

TEXTIPS^s
Supplementary information about L^AT_EX 2_ε
Second Edition

James W. Howse, Los Alamos National Laboratory,
Neill E. Doren, Sandia National Laboratories

February 16, 2000

This article is a repository of helpful advice concerning the use of T_EX and some of the more common macro languages for it, specifically L^AT_EX 2_ε and $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX. Currently these pages contain the knowledge of only one person¹, but it is to be hoped that others will add to this as time goes on. For those of you using T_EX for the first time, there are several things which are important to keep in mind. First of all, it is very important to understand that T_EX is NOT a word processor. It is nothing like WORD[®] or WORDPERFECT[®] in that it has no built-in text editor, page previewer, spelling checker, or printer drivers. T_EX was written in the dark ages of computing before these additions were believed to be of any use. It is much more helpful to think of the commands in T_EX as representing a programming language in which the user describes the desired document layout in a procedural manner. The program `tex` compiles this language into a machine *independent* binary file which is denoted by the extension `.dvi`. L^AT_EX 2_ε and $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX are macro packages for T_EX whose commands are written in the T_EX language. Second, as one sage was heard to say, “L^AT_EX is just a hack”². For the most part, no one wants to program in T_EX, so higher level languages, such as L^AT_EX 2_ε and $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX, were developed from it that are hopefully easier to learn and use. The difficulty for the end user is that some of the commands in these languages implicitly contain very specific concepts of how a page of text should be laid out. This presents no problem so long as the user’s and the program author’s concepts of proper page layout agree. However, if they do not, it is in some cases extremely difficult to obtain a specific page format.

NOTE: The documentation files described herein can be downloaded via the web from from the site http://www.math.unm.edu/~nedoren/latex/textips_docs. However, they may already be present on your local system. Ask your administrator for their location, or for help in installing these utilities and documents, if required.

As was previously mentioned, T_EX does not have a built-in text editor, page previewer, spell checker, or output device drivers. However, there are programs available which perform each of these complementary functions. A list of public domain programs for this purpose is documented in Table 1. In addition to document formatting, `emacs` has editing modes for handling e-mail, writing programs, and a huge number of other things. Although `xdvi` and `xtex` are both `.dvi` file previewers, they have somewhat different features. `xdvi` handles POSTSCRIPT far better than `xtex`, particularly if POSTSCRIPT fonts are used in the main body of the document. Also the screen fonts of `xdvi` have a nicer appearance than those of `xtex`. On the other hand, `xtex` has some useful features that `xdvi` lacks. For example, `xtex` has a mouse driven ruler, which is very helpful if your document must conform to some type of page specification. This is common practice both for articles in conference proceedings and for masters theses and doctoral dissertations. Furthermore, `xtex` has commands that can be embedded in your document which display a red box around all cross references in the document. Pressing the mouse button within any one of these boxes takes you to the occurrence of that cross reference. For instance, at a bibliographic citation, you can press the mouse button in the red box around the reference number and you are taken to the point in the bibliography where the actual citation appears. Releasing the mouse button returns you to your original position in the document. Also `xtex` keeps track of the page number as part of its interface, which can be very helpful if the document has no page numbers, or it is inconvenient to be constantly moving to the part of the page on which they are located. Note that neither of these programs displays documents containing color properly. In order to preview such documents, convert the `.dvi` file to a POSTSCRIPT `.ps` file by executing the command `dvips file -o`, then display the resulting `.ps` file with `ghostview`. By default, the output of the POSTSCRIPT translator `dvips` is sent to the default printer for

¹specifically, James Howse

²attributed to Bill Horne

Programs to complement T <small>EX</small>	
<i>Program</i>	<i>Purpose of the program</i>
dvips	Translates the compiled output of tex (i.e., the binary .dvi file) into the POSTSCRIPT page description language (i.e., the ASCII .ps file). This program has both a man page and external documentation, which is in the file dvips-5.58.dvi .
xdvi xtex	Both are X-Windows screen previewers which provide an almost WYSIWIG display of a compiled tex output file (i.e., the binary .dvi file). The only documentation for either of these programs is obtained by using the command man program . These man pages can be printed using the command man -t program . Note that xtex has <i>not</i> been kept up to date by its authors.
ghostview	An X-Windows screen previewer which provides an almost WYSIWIG display of a POSTSCRIPT file (i.e., the ASCII .ps file). The only documentation for this program is obtained by using the command man ghostview . These man pages can be printed using the command man -t ghostview .
ps2pdf DISTILLER TM pdftops	Programs for converting POSTSCRIPT generated by dvips into (and back from) ADOBE [®] PDF format. The Aladdin ps2pdf and Derek Noonburg's pdftops programs are freeware converters for both the Unix [®] and MS-WINDOWS TM platforms. DISTILLER TM is available from ADOBE [®] for MS-WINDOWS TM (at a significant cost). It has been observed that DISTILLER TM produces a tighter (smaller) output PDF file, with no loss of output quality. Find pdftops at http://www.foolabs.com/xpdf/ .
ispell	A spelling checker which is able to filter out most T <small>EX</small> and L <small>A</small> T <small>EX</small> 2 _ε commands. This program also has non-English dictionaries and allows the user to build a personal dictionary. This program has both a man page and external documentation, which is in the file ispell-3.1.dvi .
bibcard	An X-Windows utility for simplifying the often daunting task of submitting bibliographic entries to the B <small>Y</small> B <small>T</small> E <small>X</small> database. Pull-down menus for book, article, thesis, conference and technical report citations are at the touch of the mouse. The source code for this freeware program compiles easily on most workstations and is available from ftp.iam.unibe.ch in directory /pub/X11/Bibcard-1.0.tar.Z .
emacs	A text editor that has a number of different editing modes, each well suited to a particular task. The editor has an extensive array of features, all of which can be customized by the user. Arguably the best text editor available on a Unix [®] platform. This program has both a man page and an <i>extensive</i> interactive help facility. The external documentation is in the file emacs-19.22.dvi .
AUC <small>T</small> E <small>X</small>	This is not a program, but rather a macro package for emacs . It defines a new T <small>EX</small> /L <small>A</small> T <small>EX</small> 2 _ε mode for emacs that allows all of the above programs to be used from within a single integrated environment. Furthermore, it defines power keys which expedite the entry of many T <small>EX</small> and L <small>A</small> T <small>EX</small> 2 _ε commands. Those of you interested in this package will probably have to install it yourself. Documentation and installation instructions are in auctex-9.3.dvi . The ftp site is sunsite.auc.dk in /packages/auctex/ .

Table 1: Auxiliary programs which complement TEX.

the computer. The printer that the output is sent to, can be changed for the duration of a login session, by executing the command **setenv PRINTER printer before** running **dvips**. To alter the printer used for

only one print job, use the command `dvips -Pprinter file`.

Earlier, the TEX macro packages L^AT_EX 2_ε and $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX were mentioned. The commands available in L^AT_EX 2_ε are discussed in [Lam94]. It can be purchased for about \$36, or try to borrow a copy of it from the system administrators, a professor, or a fellow student. $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX is a macro package, written under commission for the American Mathematical Society, whose commands are a superset of L^AT_EX 2_ε's. Some of the additional features of $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX are

- 1) Seven new alignment formats for equations. For example, these alignment formats allow the following two constructions which would be rather difficult in L^AT_EX 2_ε. The first construction is

$$\begin{aligned}\varphi(x, z) &= z - \gamma_{10}x - \sum_{m+n \geq 2} \gamma_{mn}x^m z^n \\ &= z - Mr^{-1}x - \int_2^\infty Mr^{-\tau}x^{(\tau-n)}z^{(\tau-m)}d\tau \\ \zeta^0 &= (\xi^0)^2,\end{aligned}\tag{1}$$

$$\zeta^1 = \xi^0\xi^1 + \xi^1\xi^2 + \xi^0\xi^2,\tag{2}$$

$$\zeta^2 = (\xi^1)^2 + \xi^0\xi^1.\tag{3}$$

The second construction is

$$x_0 = y_0 \qquad \text{by (A.3)}\tag{4}$$

$$x_1 = y_1 + 5 \qquad \text{by (A.11)}\tag{5}$$

$$x_0 + x_1 = y_0 + y_1 + 1 \qquad \text{by Axiom 1.}\tag{6}$$

- 2) Two new math environments, **cases** and **matrix**. The **cases** environment allows the construction of equations such as

$$P_{r-j} = \begin{cases} 0 & \text{if } r-j \text{ is odd,} \\ r!(-1)^{\frac{r-j}{2}} & \text{if } r-j \text{ is even.} \end{cases}\tag{7}$$

The **matrix** environment greatly reduces the number of commands that must be entered in order to create a matrix.

- 3) A **proof** environment and three different theorem styles; **theorem**, **definition** and **remark**.

Complete documentation of all the features of $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX is in the user's manual which is in the file `amsldoc.dvi`.

The problem with both L^AT_EX 2_ε and $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX is that some of the commands make implicit assumptions about the layout of the text. If the user wants the text to appear differently, this can sometimes be a *BIG* problem. Also there are functions that are not in either L^AT_EX 2_ε or $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX which would, on various occasions, be extremely useful. One solution to these problems is for the user to try to coerce L^AT_EX 2_ε or $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX into doing what he wants. This works very well for certain problems, and not at all for others. Another solution is to revise or extend the existing commands. One way of doing this is to write your own commands. While this is a potentially interesting adventure, in many cases it is likely to be quite time consuming, as most commands to accomplish anything of moment must be written in

TEX. Another approach which is usually less time consuming is to use revisions or extensions that others have written. There are a large number of these available, so before writing your own functions, it may be helpful to see whether one already exists. The first thing to check if you have a problem that is not resolved by [Lam94] is the TEX-FAQ (i.e., Frequently Asked Questions). This is available via the World Wide Web at the site <http://www.cogs.susx.ac.uk/cgi-bin/texfaq2html?introduction=yes>. It is also posted monthly to two bulletin boards, comp.text.tex and news.answers. It can also be obtained by anonymous ftp from the server rtfm.mit.edu in the directory /pub/usenet/news.answers/tex-faq. The next place to look for solutions to your problems is [GMS94]. This book discusses many of the packages that extend or modify the basic capacities of LATEX 2_ε. This book is excellent because it is laid out based on the user's desired outcome, rather than being a reference manual. For instance there is an entire chapter describing how to make floating bodies behave in specific ways.

The revisions and extensions of LATEX 2_ε are called packages. Note that $\mathcal{A}\mathcal{M}\mathcal{S}$ -LATEX is merely a set of packages for LATEX 2_ε. Packages are read into a document using the command `\usepackage[package options]{package name}`. These commands are usually entered immediately after the command `\documentclass[class options]{class name}`, which is on the first line of the document. Generally speaking, the *class name* determines the overall format of the document, and *package name* determines specific aspects of the document format. The *class options* and *package options* are flags which select between specific format choices in the class or package. These options are a list of terms separated by commas and containing **no** blank spaces. In LATEX 2_ε the extensions and revisions that have been written by others are designed to be entered as packages. These packages are usually denoted by files whose names end in `.sty`. Packages whose file name ends in `.sty` can be loaded using the `\usepackage{.}` command *without* this extension. The following list describes some useful styles for LATEX 2_ε. They are invoked with the `\usepackage{package name}` command unless otherwise indicated.

$\mathcal{A}\mathcal{M}\mathcal{S}$ -LATEX: As stated previously $\mathcal{A}\mathcal{M}\mathcal{S}$ -LATEX is a package written for the American Mathematical Society which greatly expands the mathematical capacities of LATEX 2_ε. Most of the useful features are loaded using the command `\usepackage{amsmath,amsbsy,amsopn,amsthm,upref}`. Complete documentation is contained in the file `amsl.doc.dvi`.

endfloat.sty: Many journals require that all tables and figures appear at the end of the document, each on a separate page. Further, a list of both tables and figures is often required. This style file removes tables and figures from within a document, transfers them all to the end, indicates where they were in the original, puts them all on separate pages, and compiles separate lists of tables and figures. Note that this package should only be loaded *after* your document has been finished with all of the figures in their correct locations. Documentation is found in the file `endfloat.dvi`.

theapa.sty & theapa.bst: These two files define a bibliographic citation format that is used by the American Psychology Association (APA). The biggest difference between this format and the standard LATEX 2_ε one is that rather than citing references by number from the bibliography, they are cited alphabetically by the first author's last name and the year of publication. This citation format is required by some journals, for instance *Neural Networks*. The first file defines the way citations appear in the main body of the paper, the second defines the way that they appear in the bibliography. In order to use this format, the commands `\usepackage{theapa}` and `\bibliographystyle{theapa}` must be placed into your document. The documentation is in the file `theapa.doc`.

fancyheadings.sty: This file defines a format to customize the headers and footers on each page. Both

the headers used in the [Lam94], and the headers and footers in this document, are examples of the sort of customization that can be done. The package is documented in the file `fancyheadings.dvi`.

fancybox.sty: This file defines commands which allow a box to be drawn around virtually anything. Equations, lists, floats, and entire pages are among the things that can be boxed. The box on the title page of this document is an example of what can be done with this package. The documentation for the package is in the file `fancybox.dvi`, and it contains many useful examples of how to box various things.

merge.sty: This file defines a format for producing form letters by merging addresses, openings and closings into a standard main body. Note that the file containing the addresses, etc., can *not* contain *any* blank lines, even at the end of the file. The package is documented in the file `merge.doc`.

setspace.sty: This package provides commands to set arbitrary line spacing in documents. Professors often request double spacing for drafts of papers, as it gives them more space in which to write encouraging remarks. The package provides three commands, `\doublespacing`, `\onehalfspacing`, and `\singlespacing`, for setting the overall spacing for the document in the preamble. There are also three environments with the same names which allow the spacing to be changed within the document. If a different spacing is required initially, then the `\setstretch{spacing multiplier}` command can be used in the preamble to set `baselinestretch` appropriately. The package is documented in the file `setspace.doc`.

subequations, **subfigure.sty**: The first list entry is an environment that is contained in $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, and the second entry is a package. These environments allow related equations or figures to be given the same reference number followed by different letters. Note that the individual members of the subequation or subfigure can be referenced separately by using a different `\label{name}` command for each member. The equation

$$\zeta^0 = (\xi^0)^2, \tag{8a}$$

$$\zeta^1 = \xi^0 \xi^1 + \xi^1 \xi^2 + \xi^0 \xi^2, \tag{8b}$$

$$\zeta^2 = (\xi^1)^2 + \xi^0 \xi^1, \tag{8c}$$

is an example of the use of the **subequations** environment. The **subfigure** package defines the command `\subfigure[caption text]{figure name}`. Note that each of the subfigures may have a different caption. The **subequations** environment is documented in the file `amsldoc.dvi`, and the **subfigure** package is documented in `subfigure.dvi`

array.sty: This package expands the column formatting options available in the **array** and **tabular** environments. For instance, columns whose entries are *vertically* centered, or columns whose entries are assumed to be mathematical expressions are easily created. Furthermore it allows the vertical line to be replaced by an arbitrary command. The package is documented in the file `array.dvi`.

tabularx.sty: This package is an extension of the **array.sty** package. It implements a version of the **tabular** environment in which the total width of the table is specified, and the widths of some columns are adjusted in order to achieve this overall width. Note that this is *different* from the **tabular*** environment. The package is documented in the file `tabularx.dvi`.

hhline.sty: This package defines the command `\hhline{options}` which is similar to `\hline` except that resulting horizontal line's interaction with vertical lines can be controlled by the user with the *options* list. The package is documented in the file `hhline.dvi`.

enumerate.sty: This package redefines the standard `enumerate` environment so that it has an optional argument which specifies the style of the list counter. The package is documented in the file `enumerate.dvi`.

caption.sty: This package redefines the standard `\caption{caption text}` command so that the caption format can be customized by the user. For instance, the package allows the font size used in the caption to be changed, and allows the caption text to be hung from the caption label. The package is documented in the file `caption.dvi`. The package has an unusual bug, the length `\captionmargin` can *not* be specified in inches, all other units appear to work.

twocolumnconf.cls: This item is actually a new document class written by James Howse, rather than a package. This class defines a two column format which conforms to that required by many conferences for proceedings submissions. This class is selected with the command `\documentclass[options]{twocolumnconf}`. The major difference between it and the standard two column format is that in this format the abstract spans both columns, the section headings are not numbered, and the pages have no numbers on them. The title section is entered exactly as in [Lam94, pages 181–182] except that the command `\begin{abstract} text \end{abstract}` occurs *before* the `\maketitle` command. This class has the same *options* as the standard `article` class in [Lam94, page 177], *except* the `notitlepage` | `titlepage` and `onecolumn` | `twocolumn` options do not exist. Also the `final` | `draft` option has been modified so that under the `draft` option the page numbers appear at the bottom of each page, and under `final` the pages have no numbers. The amount that the abstract text is indented is controlled by the length `\abstractmargin`. The default value of `0.5in` can be changed with the command `\setlength{\abstractmargin}{length}`.

unmeethesis & unmeereport: These two items are actually new document classes written by James Howse, rather than packages. The `unmeethesis` class conforms to the dissertation and thesis requirements of the Office of Graduate Studies at the University of New Mexico³. The `unmeereport` class is intended to produce well-formatted technical reports for the Department of Electrical Engineering at the University of New Mexico. Both of these classes are documented in the file `styles.ps` (POSTSCRIPT) or `styles.pdf` (ADOBE® PDF), available through the UNM Office of Graduate Studies (OGS). Or alternatively, on the mirror website <http://www.math.unm.edu/~nedoren/latex> in the subdirectory `stylesheets`.

Note that this list contains only a small number of the packages available for LATEX 2_ε. For the documentation of additional packages, ask your system administrator.

There are two major issues not covered by the above list of style options. They are the construction and inclusion of figures in TEX documents, and the use of POSTSCRIPT fonts in TEX documents. There are two possible ways to put figures into TEX documents. The first is to use an auxiliary program to draw the figure and then use some package to incorporate the figure into the document. The second approach is to use some set of macros to draw the figures within TEX. The advantage of the first approach is that it allows almost any type of drawing to be created fairly quickly. TEX has an *extremely* limited ability to perform mathematical computations, so drawing arbitrary shapes, particularly graphs of mathematical

³Although compliance is not guaranteed, the authors' dissertations were accepted without difficulty.

functions, is generally very slow at best. Also, since TEX was not designed for figure drawing, the figure must be drawn by command entry, and there is no interactive way to see what is being drawn. The advantage of the second approach is that all of TEX's formatting capabilities can be used to annotate the figure. This means that mathematical equations and symbols can be used in the figure. It is generally not possible to do much, if any, mathematical annotation in a drawing program. Also, in this scheme the type font of the figure and document will be identical. A package called **psfrag** gives the best of both worlds by allowing *any* text string in a POSTSCRIPT figure to be replaced by *any* LATEX 2_ε string. Table 2 lists some public domain drawing programs, and covers several methods for incorporating their output into TEX documents.

The packages **epsfig**, **graphicx**, **graphics**, and **psfrag** can only use POSTSCRIPT files that follow the bounding box comment convention. This is a comment of the form `%%Bounding Box: bblly bblly bburx bbury`. The four numbers specify the lower left and upper right *x* and *y* coordinates in points. If this comment does not appear at the beginning of your POSTSCRIPT document, then you must specify the bounding box by hand in order to use any of these packages. The packages **epsfig**, **graphicx**, and **graphics** all allow scaling, rotation, and clipping of the POSTSCRIPT image. POSTSCRIPT files incorporated using any of these packages can be previewed with either **xdvi** or **xtex**. Alternately the **.dvi** can be converted to POSTSCRIPT using **dvips**, and the resulting **.ps** file previewed using **ghostview**, but this introduces two additional steps into document processing. If your favorite drawing program does not output POSTSCRIPT, then its output must be converted. The program **xv** can convert **.gif**, **.tif**, and **.jpg** formats to POSTSCRIPT. By the way, this program has a number of other useful functions for image processing work. As stated earlier, there are also packages of macros for drawing within TEX. These packages are briefly discussed in Table 3.

The final issue to discuss is the inclusion of POSTSCRIPT fonts in TEX documents. If you are using LATEX 2_ε, then changing fonts in the main body of the document is actually quite easy. The general font selection procedure is described in detail in the document **fontguide.dvi**. This font selection scheme, which is often called **nfss**, was developed by Mittelbach and Schöpfung for **AMS-LATEX**, and was recently incorporated into LATEX 2_ε. With the introduction of this font selection mechanism, accessing both the original TEX fonts and POSTSCRIPT fonts has become much easier. For instance, the Concrete fonts developed by Donald Knuth are accessed with the command `\usepackage{concmath}`. The naming convention used in LATEX 2_ε for POSTSCRIPT fonts is described in the document **fontname.dvi**. The **psnfss** package provides numerous style files which redefine the default fonts to POSTSCRIPT fonts. The fonts Avant Garde, Bookman, Helvetica, New Century Schoolbook, Palatino, Times Roman, Utopia, and Zapf Chancery all currently work. These fonts are made the default font in a document using the commands `\usepackage{avant}`, `\usepackage{bookman}`, `\usepackage{helvet}`, `\usepackage{newcent}`, `\usepackage{palatino}`, `\usepackage{times}`, `\usepackage{utopia}`, and `\usepackage{chancery}`, respectively. The file **PSFonts/ps_font_test.dvi** contains complete character tables and tests of the available series and shape variations for all of these POSTSCRIPT fonts and for all of the original TEX fonts. Note that character tables and a great deal of other information can be obtained for *any* font by copying the file **/usr/local/texmf/lib/tex/latex/inputs/testdoc/nfssfont.tex** into a personal directory, executing the command `latex nfssfont.tex`, and following the subsequent instructions. The real difficulty is incorporating POSTSCRIPT math fonts into a document. The package **mathtime.sty**

⁴The author prefers **xfig** to **idraw**, and has very little experience with **tgif**.

⁵The author recommends either **gnuplot** or **xmgr**.

⁶The author feels that PTCTEX and Xy-pic are far superior to the **picture** environment and its extensions.

⁷The author has never used either of these packages.

Incorporating External Drawings into T _E X	
<i>Auxiliary Drawing Programs for T_EX</i>	
<i>Program</i>	<i>Description of the program</i>
xfig tgif idraw ⁴	All of these are interactive drawing programs. They are all designed for free hand type drawing. None are suitable for high precision drawing such as CAD, for plotting sets of data, or for plotting mathematical functions. All are capable of POSTSCRIPT output. Essentially they are all more or less imitations of Mac [®] drawing programs, such as Canvas [®] , designed to work under X-Windows. The only documentation for any of these programs is obtained by using the command <i>man program</i> . These man pages can be printed using the command <i>man -t program</i> .
dataplot gnuplot xmgr xprism2 xprism3 ⁵	All of these programs are designed to plot lists of data points and mathematical functions. <i>xmgr</i> , <i>xprism2</i> , and <i>xprism3</i> are menu-driven packages with built-in X-Windows interfaces. <i>dataplot</i> and <i>gnuplot</i> are command-driven packages whose plots appear in separate windows. Both of these packages have auxiliary X-Windows interfaces built on Tcl/Tk. <i>xprism2</i> is for 2-dimensional plots, while <i>xprism3</i> produces 3-dimensional plots. <i>xmgr</i> , <i>dataplot</i> , and <i>gnuplot</i> can create either 2 or 3-dimensional plots. All are capable of POSTSCRIPT output. The only documentation for <i>xmgr</i> , <i>xprism2</i> , and <i>xprism3</i> is that available interactively via the help menu. Both <i>dataplot</i> and <i>gnuplot</i> have man pages and external documentation. For <i>gnuplot</i> , the manuals are in the files <i>gnuplot-3.5.dvi</i> , <i>gnuplot_tutor.dvi</i> , <i>gnuplot_refcard.dvi</i> , and <i>xgnuplot.doc</i> .
<i>Packages to Incorporate Drawings into T_EX</i>	
<i>Package</i>	<i>Description of the package</i>
epsfig.sty graphicx.sty graphics.sty	All of these packages are designed to incorporate POSTSCRIPT files into a T _E X .dvi output file. They are entered as packages with the command <code>\usepackage{package name}</code> . To enter a centered figure into a document with <i>epsfig.sty</i> , use the command <code>\centerline{\psfig{figure=file,options}}</code> at the desired location in the document. For <i>graphicx.sty</i> and <i>graphics.sty</i> the required command is <code>\centerline{\includegraphics[options]{file}}</code> . The documentation for all three of these packages is in the file <i>grfguide.dvi</i> .
psfrag.sty	This package allows <i>any</i> text string in a POSTSCRIPT figure to be replaced by <i>any</i> L ^A T _E X 2 _ε construction. This allows graphs and drawings produced by other packages to be annotated with equations or other scientific text from L ^A T _E X 2 _ε . The package is invoked with the command <code>\usepackage{psfrag}</code> . The figure text is replaced by the L ^A T _E X 2 _ε text with the command <code>\psfrag{figure text}{L^AT_EX 2_ε text}</code> . In order for this command to work the figure must be preprocessed with the command <i>ps2frag file</i> . The documentation for this package is in the file <i>pfpguide.ps</i> , and there is a man page for <i>ps2frag</i> .

Table 2: Programs and packages for incorporating external drawings into T_EX.

replaces some of the normal T_EX math symbols with their POSTSCRIPT equivalents. There are two principle problems with using POSTSCRIPT math symbols. The first is that not all of the math symbols in L^AT_EX 2_ε and $\mathcal{M}\mathcal{S}$ -L^AT_EX are available in the public domain POSTSCRIPT fonts. This means that the two sets of fonts must be combined in order to retain all of the symbols. The second is that the math symbols in T_EX are accessed by commands of the form `\symbol` (e.g., `\gamma`, or `\sum`), so these

Producing Drawings in T <small>EX</small>	
<i>Package</i>	<i>Description of the package</i>
<p>picture epic.sty eepic.sty</p>	<p>picture is picture drawing environment that is built in to L<small>A</small>T<small>EX</small> 2_ε. It is very low level and requires that explicit coordinates be specified for <i>every</i> object in the picture. Also it can only draw lines of certain slopes and circles of certain radii. epic is an extension of picture and eepic an extension of epic. The picture environment is documented in [Lam94, pages 118–129].</p>
<p>P<small>I</small>C<small>T</small>E<small>X</small> X<small>Y</small>-pic⁶</p>	<p>Both of these packages are extensive extensions of the L<small>A</small>T<small>EX</small> 2_ε picture environment. They are designed to allow graphs and diagrams to be drawn inside T<small>EX</small> and L<small>A</small>T<small>EX</small> 2_ε. They are both entered as packages in L<small>A</small>T<small>EX</small> 2_ε with the commands <code>\usepackage{pictex}</code> and <code>\usepackage[options]{xy}</code> respectively. The manual for P<small>I</small>C<small>T</small>E<small>X</small> is <i>not</i> public domain, even though the macro files are. The system administrator should have a copy of the manual, which is excellent due to the plethora of useful examples. The documentation for X<small>Y</small>-pic is in the directory doc in the files xyguide.dvi and xyrefer.dvi. The X<small>Y</small>-pic package has been kept far more up to date by its authors than the P<small>I</small>C<small>T</small>E<small>X</small> package.</p>
<p>T<small>EX</small>draw P<small>STRICKS</small>⁷</p>	<p>Both of these are macro packages that allow POSTSCRIPT drawings to be produced in T<small>EX</small>. Both of these packages basically allow a large number of both low and high-level POSTSCRIPT commands to be used from within a T<small>EX</small> or L<small>A</small>T<small>EX</small> 2_ε document. They are both entered as packages in L<small>A</small>T<small>EX</small> 2_ε with the commands <code>\usepackage{texdraw}</code> and <code>\usepackage{pstricks}</code> respectively. The documentation for both packages is in the directory doc. For T<small>EX</small>draw the manual is the file texdraw.ps, and for P<small>STRICKS</small> the documentation is in the files pst-usr1.ps, pst-usr2.ps, pst-usr3.ps, and pst-usr4.ps. Note that neither of these packages has been updated for L<small>A</small>T<small>EX</small> 2_ε, although both should work with it.</p>

Table 3: Packages for producing drawings in TEX.

commands must all be redefined in terms of the POSTSCRIPT fonts. If you want to see how this can be done, see the file **mathtime.sty**, and [Knu86, pages 154–156, 357–360, and 428–433]. The **psnfss** package is documented in the file **psnfss2e.dvi** and in [GMS94, pages 332–340]. One nice feature of the **psnfss** package is the **pifont.sty** package. This allows the characters from the Zapf Dingbats font to be used as labels in a list or elements in a line. Note that when using POSTSCRIPT fonts in a document, it is best to preview it with **ghostview** rather than **xdvi**. For some unknown reason **ghostview** displays ragged right margins, even when the margins are justified as in this document. In spite of the on-screen appearance, the document margins are justified when printed.

If all of these hacks to get POSTSCRIPT into TEX are making you nauseous⁸, there is one other alternative. There is a document layout program called **LOUT**, which is similar to TEX in that it resembles a programming language, whose compiled output is native POSTSCRIPT. Furthermore, in the same way that TEX resembles a procedural programming language, **LOUT** resembles an object oriented programming language. Essentially **LOUT** was written as the result of 10 years of using TEX and seeing some of its pitfalls in hindsight. **LOUT** was written by Jeffrey Kingston of the Basser Department of Computer Science at the University of Sydney. It is available from ftp.cs.su.oz.au in the directory **/jeff/lout** as **lout.3.06.tar.Z**.

⁸The author definitely falls into this category.

In closing, here are a few TEX tricks that are not in the TEX-FAQ. The 3 characters `\`, `^`, and `~` are used in LATEX 2_ε as parts of commands, and can not normally be printed. One way to print them is to use the commands `\setminus`, `\symbol{"5E}`, and `\symbol{"7E}` respectively. By default, \mathcal{AMS} -LATEX does *not* break pages in the middle of equations. Putting the command `\allowdisplaybreaks` after `\documentclass{}` and before `\begin{document}` allows page breaks to occur in equations. If your document is a few lines over some page limit, insert the command `\enlargethispage*{length}` somewhere in the text of the what you want to be the final page. As a rule of thumb, set *length* to `2ex` for each line that does not fit on the page. The font used on the title page is selected using the command `\fontfamily{cminch}\selectfont`. If the chosen font size is 12 points or larger, then the letters are one inch tall, otherwise they are about three-fourths of an inch tall. Also, if you like the tables in this document, read the examples in [Lam94, pages 204–207] carefully. The tables were formatted using the commands shown in the following example.

Table Example: LATEX 2_ε code

```
\usepackage{array}
\usepackage{hhline}
\usepackage[sl,bf,hang,small]{caption}

:

\begin{table}[htbp]
\setlength{\topsep}{0in}
\begin{center}
\begin{tabular}{|>\centering m{1in}||m{5.5in}||}
\hhline{--}
\multicolumn{2}{|c|}{\rule[-0.075in]{0in}{0.25in}\Large\bf Incorporating
  External Drawings into \TeX}
\\ \hhline{||=}
\multicolumn{2}{|c|}{\large\textit{Auxiliary Drawing Programs for \TeX}}
\\ \hhline{--}
\textsl{Program} & \multicolumn{1}{c|}{\textsl{Description of the program}}
\\ \hhline{-||-}

:

{\texttt{epsfig.sty} \ \ graphicx.sty \ \ graphics.sty}} &
All of these packages are designed to incorporate \textsc{PostScript}
files into a \TeX\ \texttt{.dvi} output file. They are entered as packages
with the command \verb+\usepackage{+}\textit{package name}\verb+}+. To enter a
centered figure into a document with \texttt{epsfig.sty}, use the command
\verb+\centerline{\psfig{\figure=+}\textit{file
  name}\verb+,+}\textit{options}\verb+}}+ at the desired location
in the document. For \texttt{graphicx.sty} and \texttt{graphics.sty} the
required command is
\verb+\centerline{\includegraphics[+]\textit{options}\verb+}[+]\textit{file
  name}\verb+}}+. The documentation for all three of these packages is in
the file \verb+grfguide.dvi+.
\\ \hhline{-||-}
```

```

:
\\ \hline{--}
\end{tabular}
\end{center}
\caption{Programs and packages for incorporating external drawings into \TeX.}
\label{tbl:ext_draw_tex}
\end{table}

```

Note that the column specifier `m{length}` vertically centers the two adjacent columns. The prefix command `>\centering` horizontally centers the entries in the first column. Also note the use of the `\multicolumn` environment spanning only one column in order to center the header for column two. The vertical bar only needs to appear on the right of this specifier because the original declaration following `\begin{tabular}` inserts it on the left side. The enumerated list was formatted using the commands shown in the following example.

List Example: L^AT_EX 2_ε code

```

\usepackage{enumerate}

\newlength{\enumerateindent}
\setlength{\enumerateindent}{\leftmargini}
\addtolength{\enumerateindent}{-\labelsep}
\addtolength{\enumerateindent}{-0.4\labelwidth}

:

\begin{enumerate}[\hspace{\enumerateindent}1) ]
\item Seven new alignment formats for equations. For example, these alignment
      formats allow the following two constructions which would be rather
      difficult in \LaTeXe. The first construction is
:
\end{enumerate}

```

Notice that in order to indent the elements of the list, a space *must* be explicitly inserted. One last example which may be useful is an example of putting a double box around a title page using `fancybox.sty`. Note that this set of commands is *not* taken from this document.

Box Example: L^AT_EX 2_ε code

```

\usepackage{fancybox}

:

\begin{document}

\thisfancyput*(-0.1in,-9.1in){%
  \setlength{\unitlength}{1in}%

```

```

\linethickness{2\fbboxrule}%
\framebox(6.7,9.2){%
  \linethickness{\fbboxrule}%
  \framebox(6.6,9.1)\protect\rule{0in}{0pt}%
}%
}
\title{{\fontfamily{cminch}\selectfont \TeX\TiPs\}[0.5ex]} Supplementary
information about \LaTeXe}
\author{James W. Howse}
\maketitle

```

The command `\protect\rule{0in}{0pt}` is an “invisible” character which insures that the two boxes are both vertically and horizontally centered with respect to each other. Note that the position of the box is determined by the location of the lower-left hand corner. If you are curious about the commands used to achieve any other formatting in this document, see the file `tex_tips.tex`.

Contacting the Author

This document is maintained by Neall Doren. He can be reached via email at `nedoren@sandia.gov`. Please report any anomalies or suggestions for changes to Dr. Doren, via email only. The document you are reading is available via the OGS website. As an alternative, it can be downloaded from the mirror website <http://www.math.unm.edu/~nedoren/latex> in the subdirectory `user_manuals` under filename `textips.ps` (POSTSCRIPT) or `textips.pdf` (ADOBE[®] PDF).

References

- [GMS94] M. Goossens, F. Mittelbach, and A. Samarin. *The LATEX companion*. Addison-Wesley Publishing Co., Inc., Reading, MA, 1994.
- [Knu86] D.E. Knuth. *The TEXbook*, volume A of *Computers & Typesetting*. Addison-Wesley Publishing Co., Inc., Reading, MA, 1986.
- [Lam94] L. Lamport. *LATEX: A document preparation system: User's guide and reference manual*. Addison-Wesley Publishing Co., Inc., Reading, MA, 2nd edition, 1994.