AmigaTEX

An Implementation of TEX
for the Amiga

Version 1.2
12 May 1987

Distributed by:
n^2 Computer Consultants
P. O. Box 2736
College Station, TX 77841
(409) 696-1078

Software Licensing

AmigaTEX is not a public domain software package. Use of AmigaTEX without a valid license is prohibited. Nothing included with AmigaTEX is copy-protected in any form both for the convenience of the user and because of the author's personal feelings on copy protection. Copies may be made of the components of AmigaTEX (which includes the programs `tex`, `initex`, `bibtex`, `preview`, `gftopk`, `pxtopk`, `pktype`, and `set`) only as required for backup and convenient use on a single Amiga personal computer; any further distribution is strictly prohibited.

Manual Copyright

 The TEX source for this manual is stored on the distribution diskettes as one of the examples; this does not imply the right to print it out or transmit it in any form.

Disclaimer

Radical Eye Software makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Radical Eye Software reserves the right to revise this document and the enclosed software without obligation of Radical Eye Software to notify any person of such revision.

Trademarks

TEX is a trademark of The American Mathematical Society. Amiga is a trademark of Commodore-Amiga, Inc. Unix is a trademark of Bell Laboratories. IBM is a trademark of International Business Machines. HP and LaserJet are trademarks of Hewlett-Packard Incorporated. PostScript is a trademark of Adobe, Incorporated. Dr. Pepper is a trademark of the Dr. Pepper Company.

Table of Contents

AmigaTEX
Users Manual and Installation Guide

Welcome to the world of TEX on the Amiga. If you have used TEX on other computer systems, you will feel at home with AmigaTEX. If you are a newcomer to TEX, welcome! You will find the Amiga to be a powerful machine on which to learn TEX and develop your documents.

To run AmigaTEX, you need at least 512K of RAM and at least two floppy drives, one floppy drive and 1.5M of RAM, or one floppy and a hard disk, and version 1.1 or later of the Amiga operating system. Before we jump in, a few comments must be made about conventions used in this manual.

All commands and file names will be set in typewriter type. Commands are separated from the accompanying text by blank lines, and are preceded by any prompts that are not typed by the user. For instance, to list the size of the file `foo.bar`, you would type

```
1> list foo.bar
```

In this example, '`1>` ' is the CLI prompt (yours may be different), and you would not type those letters. The word `tex` refers to the program TEX as it has been ported to the Amiga; the word TEX refers to the more general software package and typesetting language that runs on a wide variety of machines. Any unfamiliar terms may be looked up in the glossary at the end of this manual.

1. First Run

Before you run any of the supplied software or even examine the contents of the disks, you should back them all up. There are two ways to back up floppies on the Amiga: using the CLI and using the workbench.

1.1 Copying floppies using the CLI

If you are familiar with the CLI, the easiest way to back up floppies is to type

```
1> diskcopy df0: to df1:
```

The system will prompt you to insert the floppies. Insert the original floppy into the drive on the main system; insert a blank floppy into the external drive, and hit carriage return. The system should copy the disk and give you another prompt when it is done. If the system responds instead with:

```
1> diskcopy df0: to df1:
Unknown command diskcopy
1>
```

First, insure that you typed the command name correctly. If you did, then your system does not have a CLI diskcopy routine; so you will have to use the following procedure for the workbench. Even if it did work, if you are unfamiliar with the Amiga, you might read the following description to understand how to interact using the mouse, icons, and menus.

1.2 Copying floppies using the workbench

Copying floppies using the workbench is easy, despite the length of the following description. Insert the original floppy into the internal drive, and a blank disk into the external drive. After a second or two, an icon (small symbol above a name used to represent a file or device on the amiga) should appear on the screen for both disks. The blank disk might have an icon with the name `DF1:BAD`. Don't let that worry you. The floppy is simply unformatted, and the diskcopy operation will format it as it copies. Now, using the mouse, move the arrow until its tip is over the icon representing the original disk. Hold down the left mouse button (the select button) and 'drag' that icon until it rests on top of the icon for the blank disk. Now release the select button.

The machine might now prompt you to replace the workbench floppy into any drive. The prompt is called a 'requester' because it requests something from the user before the operation can proceed. In this case, it needs to load the diskcopy program into memory, and that program resides on the workbench disk. Remove the floppy from the internal drive and re-insert the workbench floppy. After a few seconds, the system will prompt you to replace the disk you are copying from into the internal drive. Remove the workbench floppy and re-insert the disk it replaced.

NEVER REMOVE A DISKETTE WHILE THE DRIVE LIGHT IS ON. This can destroy any data on the floppy. Even when the system puts up a requester, wait until the drive light goes out before removing the floppy. This is important; you might consider re-reading this paragraph.

If everything went as it should, you should now have another requester on the screen titled `Disk Copy`. Move the mouse pointer over the *Continue* option, and hit the select button. The copy operation should start and finish a few minutes later. You have just backed up a floppy. Remember this operation; you should use it often to guard against bad floppies, electrical storms, and three-year-old popsicle eaters.

We are not done yet. If you look at the icons on the workbench, you will notice that the new floppy is called `copy of foo.bar` if the original disk was called `foo.bar`. If you are making copies of the distribution floppies from which to work, this name will not do, since AmigaTEX will often request a specific floppy by name. Therefore, you will need to rename the floppy. To do this, move the mouse pointer over the icon for the new copy, and press and release the select button. The icon should turn from white to black. Now hold down the right mouse button (the menu button). The title bar on the workbench should change to show `Workbench`, `Disk`, and `Special`. Move the pointer (still holding down the menu button) over the word `Workbench`. A menu should appear.

Now, still holding the menu button down, drag the mouse pointer down until it is over the word `Rename`, and release the menu button. This is how menu selections are made on the Amiga. A thin box should appear in the middle of the screen with the name of the

selected icon in it. (If you are using version 1.1 of the operating system, you will need to move the mouse pointer until it lies within this box, and hit the select button. Anything you type before this will be ignored.) Now, simply hit the *DEL* key on the keyboard (in the upper right) eight times, until `copy of` disappears and the cursor is on top of the first character of the original name. Now hit the *RETURN* key, and the name of the icon should change. Appropriatel label the new copy you just made, with a paper floppy label, put the original disk away in a safe place, and repeat for all of the distribution floppies.

1.3 Setting up Preferences for AmigaTEX

AmigaTEX must be run in the 80-column mode and using CLI. If this is not how your system is configured, or you are not sure, you should go into preferences.

To go into preferences from CLI, type

```
1> df0:preferences
```

From the workbench, move the pointer over the `workbench` icon, and double click (hit the select button twice rapidly). Next, double-click over the `Preferences` icon that should appear inside the workbench window. Your screen should change into the `Preferences` screen.

Look in the middle of the left side of the screen; you should see the words `Text` and `CLI`, each with two little boxes to the left of them. The selected box should be `80` for `Text`, and `On` for `CLI`. If they are not set correctly, simply select the appropriate box with the mouse and select button. If they needed changing, you now need to select the `Save` option in the lower right corner of the screen and reboot; if the options were correct, you can select the `Cancel` option. To reboot, simply hold down the *CTRL* key (middle left of the keyboard) and the two Amiga keys (fancy 'A' keys on either side of the space bar) at the same time.

1.4 Getting into a CLI

Now you need to get a CLI window to work with. If you already have opened a CLI, fine; you are in good shape. Otherwise, the following procedure will open a CLI window for you.

Double click on the `workbench` icon if the `workbench` window is not already open. Within the `workbench` window, you should see an icon labeled `CLI`. If you don't see this icon, double click on the `system` icon to open its window; the `CLI` icon should be in this window. If it is not, check preferences again to insure that the `CLI` has been turned on. If all else fails, reboot, and when the screen first turns blue, hold down the *CNTRL* key and hit the 'd' key until the word *BREAK* appears. Now you are in a CLI.

If the CLI you are in does not extend to the full width of the screen, you will need to resize it. This is done by first selecting the title bar of the CLI window (where the word `CLI` or `AmigaDOS` is) and dragging the whole window until it is flush with the left of the screen. Next, select the sizing gadget in the lower right of the window, and drag it until it is flush with the right of the screen.

1.5 Setting up a 512K system

If your Amiga has 512K of RAM, you will need to make some changes to the supplied disk to start. Before you read any further, make sure that you have backed up the distribution diskettes, and are working from your own copies. The following step, if run on the distribution floppies, will destroy a file that you will need if you ever get more memory. With that caveat, please continue.

To set up AmigaTEX for a 512K system, insert the floppy labeled `TeXfiles` into the external drive. From the CLI, type

```
1> execute texfiles:s/to-512
```

(Note that the Amiga is case insensitive; upper and lower case in file names are equivalent. Thus, though the floppy is really named `TeXfiles`, it can be referred to as `texfiles`.) After a few seconds, it should prompt you to insert the disk labeled `TeX`, and then to reinsert the workbench disk. After the operation is complete and the floppy drive lights have gone out, reboot the machine.

1.6 Running TEX

Now you are ready to run TEX. If you are ambitious and familiar with TEX, you may choose to prepare a short sample file at this time. For those not familiar with TEX and for those in a hurry, we have prepared a few sample files. At this point, you should be at a CLI prompt. Insert the disk labeled `TeXfiles` into your external drive, and type

```
1> texfiles:c/tex
```

The disk will grind for a while, and then a requester for the floppy labeled `TeX` will appear. Remove the `TeXfiles` floppy and insert the `TeX` floppy in its place. After a second or two, TEX should appear with the following hello:

```
This is CTeX, Amiga Version 2.1 (no format preloaded)
**
```

If instead you get the message

```
Out of memory during allocation!
```

then there are several possibilities. If you are running on a system with more than 512K of RAM, you must have too many things in memory at the moment. Shut down some other programs and try again. Or possibly you have not told the system about your extra memory; the Amiga 1.1 operating system does not automatically recognize expansion memory, and some memory boards are not recognized under 1.2. Consult the documentation supplied with your memory boards. If you are running on a 512K system, perhaps you didn't follow the above instructions that pertain to you. Alternatively, you might have some files in RAM disk. Try rebooting.

The double asterisk (**, also referred to as double splat) prompt is TEX's way of asking you for a file name. Type your own file name if you have prepared a file, or type

```
**tex:samples/story
```

if not. There will be a pause of approximately ten seconds, after which the screen should look like this:

```
This is CTeX, Amiga Version 2.1 (no format preloaded)
**tex:samples/story
(tex:samples/story.tex [1])
*
```

The single asterisk prompt indicates that it is finished, and is waiting for more commands. At this point, simply type

```
*\bye
```

A few more seconds should go by, and then TEX should reappear with another double splat prompt. This is because of the 'loop mode' that will be explained later. For now, just hold down the *CNTRL* key and press the '\' to get the CLI prompt back. Congratulations! You have just finished your first TEX job on the Amiga!

1.7 Running preview

Now you are probably anxious to see what the output looks like. To see, insert the `TeXfiles` disk into the external drive, and type

```
texfiles:c/preview story
```

Wait for the external drive light to go out, and insert the `TeX` disk. Your file should appear on the screen. If you are ambitious, you might try figuring out how to work the preview program at this point, using what you know about gadgets and menus. Otherwise, you can exit the program by selecting the square gadget in the upper left corner of the screen with the select button, or simply by hitting the 'q' key on the keyboard. If all else fails, reboot and read further.

Do not be discouraged by the apparent awkwardness of this process. Setting up an interactive, multitasking TEX environment is described later. You might care to take a breather here, so you can enjoy the descriptions of TEX and the Amiga which follow.

2. What is TEX?

TEX is a typesetting language and system developed by Professor Donald Knuth of Stanford University over the period 1977 to 1986. It was designed to be used in typesetting books and manuscripts, especially those containing much mathematics. The current version of the language has been stable since 1982, and is currently in use at thousands of computer installations all over the world.

The input to TEX is somewhat free format; it is emphatically not a WYSIWYG (what you see is what you get, often rephrased as what you see is all you get) system. The source can be created with any typical text editor, and for the most part spacing and line breaks in the input are ignored. Interspersed with the text are various typesetting commands that might, for instance, change fonts or skip to the next page. TEX is programmable with paramaterized macros and user variables of a number of types, allowing the creation of macro packages that extend the power of TEX or make it more accessible.

TEX is not the solution to all the world's ills. As a document preparation system, it has three major deficiencies. First, it does not directly support graphics. You can use horizontal and vertical rules, and do some limited graphics using a variety of methods: LaTEX supports some graphics operations, and most PostScript printer drivers allow inclusion of graphics commands. Secondly, TEX does not easily do complicated page layout, as might be required for newspapers, for instance. And finally, TEX does not allow interactive editing and viewing of the document as it is being developed.

All of these are design decisions; TEX was never intended to address any of these concerns. What TEX does, however, TEX does well. It supports a huge library of fonts, and additional fonts can be purchased from a number of companies. It supports full kerning and ligatures in its fonts. It has no equal when typesetting complicated mathematical formulas and displays. Its hyphenation algorithm is among the best in typesetting systems, and its line- and page-breaking algorithms are nothing short of incredible.

TEX is a complicated system, because typesetting is not a simple task. It is easy to generate quality documents without knowing much about TEX, but as your experience and requirements grow, you will find that TEX has the power, programmability and expandability to handle your most demanding needs.

One of the primary objectives of Knuth in writing TEX was computer and printer independence; the output you display on your Amiga screen should appear (within the resolution limits of the device) exactly the same as the output on the professional typesetting equipment of Addison-Wesley, for instance. The same source should work and generate identical output on all computer installations that run TEX.

To illustrate TEX, the processing of our sample `story.tex` above is going to be examined, and each file used by TEX in the creation of the document will be explained. The source was typed in by hand from the first example in *The TEXbook,* page 24, and as all source files for TEX should, it has the extension `.tex`.

Some of the commands used in `story.tex` are not a part of primitive TEX; primitive TEX is very primitive indeed. Therefore, TEX by default loads a set of macros called 'plain

TEX' into memory. Rather than load these macros in as TEX source (which it could), it instead loads them in a pre-digested form called a 'format file', with the extension `.fmt`. This `plain.fmt` file contains essentially a memory image of TEX after all of the macros in plain TEXare loaded.

Primitive TEX does not know any characteristics of any fonts, either. Therefore, a number of font metric files with the extension `.tfm` were loaded while the `plain.fmt` file was created. These files contain information about individual characters in the font and the font as a whole, such as the height, width, and depth of each character, what ligatures the font contains, and what kerning is necessary between which characters. There is one of these files for each font accessible to TEX, and they contain only information about the fonts that is independent of the output device.

As TEX was processing `story.tex`, it created two files, one with the extension `.log` and one with the extension `.dvi`. The log file contains essentially a listing of the run as it was seen from the terminal, including any error messages that were displayed. This file is useful if an error message was too long and scrolled off the display, for instance; various switches can also be turned on inside TEX to display more information about what is happening during the typesetting process. The main output file of TEX is the device-independent file, with the extension `.dvi`. This file contains a description of the final document, including the location and font of each character used. The units used in this file are not based on any particular device, and the actual characters themselves are not described. It is up to another program, the 'driver', to interpret this `.dvi` file, look up the appropriate raster representations of the individual characters, and create the final document.

The preview program is one such program. It 'drives' the Amiga screen, loading font raster information from a collection of packed, or `.pk`, files. There is one of these files for each font at each size, since the raster description of a character is by its very nature dependent on the resolution of the output device.

This was intended to be a simple description of TEX; for more information, please refer to *The TEXbook* by Donald Knuth and *LaTEX User's Guide and Reference Manual* by Leslie Lamport, both published by Addison-Wesley Publishing Company and available in most college bookstores.

3. Introduction to the Amiga

The Amiga is an impressive computer system that is well suited to TEX for a number of reasons. The high-resolution graphics built into the machine allow high-quality preview of the output. Multitasking and a large memory capacity allow the editor, previewer, TEX, and even other programs to be concurrently resident and active, yielding a much more interactive environment than is possible on any other microcomputer. Intuition, the Amiga windowing software, ties these and other packages together into a unified, convenient, natural interface for the user.

The section title is somewhat of a misnomer, since if you have run the sample, you are probably familiar with the Amiga by this time. Nonetheless, there are a few things that have not been covered yet.

The mouse is the primary input device for the Amiga. The point and click approach is easy to master. The Amiga mouse has two buttons, the select button on the left and the menu button on the right. The select button is used for selecting data, operations, windows; the menu button is used to select among a number of alternative actions or settings. Gadgets are graphic images that are associated with a particular action; moving the mouse pointer over them and clicking the select button starts the operation. Note that for most gadgets, the mouse must remain over the gadget while the button is both pushed and released; it is the release on the gadget that actually signals the operation to begin.

Menu items are selected by pressing and holding the right mouse button as the pointer is moved first to the title bar to select the particular menu, and then down the menu (and possible sub-menus) until the appropriate action is found. It is the release of the menu button that starts the operation.

Windows and screens help separate the input and output of programs from one another, and to bring to display the information currently being worked with and hide irrelevant data. Screens occupy the full width of the video display and can only be dragged up and down. A typical screen is the workbench screen. Windows reside in screens, and can be an arbitrary rectangular portion of a screen. A window cannot be moved from screen to screen; once created, it is permanently associated with a particular screen. A window can be moved horizontally or vertically so long as it remains on the screen.

A window inherits the video characteristics of a screen, but each screen can have its own characteristics. For instance, the workbench screen is a four-color high-resolution non-interlace screen; the preview screen is a two-color high-resolution interlace or non-interlace screen. Many graphics programs run in a 32-color low-resolution non-interlace screen. At any given time, either zero or one window is active. A screen is only 'active' if a window on that screen is active. Inactive windows have their title bar cross-hatched over to indicate their status; the active window does not have this cross-hatch pattern. To select a window, you simply place the pointer inside that window and hit the select button.

To drag a window or a screen, select the title bar and move the mouse with the select button held down. A screen will move as the mouse moves; for a window, an orange outline of the window will move, and when the mouse stops moving, the window will be redrawn in its new position. The two gadgets in the upper right hand corner of a window or screen

are the to-front and to-back gadgets; selecting the right-most gadget (to-front) brings the corresponding window or screen in front of all other window or screens, selecting its neighbor (to-back) pushes it in back of all others. Note that a window might cover the screen's to-front and to-back gadgets; full-sized CLI windows are typical culprits. You probably want to resize a full-sized CLI window to make it slightly shorter, and then drag it down a bit to expose the screen gadgets; otherwise, you cannot move among the various screens easily.

Windows usually have a sizing gadget in the lower right hand corner. This is used to change the size of a window; simply select and drag it. They also usually have a close gadget in the upper left hand corner; selecting this gadget terminates the program running in the window. Menus are associated with the active window, but are displayed on the screen title bar. Thus, one screen might have many possible menus, depending on which window is the active one.

An icon is a graphics image that represents a file or device. On the workbench screen, for instance, there is an icon for each disk. Files may or may not have an icon associated with them; currently, the AmigaTEX system does not use icons. Icons are selected with the select button; they are activated (if a program, it is run; if a directory, it is opened) by double-clicking on them.

Another nice feature of the Amiga is that it doesn't restrict you to use the desktop metaphor (icons, menus and windows); you are free to use the more conventional CLI, or command language interface, under which you type commands with the keyboard. This is the primary environment AmigaTEX runs in. The Amiga CLI is a simple shell to the underlying TRIPOS-based operating system. As with any command-based interface, there are commands and conventions that must be learned. For instance, `cd` changes the current working directory; `dir` displays the names of all of the files in that directory, and `type` displays the contents of a file. For a description of these commands, refer to the *AmigaDOS User's Manual* published by Addison-Wesley, available where you purchased your Amiga.

If your Amiga makes annoying clicking noises every second or so, it is because your disk drives do not have floppies in them; insert blank floppies into the disk drives when not in use to keep the Amiga happy.

4. Installation

Now you are ready to install AmigaTEX onto your system. The next few sections each pertain to a certain part of the installation; read only those sections that pertain to you.

4.1 Installation on a 512K system

AmigaTEX will run on a 512K system, but you will not be able to take full advantage of the package without additional memory. Normally TEX alone requires 642K of RAM. In order to run TEX on a 512K machine, several restrictions must be observed. The RAM disk cannot be used or even created. Note that any reference to the RAM disk will create it if it does not exist. To see if it exists, simply type

```
1> info
```

If it exists, it will be listed as a mounted disk, and you must reboot before you can run AmigaTEX. No other applications can be run at the same time as AmigaTEX; it needs all of the memory in the machine. LaTEX and other extensive macro packages cannot be run. Even with these restrictions, it is quite possible that you will eventually create a document that AmigaTEX cannot process because it runs out of memory internally. However, most typical documents should present little difficulty; AmigaTEX itself and this manual were both fully developed and run on a 512K Amiga.

The version of AmigaTEX that runs on a 512K system is identical to the version that runs on a larger system, but the format files are different because of the different array sizes. In addition, a special file must be installed that informs TEX of the sizes to make its internal arrays. Make sure that you have followed the procedure outlined in 1.5 above to move the files to their correct places.

If you later get more memory and wish to use the full power of AmigaTEX, you will need to re-install the original format file. To do this, simply make a fresh backup copy of the first distribution diskette and use it instead of the `TeX` disk you used with 512K.

4.2 Installation on a dual-floppy system

AmigaTEX runs comfortably on two floppies. The floppy that resides in the internal drive is the workbench floppy and contains all of the files for the Amiga operating system and all of your personal TEX files. The floppy in the external drive contains all of the files TEX requires to run, including the font metric, format, and packed raster files. This second floppy must be labeled '`TeX`' and is usually just a copy of floppy one of the distribution.

Usually the workbench floppies as supplied with the Amiga are very full, restricting the amount of working file space for your TEX files. There are a lot of files on the workbench that are not used much, though, and can be deleted to make more space. This is a list of some of the files and directories on the distribution floppy that can be deleted, in order from largest to smallest. The colon on the front of each name indicates the root directory of the floppy. To delete an entire directory and its contents, type

```
1> delete :fonts all
```

to delete the fonts directory and all of the files in it. To delete only a file or two, type

```
1> delete :Preferences :Preferences.info
```

Note that you can specify a maximum of ten files to the delete command. To list the contents of a directory, type

```
1> list :Utilities
```

`:Utilities` This directory contains two programs, a calculator, and a notepad, totalling some 72K. If you never use either program, delete the entire directory; otherwise, just delete whichever program you don't use. The associated `.info` files can go as well.

`:devs/printers` In this directory is a driver for all of the printers the Amiga supports, totalling 54K. After selecting your printer type with preferences, delete all of the drivers you don't use. For instance, if you are using an HP LaserJet, delete all of the files in this directory except `hp_laserjet`. If you don't have a printer, you might want to delete the entire directory.

`:Preferences` This program is used to configure the system to your tastes and hardware, but is 52K huge. After becoming comfortable with the Amiga and tweaking all of the knobs this program has, you might want to delete it. If you choose to run preferences at a later date, you can simply insert another workbench floppy with preferences intact into the external drive, and type

```
1> df1:preferences
```

`:libs/translator.library, :devs/narrator.device, :c/say` (and `:system/say` under 1.2) Three files that total 43K give your Amiga the amazing capacity of speech. Luckily, the Amiga is easier to shut up than most people are.

`:fonts` This directory contains several different fonts that the Amiga can use. Since AmigaTEX uses its own set of fonts, and the default system fonts are built into the hardware, you can feel free to delete all 40K in this directory.

`:c/edit, :c/ed` Most people need only one editor, and usually choose to use a commercial editor rather than the supplied Amiga editors. Either one or both of these files can be deleted, saving you about 19K apiece. You might wander around in the `:c` directory looking for other, unfamiliar commands to delete as well. A minimum subset might be `run`, `dir`, `list`, `delete`, `copy`, `execute`, `diskcopy`, `rename`, `cd`, `date`, `makedir`, `echo`, `endcli`, `newcli`, `type`, and `stack`. In addition, under 1.2, `:system` contains several cute programs you might not need.

`:system/iconed` This is a great program to play with; it allows you to redefine the imagery of an icon. On a typesetting system, however, it has little use. This gem uses 34K.

`:demos` This directory has a few cute demonstrations of the Amiga's graphics abilities. Once you are finished impressing your friends, wipe all 20K.

`:Clock` This is another neat program that you may never use; deleting it reclaims 17K.

There are other things you can delete on the workbench, but these are the major ones. The above instructions can free up approximately 370K of disk space to be used for AmigaTEX. What do you do with all this space? Well, the first thing you should probably

do is copy the `set` and `preview` programs to the workbench by inserting the `TeXfiles` floppy into the external drive and typing

```
1> copy texfiles:c/set c:
1> copy texfiles:c/preview c:
```

You might also want to copy the actual `tex` program itself with the command

```
1> copy texfiles:c/tex c:
```

The `tex` program is approximately 127K, though, so it will put a large dent in the available disk space. The advantage of placing `tex` on the workbench disk is that you can invoke `tex` simply by typing

```
1> tex
```

The alternative, if you don't copy `tex` to the workbench disk, is to insert the `TeXfiles` floppy into the external drive, and type

```
1> texfiles:c/tex
```

On a 512K system, since TEX cannot run with other programs, each time you invoke TEX, you will need to repeat this command and swap the `TeXfiles` and `TeX` floppies. On a system with more memory, this needs to be done only once, since AmigaTEX will stay resident using the loop mode described below.

4.3 Installation on a hard disk system

On a hard disk system, AmigaTEX really shines. To install it, first insert the `TeXfiles` directory into the external drive, and type

```
1> copy texfiles:c/tex c:
1> copy texfiles:c/preview c:
1> copy texfiles:c/set c:
```

Next, create a directory on the hard disk named `TeX`, by

```
1> makedir dfh:TeX
```

(replacing `dfh:` with the actual device name of your hard disk.) Now insert the `TeX` floppy into the external drive, and type

```
1> copy tex: dfh:tex all
```

This will copy the entire contents of the floppy onto the hard disk, and it will take several minutes. You might go get a Dr. Pepper while it is running, and relax. Before running AmigaTEX on the hard disk, you should type

```
1> assign tex: dfh:tex
```

to tell AmigaTEX where to find the files. When you type this command, the `TeX` floppy should not be in either drive, because the system might get confused as to what `tex:` refers to. If you are familiar with the `s/startup-sequence` file, that is a good place for this command.

4.4 Running AmigaTEX

Now everything should be in place to run AmigaTEX. A few scenarios for running AmigaTEX from a cold boot will be illustrated, so you might want to reboot at this point.

If you are running on a 512K system with two floppies, you will need to run the editor, previewer, and TEX separately. To run TEX, you will either type

```
1> tex
```

(if you installed `tex` on the workbench disk) or insert the `TeXfiles` floppy into the external drive and type

```
1> texfiles:c/tex
```

In either case, insert the `TeX` disk into the external drive when prompted for it. To exit from TEX, type `\bye` if you have a single asterisk prompt, and then hit *CNTRL-*\ when you get the double asterisk prompt. If TEX is in some strange mode where `\bye` does not work, you can 'emergency-exit' TEX by hitting *CNTRL-*\.

If you are running on a machine with more memory, you can load TEX, the editor, and the previewer all into memory at the same time and switch among them quickly and easily. If you are using a hard disk, you might need to type

```
1> assign tex: dfh:tex
```

before you run `tex` or `preview`. At this point, you are ready to open a window for each program you will run. Type the following command to the CLI:

```
1> execute tex:s/openwindows
```

Three new CLI windows should open on top of the one you are currently running. They should be labeled `TeX`, `preview`, and `edit` respectively. First, select the `edit` window, and invoke your favorite editor from it. For example, you might type

```
1> ed story.tex
```

After the editor comes up, select the `tex` window (you may have to hide the editor screen or window to expose the `tex` window) and type

```
1> tex
```

to invoke TEX. If you are prompted for the `TeX` floppy, insert it. Wait for the TEX prompt. Next, select the `preview` window, and type

```
1> preview
```

to bring up the previewer. Congratulations! You are running all three programs simultaneously! Practice switching back and forth among the windows and screens; you will be doing this a lot. You might move the windows or screens around to suit your tastes.

Note the instructions to wait until the disks stop before typing the next command. This is because under the 1.1 version of the operating system, if two files are being accessed off the same floppy simultaneously, the system slows down to a crawl while the disk head seeks back and forth.

Another caveat is appropriate here. Another way to run multiple programs at the same time is to invoke them with the `run` command, rather than to open a new CLI window for each program. The `run` command automatically lowers the priority of the program it is running, and this can keep it from getting any processor time. Use `run` with care.

A final digression. At this point, the screen might be flickering because the `preview` program is running in interlace mode. To eliminate this shaking, bring the preview screen to front and select (click inside) its screen. Select the `non-interlace` option in the `Modes` menu, and the flicker should go away. Now you can comfortably continue.

5. Using TEX

True to the spirit of computer independence, which is an integral idea in TEX, the AmigaTEX implementation performs exactly as any other port of TEX. If you invoke AmigaTEX with a file name given on the command line, it will process that file and return with a prompt. For example, to process the file `tex:samples/story`, the user dialog would appear as follows:

```
1> tex tex:samples/story
This is CTeX, Amiga Version 2.1 (no format preloaded)
(tex:samples/story.tex [1])
*\bye
Output written on story.dvi (1 page, 672 bytes).
Transcript written on story.log.
1>
```

The user typed only the original command line and `\bye`; the remaining output is from TEX.

5.1 The Amiga Loop Mode

A significant part of the time spent processing `story` was spent loading the `tex` executable into memory. To help eliminate this wasted time, AmigaTEX runs in a loop if it is not given a file name on the Amiga. For instance, if we had wanted to process `story.tex` but leave TEX in memory for the next time we needed it, we could have simply typed

```
1> tex
This is CTeX, Amiga Version 2.1 (no format preloaded)
**tex:samples/story
(tex:samples/story.tex [1])
*\bye
Output written on story.dvi (1 page, 672 bytes).
Transcript written on story.log.
This is CTeX, Amiga Version 2.1 (no format preloaded)
**
```

Note that this time, the banner message and double splat prompt returned, waiting for the name of the next file we want TEX to process. It will continue to wait patiently as you preview the output from the previous run and make any changes to the editor source, after which you can simply retype the name of the file to run TEX again.

If you want to run TEX in the loop mode and give it the name of the first file to process, you can do that with the `-r` option, as follows:

```
1> tex -r texfiles:samples/story
This is CTeX, Amiga Version 2.1 (no format preloaded)
(tex:samples/story.tex [1])
*\bye
Output written on story.dvi (1 page, 672 bytes).
Transcript written on story.log.
This is CTeX, Amiga Version 2.1 (no format preloaded)
**
```

5.2 Writing and Reading the Same File

The editor creates the source file, with the extension `.tex`. This file is read by TEX, which writes a `.dvi` file, which is then read by the preview program and displayed on the screen. There are restrictions on reading and writing a file at the same time, however; chaos can easily result. Therefore, it is not wise to ask the editor to update or write out the source file while TEX is processing it. The same restrictions apply to the previewer; you must close the file with the 'close file' option of the project menu before running TEX again.

If you violate this restriction, the editor might complain that it cannot write your file, or TEX might complain that it cannot write the `.dvi` file. If the editor complains, finish running TEX, and then reexecute the save-file command from the editor. If TEX complains, close the file from the previewer and type the `.dvi` file name again.

5.3 The `nul` Source File

When TEX gives you the double splat prompt, it is requesting a file name to process. If it can't find the file you specify, it says so, and asks for another name.

```
1> tex
This is CTeX, Amiga Version 2.1 (no format preloaded)
**lksaj
!  I can't find file 'lksaj.tex'.
<*> lksaj

Please type another input file name:
```

Sometime you may decide that you don't want to process a file at that point, but it keeps giving you this annoying error message. There are two ways to get out of the loop; hit the end of file character (*CNTRL-*), which gives you a fatal error and TEX exits, or simply give it the name `nul`, which exists in the `tex:inputs` directory. TEX will then respond with a single asterisk prompt, at which point you can type `\bye` or `\end` to exit.

5.4 Amiga Environment Variables

There are several files TEX must look for when processing a document; these include format files, font metric files, and auxillary input files. Since people might choose to move these files from their default places, or rename the volume on which they are stored, TEX reads several environment variables and uses them to search for these files.

An environment variable consists of two strings of characters: a name, and some contents. Included with the AmigaTEX package is a program `set` that can be used to examine and change these environment variables. Invoke this program now by typing:

```
1> set
```

The program should simply return with no output; this is because no environment variables are set. If you invoke the program with no arguments, it displays all of the current environment variables. Now, type

```
1> set foo=bar
1> set
foo=bar
1>
```

The first command set the environment variable `foo` to `bar`, and the the second simply displayed the current variables. Now, set `foo` to `mung`:

```
1> set foo=mung
1> set
foo=mung
1>
```

exactly as expected. You can delete an environment variable by not following it with an equals sign or a string:

```
1> set foo
1> set
1>
```

TEX examines the following environment variables. If they are not set, it uses the given default.

```
texmacros=tex:macros
texfonts=tex:fonts
texinputs=.,tex:inputs
texformat=plain.fmt
```

The first three are paths. They are the devices or directories that contain the appropriate files. For instance, when TEX needs to load a format file, it searches the directory `tex:macros`. When it is looking for a source file, it searches the current directory first (the '`.`', borrowed from Unix) and then the directory `tex:inputs`. The last environment variable is the `texformat` variable; this is used to set the default format file that TEX loads

in. If you create your own format file called, for example, `plane.fmt`, you can tell TEX to load it rather than `plain.fmt` by simply setting this variable to `plane.fmt`. This is handy if you use LaTEX rather than plain TEX, since you can simply set this variable to `lplain` and voila! TEX becomes LaTEX.

6. Using Preview

The preview program allows you to display your TEX output on the Amiga monitor in a highly interactive fashion. Its primary purpose is to shorten the edit-TEX-print cycle, to allow you to create your documents more quickly and more conveniently.

As the preview program goes through its operations, its pointer will change. When preview is idle and waiting for a user action, the pointer will appear a small arrow, similar to the workbench pointer. When preview is redrawing a page, the pointer will appear to be a pencil drawing a black line. Finally, when preview needs to load a font, the pointer will become a large black lower-case 'a'. By watching the pointer, you can follow the operations of the preview program.

If a font that the previewer needs is not found, the screen flashes and an error message appears on the screen title bar. In addition, the font metric file for the font will be loaded, and the information contained there used to create rectangular 'characters' of the size and shape of the actual font characters. Thus, though these fonts will certainly not be readable, they will give you some idea of layout and style.

Note that at very low resolutions, like those required for preview, some small characters might degenerate badly, or even disappear entirely. The typewriter font is a particularly good example of this.

Unlike TEX, where the user input is strictly through the keyboard, the user supplies input to the preview program mainly through the gadgets and menus of the screen. A few keys on the keyboard duplicate the function of the gadgets for expert users. The gadgets are active when the cursor is an arrow or a pencil; the menus are only active when the program is idle and the cursor is an arrow.

6.1 Preview Gadgets

At the top of the preview screen are the three default screen gadgets. In the upper right corner are the screen-to-front and screen-to-back gadgets; the rest of the title bar is the screen drag gadget.

Now we progress to the gadgets on the window title bar. The close gadget is a little square with a dot in the middle in the upper left corner. Selecting this gadget will terminate the program, freeing all of its memory. Normally you would use only this gadget if you are through with TEX for the day, or if you have only 512K and want to run TEX. The previewer will peacefully coexist with other well-written Amiga programs. Another way to exit the program is with the 'Q' key on the keyboard.

Next consider the file name gadget. This gadget is the blank area in the left portion of the window title bar. When have selected this gadget, a cursor will appear on the title bar; you can now type the name of a file to preview (less the `.dvi` extension). (If a file name is already in the gadget, you will need to clear it by hitting the DEL key a few times.) Hit carriage return, and the file will be opened. If the file does not exist, the screen will flash and an error message will appear on the screen title bar. Another way to select the file

gadget is to simply hit the 'F' key on the keyboard; the current file name will be cleared and the cursor will be positioned in the gadget ready for your file name.

The previewer buffers information about the current job in memory to make redrawing the page faster. If you make changes to the document and rerun TEX, this information is no longer valid and must be flushed. To do this, simply select the file name gadget and hit carriage return. All information about the file will be flushed, and the file will be reopened as if you had started preview fresh.

The page number gadget is the blank area next to the file name gadget. As you scan around through the document, this gadget will contain the current page number. To go to a random page, simply hit this gadget and type a new page number; if the page exists, it will be displayed. Again, the 'P' key provides a convenient shorthand for selection of this gadget.

The portions of the window title bar that are not gadgets are used as a window drag gadget. You can drag the window around on the screen by selecting and dragging on the title bar.

TEX allows more than one page to have the same number. For instance, if you turn page numbering off, all of the pages will be numbered zero. Subsequent occurrences of the same page will be numbered, for instance, 0.1 for the second occurrence, 0.2 for the third, etc. The first occurance will always be numbered just 0.

The next gadget, which has two small rectangles, one larger than the other, is the zoom/unzoom gadget. At any time the previewer is either in a zoom state or an unzoom state; the zoom state usually displays the document at a higher resolution than the unzoom state. Selecting this gadget will toggle between the zoom and unzoom states. The default unzoom resolution is 91 dots per inch; the default zoom resolution is 120 dots per inch. Note that the units are screen pixels per document inch, not screen pixels per screen inch, which is of course fixed. The 'S' key duplicates the function of this gadget.

The next two gadgets, which appear as arrows pointing to the left and right, are the forward and back page gadgets. Selecting the arrow pointing left displays the page before the current one; selecting the one pointing right displays the next page. The left and right arrow keys on the keyboard also provide this functionality. In addition, the space bar may be used to go to the next page, and the delete and backspace keys go to the previous page.

The long, narrow gadgets on the left and bottom of the screen are the scroll bar gadgets. These gadgets are very similar to the scroll bar gadgets in other Amiga applications, and it is worthwhile to get familiar with their operation. The scroll bar appears as a long, thin rectangle inside a longer, slightly thicker rectangle. The length of the larger rectangle represents the size of the page in one dimension, and the length of the shorter rectangle is proportional to the amount of that page that is currently displayed on the screen. Thus, if one third of the vertical length of the document fits on the screen, then the scroll bar body (the inner rectangle) will be one third of the length of the scroll bar. The position of the scroll bar body within the scroll bar is also proportional to the position of the displayed text on the entire page.

To move the scroll bars and thus the page, you need to select the scroll bar body, and drag it in the direction you want the page to move. If you are currently looking at the middle third of the page and you wish to see the top third, you must move the pointer to the vertical scroll bar (on the right), select the body and hold the select button down, and move the mouse up. The scroll bar body will follow. When the position is correct, release the select button. The page will then be redrawn.

Alternatively, if you click the select button above or below the scroll bar body, the body will move its length in the appropriate direction. This is a quick way to get to the top of the page; simply move the mouse to the top of the vertical scroll bar, and click several times. The up and down arrow keys also move the page up and down on the screen.

In the lower-right hand corner is the resize gadget. Select and drag this gadget to resize the previewer window.

6.2 Preview Menus

There are six main menus on the preview program. The first is the 'project' menu, and it has five items on it. The first item is 'colors', which allows you to change the foreground and background colors for preview. (If the previewer is running on the workbench screen, this selection will be disabled; used Preferences to change the colors of the workbench screen.) Selecting this item brings up a colors requester with several gadgets on it. While the colors requester is on the screen, all other input to the program is blocked; you must select the 'use' (to retain your color changes) or 'cancel' (which restores the colors to their previous state and exits) gadgets to use the program further. The 'background' and 'foreground' gadgets select which color is currently being modified. The 'reset' gadget sets the colors back to what they were when the requester first appeared on the screen. Finally, the three scroll-bar gadgets allow you to change the intensities of red, green, and blue in the current color.

The second option is the 'close file' option, which is used to close the `.dvi` file so TEX can be run again. If this option is not used before TEX is rerun, it will complain that it can't open the `.dvi` file because the previewer has it open for reading.

The next option on the project menu is the 'release memory' option. As the previewer needs fonts to draw the pages on the screen, it loads them off the disk. When memory starts running low, preview starts to delete the fonts it loaded on a least-recently-used basis. However, preview can easily use a lot of memory for font storage. This second option instructs the program to delete all of the fonts from its memory. This is useful if you want to run another program and you suspect memory is low.

The fourth option on the project menu is the 'save configuration' option. This option saves the current state of the preview program, including screen mode, zoom and unzoom values, and colors. The next time the preview program is run, the settings of these values will be read from this configuration file.

The last option on the project menu is the 'quit' option; it behaves exactly as if you had selected the close gadget on the window by exiting the program.

The next two menus are the 'zoom' and 'unzoom' menus; these are used to select the resolution for these two modes. Either may take on values between 58 and 189 dots per inch, and the 'zoom' mode does not necessarily have to have a higher resolution than the 'unzoom' mode. If the mode corresponding to the menu you select is the currently active mode, the page will be immediately redrawn at the new resolution; otherwise, the new resolution will simply be stored for the next time you hit the zoom/unzoom gadget.

The fourth menu is the borders menu; the border consists of the lines that represent the edges of the paper on the screen. By default, preview (and TEX itself) assumes that the size of the paper is 8 1/2 by 11, and that there are one inch margins at the top and left. The first option on this menu allows you to toggle the borders on or off. The second option allows you to 'reset' the borders back to the default values, in case they get changed. The third option sets the page size to 'landscape', that is, 11 by 8 1/2, with one inch margins. (If the `.dvi` file contains a TEX 'special' command consisting of the word 'landscape', this is selected automatically.)

Finally, the fourth option of the 'borders' menu allows you to change the size and margins of the paper arbitrarily. When you select the 'set size' option, a window will be opened on the workbench screen, and the current size and margins displayed (in inches). Simply enter new values, or hit carriage return to keep the old values. As you enter the values, they will be checked for consistency; if any of the values you enter are unreasonable, the program will prompt you for another value. To exit quickly, simply hit the carriage return key four times.

The fifth menu, 'modes', allows you to change the operating mode of the preview. There are currently three modes, each with advantages and disadvantages.

The first and default mode is the interlace mode. In this mode, the page is drawn on a 640 by 400 pixel screen with interlace on. This high resolution allows fairly readable characters with a correct aspect ratio, but the flicker caused by the interlacing might be annoying. There is a section following this one that describes how to deal with this flicker. The second mode is the stretched mode. Here, the page is drawn on a 640 by 200 pixel screen, using the same fonts as for the interlace mode. The characters are still well defined, but the aspect ratio is off by a factor of two; hence the name. The screen is rock steady in this mode. The screen is also rock steady in the third mode, non-interlace, that is also drawn on a 640 by 200 screen. The aspect ratio is corrected by modifying the fonts as they are loaded; this creates much cruder but generally acceptable characters. Which of the three modes you use depends largely on which bothers you the least; the interlace mode and flickering, a bad aspect ratio, or crude characters.

The fourth selection on the 'modes' menu allows you to move the preview window to the workbench screen, or back to a custom screen. By default, the previewer opens on a custom screen. It runs faster on a custom screen, and usually takes less memory as well. In addition, you can select any of the three modes on a custom screen. Running on a workbench screen allows you easier access to your other windows, however. If you run the previewer on your workbench screen, you will only be able to use the interlace mode if your workbench is interlaced (this is selectable in Preferences), and you will only be able to use the stretched mode and the non-interlace mode if your workbench is not interlaced.

The last menu is the 'about,' or 'blurb,' menu. It displays a copyright notice and the version number of the software. Please let us know what this version number is when reporting bugs or missing features in the preview program.

6.3 Coping With Interlace

When the Amiga uses a display mode with 400 dots vertically, it redraws the screen thirty times a second rather than the normal sixty. This is called 'interlace mode' because of the way the screen is redrawn. To the human eye, the screen appears steady when redrawn at sixty frames per second, but unfortunately, the human eye notices the redrawing at the thirty frames per second. In the interlace mode, the screen appears to shake; this is called flicker. The Amiga's flicker may be handled or reduced in several ways. The first and best way is to buy a long-phosphor monitor. These are relatively expensive at the moment, but the price is dropping slowly. (Commodore will release a long-phosphor monitor soon for $500.) On a good long-phosphor monitor, the flicker should be non-existent. Another suggestion is to wear sunglasses while working in interlace mode. Don't laugh until you have tried it. Turning down both the room brightness and the monitor brightness helps a great deal, as does selecting the right combination of colors. If none of these suggestions works for you, the stretched and non-interlace modes eliminate the flicker altogether.

6.4 MoreRows

For those with screens that are larger than 640 by 400, such as PAL users or people familiar with the `morerows` program, the previewer will work just fine, automatically.

6.5 Preview Environment Variables

The preview program reads the settings of two environment variables: `texfonts`, which is used in the same way as TEX uses it, and `texpacked`, which contains a list of the directories to search for packed files. The default value of `texpacked` is `tex:pk`.

6.6 Font Caching

Supplied with AmigaTEX are approximately 1200 previewer fonts, occupying five of the eight distribution floppies. Typically, most of these fonts will never be used by a typical user; yet, which fonts are actually used varies widely. To help handle this large number of fonts, a font caching scheme has been implemented in the previewer.

The previewer will look for fonts in the places specified in the `texpacked` environment variable. If the required font is not found in any of the specified directories, the previewer will try to locate the font. To help it locate the font, a file called `tex:pk/fontvols` should exist. (It is supplied on the `tex` distribution floppy.) This file might look something like this:

```
cache 40 tex:pk
100dpi.69.76.83.91
100dpi.100.110.120
100dpi.131.144
100dpi.158.173
100dpi.207.249
```

The first line must start with the word 'cache' or 'nocache'. If the word is 'cache', then font caching is enabled, and if a font is found on one of the listed volumes, and there is room in the cache directory, it is moved there. Otherwise, the font is just loaded off the distribution floppy to be used. Next on the first line should be an number which specifies how many blocks to leave free in the cache directory. If the number of free blocks falls below this number, caching will be disabled. It is usually nice to leave a few blocks free, in case a program decides to write a configuration file to the `tex` directory. The third part of the first line is the directory to use for caching. Note that this must be a directory; it cannot just be a logical name. For instance, `ram:` is an invalid cache directory. Typically, `tex:pk` is used for the cache directory.

The remaining lines in the file contain the names of distribution font floppies. If you buy a printer driver, it will have some more font floppies which should be added to this list. The names of the floppies must be in a very specific format, to help the previewer and other drivers know on which floppy to find a font. For fonts that have the same vertical and horizontal resolutions, as the previewer fonts do, the name should be of the form

N`dpi.`N`.`N

for instance, where each N is replaced by a number. The first number doesn't matter much; it specifies the dots per inch of the device that the fonts on this floppy are intended for. The remaining numbers are important. For each, there must be a directory on the floppy with that number as a name, in which are a set of fonts at that resolution.

So, when the previewer cannot find a font in the default directories, it searches the cache directory. If the font is not there either, it searches through the list of font volumes until it finds the appropriate disk. It then requests that disk, copies the font into the cache directory, and proceeds. If caching is turned off, the font is not copied; it is just used directly off the floppy.

If the cache directory fills up, the previewer will print a warning, and turn caching off. Some of the fonts should probably be removed from the cache directory, or other files on that device removed, to make room for more fonts. An easy way to clean the cache is to just remove all of the fonts from the cache directory. This can be done with the command

```
1> delete tex:pk/#?/#?pk
```

(I'll bet you didn't know AmigaDOS was that smart.)

It is dangerous to use the ram disk as your font cache. The previewer will automatically cache fonts in ram during its normal run. However, if you really must, you can. Because the ram device allocates its blocks on demand, however, it will always appear full. Thus,

you should set the number of reserved blocks (in the first line of `tex:pk/fontvols`) to a negative number, say, -40, to insure that the files are copied.

7. Making TEX Larger or Smaller

Occasionally, TEX will complain with a message like

```
! TeX capacity exceeded, sorry [main memory size=18000].
```

(This particular message will only occur on a 512K Amiga.) AmigaTEX allows some limited control over internal array sizes with a file called `tex:sizes`. The variables you can modify are:

Variable	512K	Default	Maximum	Minimum	Size
memmax	18000	65500	65500	5800	4
memtop	18000	65500	65500	5800	(4)
triesize (initex)	8000	8000	30000	7500	13
triesize (tex)	6100	6100	6100	4	
poolsize	30000	50000	65500	27000	1
maxstrings	2000	5000	30000	1780	2
fontmemsize	6300	25000	65500	4900	4

The 512K column contains the default values for a 512K machine, as supplied in `texfiles:512/sizes`. The default column indicates the value used if no `tex:sizes` file is found. The maximum values are strict maximums; you cannot supply values greater than this without dire consequences. The minimum values are minimum to load plain TEX. The size column indicates how much memory each element takes. Memmax should always be greater than or equal to memtop, and is usually equal for maximum flexibility. IniTEX ignores memmax, and uses the memtop value instead. The memtop size value is in parentheses because it is just a position in a array of memsize elements.

The `tex:sizes` file must consist of just a list of seven numbers, in the order given in the table above. No error checking is performed, so you must insure that you have created the file correctly and that you have not exceeded the maximums listed above.

If you change `memtop`, you *must* create a new format file. This is why a separate plain format file is supplied for the 512K version of AmigaTEX. This is because the format file is just a memory dump of TEX's internals, and some internal pointers depend on memtop. A format file created with an IniTEX which used a different `tex:sizes` file than the current TEX is the most common reason for the error message

```
(Fatal format file error; I'm stymied)
```

All other values can be changed with no change to the format file. For instance, if you are running out of font space, you could easily increase the fontmem value to load more fonts. (On a 512K machine, you cannot increase it much without running out of room, however.)

8. LaTeX

LaTeX is supplied with AmigaTeX on the third distribution diskette. LaTeX is a macro package designed by Leslie Lamport that makes TeX easier to use by pre-packaging a number of document formats. It is a large package, however; it will not run on a 512K Amiga.

8.1 Installation

To run LaTeX, there are several files you will need to move onto your `TeX` floppy or hard disk. The first of these is the LaTeX format file, called `lplain.fmt`. If you use LaTeX exclusively, you might want to delete the `plain.fmt` file to make room for this file. You will probably have to delete some `.pk` files off the `TeX` floppy to make room if you are running from floppies. I recommend deleting the fonts starting from the higher resolutions, since these fonts are used less often and take more space.

You will also need to move all of the style files (with the `.sty` extension) from the LaTeX floppy into the `tex:inputs` directory. This can be accomplished with the following command:

```
1> copy latex:inputs/#?.sty tex:inputs
```

Again, you might have to delete some files to make room. Alternatively, to bring things up with a minimum of fuss, the following two environment variable defines will allow you to run LaTeX by simply swapping floppies as they are requested:

```
1> set texinputs=latex:inputs,tex:inputs
1> set texmacros=latex:macros,tex:macros
```

To tell TeX to load the `lplain.fmt` file rather than the `plain.fmt` file, you will need to execute the following set command:

```
1> set texformat=lplain
```

Alternatively, you can simply give TeX the format file name followed by an ampersand before the file name, as in

```
This is CTeX, Amiga Version 2.1 (no format preloaded)
**&lplain foo
```

if `foo.tex` is the name of your LaTeX source.

9. IniTEX

The program TEX has only about 312 control sequences built in. These control sequences are very primitive and thus quite difficult to use. Therefore, before processing a document, TEX loads in by default a macro package called plain that contains approximately 596 more control sequences built up from the primitives.

Parsing and processing the large amount of source in plain takes a long time. To eliminate this delay, the program IniTEX can be used to create a 'format file' that contains essentially a memory image of TEX after all of the macros have been loaded. This format file loads relatively quickly, since it needs only to be copied into memory.

To illustrate the process of creating a new format file, we shall assume that we want to create a preloaded `plain.fmt` file augmented with the verbatim macros supplied in `tex:inputs/verb.tex`. You will substitute your own macro package for `verb.tex`.

Before we get started, note that you cannot run IniTEX on an Amiga with less than a megabyte of RAM. If your Amiga does not have enough memory, borrow a friend's. To create your format file, you must make sure that you have at least 150K free on your working disk.

First, you must move the file containing your macros and the `tex.pool` file to your current directory. The following commands should do the trick:

```
1> copy tex:inputs/verb.tex verb.tex
1> copy texfiles:macros/tex.pool tex.pool
```

Insert the disks as requested. Now you have to tell TEX to look on the `TeXfiles` disk for some of its input files; specifically, the `plain.tex` and `hyphen.tex` files. This can be accomplished by:

```
1> set texinputs=.,texfiles:inputs
```

Now you can invoke IniTeX, and load the basic macros of plain:

```
1> texfiles:c/initex
This is CTeX, Amiga Version 2.1 (INITEX)
**plain
(texfiles:inputs/plain.tex Preloading the plain format: codes, registers,
parameters, fonts, more fonts, macros, math definitions, output routines,
hyphenation (texfiles:inputs/hyphen.tex))
*
```

The macros of plain have been loaded, so now you load your own:

```
*\input verb
(verb.tex)
*
```

Once all of the macros you wish to include have been loaded, you can create the actual format file by typing the command `\dump` to the splat prompt. IniTEX will create the format file and exit.

In this case, the format file will receive the name `plain.fmt`, since that is the first file we specified. If you replace the supplied `plain.fmt` file (in `tex:macros` with this one, all of your TEX documents will have access to your new macros automatically. However, if you want to use the regular plain package in addition to your augmented plain, you will have to rename this format file, and copy it to `tex:macros`. Then, to use it, you will have to specify on the command line

```
1> tex &foo bar
```

if you renamed the format file `foo.fmt` and you want to process the TEX source in `bar.tex` with those macros. Alternatively, you can omit the `&foo` part by executing the command:

```
1> set texformat=foo
```

which will tell TEX to use the `foo.fmt` file by default.

10. BibTeX

BibTeX is a bibliography database system for use with LaTeX. For basic documentation, please refer to the *LaTeX User's Guide and Reference Manual.* BibTeX is supplied on six files. The first is `TeXfiles:c/bibtex`, which is the executable; if you use BibTeX, you will probably want to move this into your `c:` directory. The remaining files are in `TeXfiles:inputs`, with the extension `.bst`; these are the standard bibliography styles. To use these, move the files corresponding to the styles you will use into the directory `TeX:inputs`. That is all you need to do.

BibTeX uses two environment variables. One is `texinputs`, which is used just as in TeX. The other is `texbib`, and it tells BibTeX where to find your bibliography data base. By default, this is set to the same thing as `texinputs`.

11. Using the RAM disk

Judicious use of the RAM disk can speed up TEX processing on the Amiga significantly. You will probably only be able to take advantage of some of the tricks illustrated in this section if you have at least one megabyte of RAM. Also note that the RAM disk driver supplied with the 1.2 release of the Amiga operating system is significantly faster than that supplied with 1.1.

11.1 Moving the Format File to RAM

To reduce the delay involved in loading the format file each time TEX is run, you might want to copy it into RAM. This will require approximately 111K of RAM for plain and 260K for LaTEX. A typical sequence of operations to perform this might be

```
1> makedir ram:macros
1> copy tex:macros/plain.fmt ram:macros
1> set texmacros=ram:macros,tex:macros
```

11.2 Moving Your Working Directory to RAM

Another possibility is to move your working directory to the RAM disk. This will reduce disk traffic dramatically. The only potential problem is that if the machine crashes or loses power, all of your files in the RAM disk will be destroyed. Therefore, you always want to edit your source file off the floppy, and only create the `.dvi` and `.log` files on the RAM disk. To do this, before you invoke TEX or the previewer, type

```
1> makedir ram:tex
1> cd ram:tex
```

The `makedir` command need only be executed once, but the `cd` command should be executed from both the preview and the TEX window. Let us say that your normal working directory is `df0:mysource`. You should also execute the following command before entering TEX, so it knows where to look for your input files:

```
1> set texinputs=df0:mysource,.,tex:inputs
```

Now TEX will look first on your working directory on the floppy, then on the current RAM directory, and finally in the `tex:inputs` directory. The editor should be invoked with `df0:mysource` as the current working directory so files will be read and written to the floppy rather than the volatile RAM disk.

11.3 Moving Everything to RAM

If you have a lot of extra memory, you might consider moving all of the TEX files to RAM disk. You can move all of the files on the standard distribution disk and set things up to work entirely off RAM disk with the following sequence of commands:

```
1> makedir ram:tex
1> copy tex: ram:tex all
```

Now go out and get a cup of coffee; this will take a while. When it is complete, remove the `TeX` floppy and type

```
1> assign tex: ram:tex
```

and follow the instructions in the previous section to move your working directory to RAM.

12. Importing Font Files

AmigaTEX is fully compatible with all versions of TEX. (An implementation cannot be called TEX unless it is fully compatible with the standard.) This allows you to import font files you have created on a mainframe with METAFONT, for instance. The `.tfm` file must be moved into the `tex:fonts` directory, and any `.pk` files must be moved into the appropriate subdirectory of `tex:pk`. Since METAFONT creates `.gf` files, a program to convert these files to `.pk` files is included. It is called `gftopk`; to run it, put the `.gf` file in your current directory, and invoke `gftopk` with the name of the `.gf` file. The conversion will be performed.

Likewise, if you have some old `.pxl` font files, you can convert those to `.pk` files with the program `pxtopk`; it is invoked in the same way as `gftopk`.

To use your local fonts with the previewer, you must create them at a magnification of 100 dots per inch, multiplied or divided by some power of $1.2^{1/2}$. Thus, for a full set of previewer fonts, you must create font files at resolutions of 69.44, 76.07, 83.33, 91.29, 100, 109.54, 120, 131.45, 144, 157.74, 172.8, for instance. Actually, you can select the resolutions you use most often and just create them.

When you download any font files, whether `.tfm`, `.pk`, `.gf`, `.pxl`, or whatever, you must be careful to download them as binary files. Refer to the documentation for the software you use to download files for details.

13. History

AmigaTEX is a TEX which functions exactly like the TEXs on any other machine, from DEC-2060s to IBM-PCs. There is no difference; a TEX source file developed on one machine is fully compatible with the TEXs on any other machine. How was the huge TEX program, whose user manual is almost 500 pages long, crammed into a 512K Amiga? This section describes the history of AmigaTEX; how it was developed and ported to the Amiga. It assumes that the reader is conversant about programming languages, TEX, and some computer operating systems.

AmigaTEX, as all current implementations of TEX, started life as the original source file of TEX. TEX was written by Donald Knuth in `WEB`, a system of structured programming and documentation. `WEB` is mostly a macro preprocessor for both TEX and Pascal, allowing one to program in Pascal and document in TEX, and providing publication quality program documentation. The program `tangle` takes a `.web` file and converts it into a `.pas` Pascal source file.

Unfortunately for those running Unix, few good Pascal compilers exist for Unix, and those that do are typically very expensive. Most Unix sites running TEX, therefore, were paying a significant performance penalty. Most Unix systems have good C compilers, though, so Tomas Rokicki and n^2 Computer Consultants decided to translate TEX into C. This translation was done automatically using the compiler tools available under Unix, and it was not long before CTEX came into existence. This version of TEX has been ported to almost everything running any brand of Unix, and almost invariably shows a marked speed improvement over the Pascal version.

When Tom Rokicki got his Amiga, it was natural that he wanted to run TEX on it. So he ordered a beta copy of the Manx C Compiler and moved all of the source files for CTEX onto floppies. (At that time, and even as this is being written, no good Pascal compilers exist for the Amiga.) Then he set to work.

There were bugs in the compiler. These had to be gotten around. Certain modifications had to be made to the source to make it work under the Amiga operating system. But finally, everything worked. Even then, hundreds of hours were invested in optimizing and rewriting major sections of code in 68000 assembly language to make the final program small and fast.

And then the real work started—writing the preview program. This program was coded completely from scratch. It had to manage the dozens and dozens of fonts that might be used, interpret `.dvi` files and `.pk` files, handle user interaction through gadgets and menus, and be fast. It was a significant effort, but things finally fell into place. And that is the story of AmigaTEX.

14. Software Support

Updates and revisions will be made available at a cost of $10 per floppy required for the update. Notification of updates will be sent to all licensed owners of AmigaTEX.

If you suspect you have found a bug in AmigaTEX, please mail a complete description of the error, including any commands and source files necessary to duplicate it, to us. Include your telephone number and when you can be reached, and we will get back to you as soon as possible.

If you have any questions about the operation of this program, please re-read the manual carefully. Then, reboot and start again. If you have trouble, we'll be happy to help answer any questions you might have. Our address is

AmigaTEX
n^2 Computer Consultants
Box 2736
College Station, TX 77841

TEX has been ported to dozens of computer systems, and currently runs on everything from the IBM-PC to the CRAY. For microcomputer versions of TEX, check out the advertisements in *BYTE* and other microcomputer magazines. For mainframe distributions, contact Maria Code at (408) 735-8006; she handles the distribution of TEX tapes. Note that she is not familiar with TEX, she only distributes the tapes, so please do not ask her technical questions. The cost for the tapes is nominal.

The TEX Users Group (TUG) is an organization dedicated to the exchange of information about TEX and its implementations. General meetings are held once a year, and short courses at several levels of 'TEXpertise' are given at several sites during the year. Tapes of these courses are available for rental. Site coordinators for implementations of TEX on particular machines help those with similar configurations, and the *TUGboat*, featuring articles on TEX, is published at least semiannually. Information is available from

TEX Users Group
c/o American Mathematical Society
P. O. Box 6248
Providence, RI 02940

15. Distribution File List

This is a list of all of the files on the eight distribution diskettes, and a brief explanation of their purpose. (This list may not be complete because of last minute changes to the distribution.)

```
TeX: (disk 1 of 8)
    fonts (dir)
```

These are all the font metric files that contain resolution independent information about each font. These are the standard 91 TEX fonts.

```
circle10.tfm        circlew10.tfm
cmb10.tfm           cmbsy10.tfm
cmbx10.tfm          cmbx12.tfm
cmbx5.tfm           cmbx6.tfm
cmbx7.tfm           cmbx8.tfm
cmbx9.tfm           cmbxsl10.tfm
cmbxti10.tfm        cmcsc10.tfm
cmdunh10.tfm        cmex10.tfm
cmff10.tfm          cmfi10.tfm
cmfib8.tfm          cminch.tfm
cmitt10.tfm         cmman.tfm
cmmi10.tfm          cmmi12.tfm
cmmi5.tfm           cmmi6.tfm
cmmi7.tfm           cmmi8.tfm
cmmi9.tfm           cmmib10.tfm
cmr10.tfm           cmr12.tfm
cmr17.tfm           cmr5.tfm
cmr6.tfm            cmr7.tfm
cmr8.tfm            cmr9.tfm
cmsa10.tfm          cmsl10.tfm
cmsl12.tfm          cmsl8.tfm
cmsl9.tfm           cmsltt10.tfm
cmss10.tfm          cmss12.tfm
cmss17.tfm          cmss8.tfm
cmss9.tfm           cmssbx10.tfm
cmssdc10.tfm        cmssi10.tfm
cmssi12.tfm         cmssi17.tfm
cmssi8.tfm          cmssi9.tfm
cmssq8.tfm          cmssqi8.tfm
cmsy10.tfm          cmsy5.tfm
cmsy6.tfm           cmsy7.tfm
cmsy8.tfm           cmsy9.tfm
cmtcsc10.tfm        cmtex10.tfm
cmtex8.tfm          cmtex9.tfm
cmti10.tfm          cmti12.tfm
cmti7.tfm           cmti8.tfm
```

```
  cmti9.tfm                       cmtt10.TFM
  cmtt12.tfm                      cmtt8.tfm
  cmtt9.tfm                       cmu10.tfm
  cmvtt10.tfm                     lasy10.tfm
  lasy5.tfm                       lasy6.tfm
  lasy7.tfm                       lasy8.tfm
  lasy9.tfm                       lasyb10.tfm
  line10.tfm                      linew10.tfm
  logo10.tfm                      logo8.tfm
  logobf10.tfm                    logosl10.tfm
  manfnt.tfm                      trip.tfm
```

The `nul.tex` file is used to get out of TEX's 'could not open file, type another file name' loop. The `verb.tex` file has a set of macros for verbatim mode; the `testfont.tex` file allows you to sample particular fonts. The `plain.fmt` file is the default plain TEX format file for a machine with more than 512K. The `tex.pool` file has all of TEX's strings for IniTEX.

```
inputs (dir)
  nul.tex                         null.tex
  testfont.tex                    verb.tex
macros (dir)
  plain.fmt                       tex.pool
```

The `pk` directory contains the cache for all of the packed files. Supplied on the `TeX` disk are a few sample fonts; the rest will be loaded on demand from the font distribution disks. The file `TeX:pk/fontvols` contains a list of the distribution fonts and a few other caching parameters.

```
pk (dir)
  fontvols
     69 (dir)
       cmr10.69pk                       cmtt10.69pk
     76 (dir)
       cmr10.76pk                       cmtt10.76pk
     83 (dir)
       cmbx10.83pk                      cmr10.83pk
       cmtt10.83pk
     91 (dir)
       cmbx10.91pk                      cmr10.91pk
       cmtt10.91pk
     100 (dir)
       cmbx10.100pk                     cmr10.100pk
       cmtt10.100pk
     110 (dir)
       cmbx10.110pk                     cmr10.110pk
       cmtt10.110pk
     120 (dir)
       cmbx10.120pk                     cmr10.120pk
```

```
    cmtt10.120pk
  131 (dir)
    cmbx10.131pk                    cmr10.131pk
    cmtt10.131pk
  144 (dir)
    cmbx10.144pk                    cmr10.144pk
    cmtt10.144pk
  158 (dir)
    cmbx10.158pk
  173 (dir)
    cmbx10.173pk
```

The openwindows command file is a typical way to open three windows in which to run TEX, the preview program, and an editor. The samples directory contains three very short, very simple TEX examples—the default `story.tex` from the TEXbook, a program that draws a heart out of characters, and a routine that does nothing but print out primes forever.

```
s (dir)
  openwindows                   to-512
samples (dir)
  story.tex
```

The second disk contains the executables, all of the special files necessary on a 512K system, and the files necessary to recreate a `plain.fmt` file. It also contains this manual. Execute the `to-512` command file to install TEX on a 512K system. It also contains a few sample files for TEX and LaTEX.

```
TeXfiles:  (disk 2 of 8)
  512 (dir)
    plain.fmt                   sizes
  c (dir)
    bibtex                      gftopk
    initex                      pktype
    preview                     pxtopk
    set                         tex
  inputs (dir)
    hyphen.tex                  plain.tex
  manual (dir)
    amigatex.tex
  samples (dir)
    addendum.tex                heart.tex
    idx.tex                     lablst.tex
    letter.tex                  local.tex
    primes.tex                  sample.tex
    small.tex                   story.tex
    testpage.tex                tryfonts.tex
```

The third floppy contains everything necessary to run LaTeX. The `format` directory contains the main input files to create `lplain.fmt`. To run LaTeX, all of the files in the `inputs` directory should be accessible, and the `lplain.fmt` file from the `macros` directory should be loadable by TeX.

```
format (dir)
  latex.tex                       lfonts.tex
  lplain.tex
inputs (dir)
  abbrv.bst                       alpha.bst
  art10.sty                       art11.sty
  art12.sty                       article.sty
  bezier.sty                      bk10.sty
  bk11.sty                        bk12.sty
  book.sty                        fleqn.sty
  ieeetr.bst                      ifthen.sty
  leqno.sty                       letter.sty
  makeidx.sty                     openbib.sty
  plain.bst                       proc.sty
  rep10.sty                       rep11.sty
  rep12.sty                       report.sty
  showidx.sty                     slides.sty
  suthesis.sty                    titlepage.sty
  twocolumn.sty                   unsrt.bst
macros (dir)
  lplain.fmt
```

The remaining five floppies contain fonts. Each floppy contains two or more sets of fonts of one size. For instance, the floppy `100dpi.131.144` contains fonts at 131 dots per inch in the subdirectory `131`, and fonts at 144 dots per inch in the subdirectory `144`.

16. Glossary

This section contains the definitions of some of the more obscure terms used in this manual.

boot To start or restart the Amiga by pressing down the *CNTRL* key and both Amiga keys (the fancy 'A's on either side of the space bar). Used as either a verb or a noun; comes perhaps inappropriately from the phrase 'lifting one by one's bootstraps'.

CLI Command Language Interpreter; a program that takes the user's input and parses it into commands and arguments, loads the commands and executes it. This is the traditional mode of operation of a computer, in contrast with the desktop metaphor.

desktop metaphor A method of user interface where files are represented as icons, programs as windows, and the user 'executes' commands by selecting, dragging, and pointing with the mouse. For instance, to delete a file using the desktop metaphor, you would click on the icon you want to delete, and drag the icon over to the trashcan, where you would let go of the select button.

double click Two rapid presses of the select button; this is the way an icon is activated in the Amiga. If the icon represents a program, the program is run; if the icon represents a directory, the directory is opened and a window showing the contents of the directory is displayed.

driver A program that takes a TEX device-independent (or `.dvi`) file and interprets it, loading the rasters for the fonts as necessary, and sends the required commands to a particular printer or other output device to produce the document described.

`.dvi` file A device-independent file is the output of the TEX program; it contains a description of the typeset document in a compact form that is resolution and output device independent.

environment variables Global variables that the user can set and which are accessible to programs in the machine. An environment variable can be used to set some option or convey some information to a program; for instance, they might be used to tell TEX where to find font files in a user-configured system.

`.fmt` file A format file is a compact form of a set of TEX macros that is loaded in every time TEX is run to make those macros available. The `fmt` file is created by iniTEX and consists of essentially a memory image of TEX after the macros have been loaded. The default `fmt` file is called plain TEX; LaTEX uses another format file called `lplain.fmt`. The format files must reside in the `tex:macros` directory.

font A collection of printing characters of one style or size; for instance, `cmr10` is Computer Modern Roman at 10 points. The word font is used to refer to both a particular style, independent of size, and a particular style at each individual size. AmigaTEX is currently supplied with the standard TEX 91 fonts; many different sizes of these fonts are distributed for the previewer.

gadget Some graphics imagery that, when selected with the mouse, starts some operation. For instance, the window-to-front gadget is in the upper right hand corner of a window, and when you select it, it brings that window in front of any other windows that might overlap it.

icon Some graphics imagery that represents a file, program, device, or directory. These can be activated or opened by double-clicking on them with the mouse.

`.info` file A small file containing the graphics imagery of the icon; associated with an executable, directory, or data file. When the executable, directory, or data file is

copied, renamed, or deleted, this file should also be copied, renamed, or deleted, to keep the desktop metaphor consistent.

iniTEX The iniTEX program is a larger version of TEX that takes more memory to run; its purpose is to create format files from a set of macros.

interlace The process of refreshing a video screen at half the normal rate by drawing alternate lines in each scan. The Amiga uses interlace for the screen modes that use 400 lines vertically; with a normal monitor, interlace can cause flicker that can be annoying.

Intuition The set of routines in the Amiga that controls the user interface, including the windowing, gadget, and menu routines. A properly written program on the Amiga will follow certain guidelines established by Intuition to allow the simultaneous and convenient use of many several different programs in the Amiga operating environment.

kerning The addition or removal of spaces between characters due to their shape. For instance, in the word 'AVAILABLE', the 'A' and 'V' are moved closer together; compare this with 'AVAILABLE', with the kerning turned off. TEX handles kerning automatically.

LaTEX A macro package intended to both make TEX easier to use and more powerful, LaTEX was written by Leslie Lamport. It is supplied with AmigaTEX, but requires more than 512K to run.

ligature A ligature is the combination of two or more characters into one character. For instance, the 'f' and 'i' in 'file' is a ligature; compare this with 'file', using separate characters instead. TEX handles ligatures automatically; no special characters need be typed.

`.log` file A file that contains all of the terminal output during a run of TEX; it is created automatically along with the device-independent file for later perusal of any errors. It serves no purpose other than user enlightenment, so it can be safely deleted at any time.

loop mode When TEX is run on the Amiga in loop mode, the program does not exit after finishing a job. Rather, it restarts, and waits for another file name. This is to eliminate the delay in loading the TEX executable off the disk. TEX runs in the loop mode when it is invoked with no file name, or when it is given the `-r` option before the file name.

menu button The right mouse button. This is called a menu button because pressing it displays the menus for the currently active window on the screen title bar. To select a particular menu item, you hold the menu button down, move the mouse to the menu name and then to one of the items that will appear, and then release the menu button.

`.pk` file A packed file contains raster information for a particular font at a particular resolution in a compressed format. TEX does not use these files at all, but the preview program and any other drivers require them at certain resolutions.

plain format A basic macro package that contains several hundred predefined macros; TEX loads these in at the start of each job unless another format file is specified.

pool file TEX has over 25000 characters worth of error and other messages. Rather than place these in the executable image, they are put into a special `tex.pool` file that iniTEX reads and places in every format file. If iniTEX complains about not being able to find the `tex.pool` file, this file should be placed in the current working directory.

RAM disk A 'disk' whose files are stored in the computer's memory rather than on magnetic media. It is generally faster than floppies or hard disks, but is volatile; if the computer is booted or powered down, all of the files in the RAM disk are destroyed. It is usually used for fast, temporary storage. Proper use of the RAM disk can make the computer system run noticeably faster, especially on a system with a lot of memory.

raster A graphics image stored as actual pixels rather than an analytical description. For instance, the TEX packed files contain the raster, or pixel by pixel representation, of each character in a particular font. Raster graphics are by their very nature resolution dependent; analytical graphics are not.

reboot To boot again; see *boot.*

requester A window opened when a program or the operating system requires some response or action from the user. With a requester, all further input to the program is blocked until the requester is answered. Requesters might require you to insert a particular disk or ask you to change the colors of the screen.

resolution In raster graphics, the resolution refers to the number of dots (or pixels) per unit measure, usually per inch. For instance, a dot matrix printer might print at a resolution of 72 dots per inch, a laser printer at 300 dots per inch, and a typesetting machine at 720 dots per inch.

screen Under Intuition, a screen is a display memory region in which one or several programs may operate. Each screen has its own video characteristics, such as number of colors and resolution. Screens can overlap each other vertically, but each must be the full width of the video monitor. Preview opens up its own custom screen, while TEX runs in the standard workbench screen.

select button The left mouse button; used to select gadgets or icons. Gadgets are selected by moving the pointer over the gadget and hitting the select button. Icons are selected the same way, but are activated by hitting the select button twice rapidly. The select button can be used to drag screens or windows around, or to resize them, by selecting the proper gadget and holding the select button down while moving the mouse.

splat Another name for the asterisk (`*`).

`.sty` file An auxillary file read in by LaTEX that describes the layout of a particular document style; `rep10.sty` describes a ten-point version of the report style. These must reside in the `tex:inputs` directory when running LaTEX.

`.tfm` file A font metric file contains resolution-independent information about a particular font, such as character sizes and ligature and kerning information. TEX reads these files when you declare a new font; they must reside in the `tex:fonts` directory.

title bar The top portion of a window or screen that contains the title of the window or screen and usually the to-front and to-back gadgets. The drag gadget is usually all portions of the title bar not occupied by other gadgets. Menus will appear in the title bar of a screen.

TRIPOS An English message-based operating system on which AmigaDOS is based.

window An Intuition window is a rectangular section of a screen that a program uses for user interaction. Several windows may all reside on one screen, and can overlap each other arbitrarily. Each window must share all video attributes of the screen on which it resides; for instance, you cannot open a thirty-two color window on the four color workbench screen.

workbench The name of the standard screen on which Intuition runs its desktop metaphor user interaction.