdef\parstyle{\bookstyle}
  \gdef\normalhsize{\hsize 9truein}
  \gdef\smallhsize{\hsize 9truein} % too small to indent right
  \gdef\normalvsize{\vsize 3truein}
                                % these sizes reuefined in \endoftitle
                                % and \acmoutput

  \normal
  \gdef\tpage{T}
  \gdef\column{T}
}

\useacmformat

```

OPLAIN.TEX

```

% Plain Output routine

\def\plainoutput{\advpagecount % use the correct page number in \send
  \page
  \everyoutput}

\def\standardoutput{\output{\plainoutput}}

\def\useplainoutput{\standardoutput
  \gdef\normalhsize{\hsize 6.5truein}
  \gdef\normalvsize{\vsize 9truein}
  \normal
}

\useplainoutput

```

OBASIC.TEX

```

% Basic output routine

\def\basicoutput{\advpagecount % use the correct page number in \send
  \vbox to 9truein{\page
    \vfill
    \ctrline{\carten \count0}}
  \everyoutput}

\def\standardoutput{\output{\basicoutput}}

\def\usebasicoutput{\standardoutput
  \gdef\normalhsize{\hsize 6.5truein}
  \gdef\normalvsize{\vsize 8.75truein}
  \normal
}

```

```
\usebasicoutput
```

OWOODS.TEX

```
% Woods output
% (To look like previous versions of the annual report.)

\def\Woodsoutput{\advpagecount
  \vbox to 9truein{\ctrline{\ninepoint\sl -\count0-}
    \vfill
    \page}
  \everyoutput}

\def\standardoutput{\output{\Woodsoutput}}

\def\useWoodsoutput{\standardoutput}
  \gdef\normalhsize{\hsize 6.5truein}
  \gdef\normalvsize{\vsize 8.75truein}
  \normal
}

\useWoodsoutput
```

SBLOCK.TEX

```
% Block Style

\def\blockstyle{\maxdepth 2pt
  \parindent 0pt
  \parskip 6 pt plus 6 pt minus 2 pt % Skip a line between paragraphs.
  \lineskip 1pt plus 0pt
  \topskip 24pt plus 6pt minus 10pt
  \botskip 15pt plus 3pt minus 9pt
  \topbaseline 0pt
}

\def\useblockstyle{\gdef\parstyle{\blockstyle}
  \normal
}

\useblockstyle
```

SBASIC.TEX

```

% Basic Style

\def\basicstyle{\maxdepth 2pt
  \parindent 20pt
  \parskip 0pt plus 1 pt
  \lineskip 1pt plus 0pt
  \topskip 24pt plus 6pt minus 10pt
  \botskip 15pt plus 3pt minus 9pt
  \topbaseline 0pt
}

\def\usebasicstyle{\gdef\parstyle{\basicstyle}
  \normal
}

\usebasicstyle

```

MNOTES.TEX

```

% notes

\open1=fixnot.tex

\def \sendnotes#1{\send1{Page \count0. #1\par}}

\def \putnotes{\specialbegin{Fixup Notations}
  \open1=dummy1.tmp      % Close the fixnot file
  \input fixnot.tex      % Now put text here.
}

```

DEFER.TEX

```

% defernode based on that written by Brent Hailpern and Jim Boyce

% box 1 is slop on current page
% box 2 is extra slop on current page that will go on following page

\def\defer{\save2\ vbox{}      % no extra slop yet
  \output{\save1\page\output{\save2\page}} % cause stuff to be saved
  \eject                      % flush out current page
  \standardoutput
  \ifdimen 1ht2>0pt{\unbox1\save1\box2}\else{} % comment below
  % put out full page and copy partial page
  \if F\deferredpage{\gdef\deferredpage{T}
    \advppagecountone
    \savethepagecount
  }\else{}
}

```

```

    \gdef\indefermode{T}
}

\def\enddefer{\eject
  \unbox1
  \gdef\indefermode{F}
}

\def\savethepagecount{\setq{\savedpagecount}9}

\defer % do it this time too!

```

MTOFC.TEX

```

% table of contents

\def\beginofc{\gdef\chapternumber{0}
  \setpagecount{-1} % initial page number for cover page
  \specialbegin{Table of Contents}}

\def\chapterto#c1#2.{\par
  \advcountq{\chapternumber}
  \gdef\sectionnumber{0}
  \hbox to size{\hbox to 30pt{\bf\chapternumber\hfill}}{#1}
  \leaders\hrule\hfill\hbox to 20pt{\hfill#2}}

\def\sectionto#c1#2.{\par
  \advcountq{\sectionnumber}
  \gdef\subsectionnumber{0}
  \hbox to size{\hbox to 45pt{\bf\chapternumber.\sectionnumber\hfill}}{#1}
  \leaders\hrule\hfill\hbox to 20pt{\hfill#2}}

\def\subsectionto#c1#2.{\par
  \advcountq{\subsectionnumber}
  \gdef\dimsectionnumber{0}
  \hbox to size{\hbox to 60pt{\bf
    \chapternumber.\sectionnumber.\subsectionnumber\hfill}}{#1}
  \leaders\hrule\hfill\hbox to 20pt{\hfill#2}}

\def\dimsectionto#c1#2.{\par
  \advcountq{\dimsectionnumber}
  \hbox to size{\hbox to 75pt{\bf
    \chapternumber.\sectionnumber.\subsectionnumber.\dimsectionnumber
    \hfill}}{#1}
  \leaders\hrule\hfill\hbox to 20pt{\hfill#2}}

\def\endtofc{\par\vfill\eject % put out this page before screwing up page #
  \gdef\chapternumber{0}
  \setpagecount{0}
}

\beginofc % do it now too!

```