SLAC-TN-91-1
February 15, 1991
TN

Introduction To AmigaTEX*

AMANDA J. WEINSTEIN

and

MARVIN WEINSTEIN

Stanford Linear Accelerator Center
Stanford University, Stanford, California 94309

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.

Table of Contents

1. Introduction

1.1 What Is AmigaTEX ?

AmigaTEX is the Amiga's version of TEX, Donald Knuth's computer typesetting system, designed to take a text file and produce a printed manuscript. People who have used TEX on the IBM mainframe know that producing a final copy using the TEX system involves three independent steps. First, one uses an editor, e.g. XEDIT, to prepare the text file; next, one TEX's the file in order to produce a device independent (DVI) file; finally, one uses the program IMATEX to send the DVI file to a laser printer. Using TEX on a personal computer involves the same sequence of steps. In the case of the Commodore Amiga, the editor is probably TxEd Plus and the TEX program will be AmigaTEX. One major difference between the IBM mainframe and the Amiga environment is that one does not have to print a TEX file in order to read it. One can instead use another program, Preview, to display the formatted file on the screen. Obviously, the ability to preview a TEX file before printing it saves paper and time; especially if one is making many revisions to the text, or changes in page layout.

In general, the output produced by TEX is far superior to that produced by a "What You See Is What You Get" (WYSIWYG) word processor; this is especially true for technical documents. Also, because the files which TEX uses are plain text files, transporting such files from site to site or machine to machine is straightforward. For these reasons SLAC uses TEX to prepare most technical documents. The only disadvantage to using TEX is that learning to use all of the power at your command takes some time. Fortunately, most people don't want to use all this power, they just want to get their work done in as straightforward a manner as possible. Learning to do this is not so difficult, especially if one begins on a machine like the Amiga which can be set up to provide a user-friendly interface to TEX. The aim of this document is to provide an introduction to the Amiga's TEX environment for the novice user.

1.2 A Quick Overview

We have configured the Amiga's TEX environment so that it provides the novice TEX user, as well as the TEX afficionado, with almost WYSIWYG behavior. This means that a person wishing to create a document opens an editor window and

begins typing. If and when he wishes to preview the document he either uses the mouse to select a menu-item, or executes its equivalent keyboard-sequence. This procedure, executed entirely from within the confines of the editor, causes the programs TEX and Preview to begin running, if they are not already there. It also causes the current version of the document to be automatically displayed in the previewer's window. If a TEX error should occur, TEX stops processing the file and another editor window, containing the TEX log file, opens. At the same time, the window containing the text file being processed is adjusted so that the offending line is displayed and the cursor positioned immediately following the location at which TEX believes the error occurred.

This TEX environment is quite user-friendly, but not entirely self-explanatory. To work with AmigaTEX you will need to understand the Amiga's filing system and learn how to operate the standard file requester. You will also need to learn your way around TxEd, the Amiga text editor currently in use at SLAC, as well as the document previewer provided by AmigaTEX. We will attempt to explain these aspects of the Amiga's TEX environment as clearly and intuitively as possible. We will assume that the reader is already familiar with TEX. It will also probably be true that our average reader will be acquainted with the filing system and TEX environment available on the SLAC IBM mainframe; however, no knowledge of how TEX works on an IBM mainframe is necessary. The Amiga user who wishes to get his work done as quickly as possible should limit their reading to the first five chapters of this tutorial. The sixth chapter introduces special facilities provided in order to make it easier to produce SLAC letters, SLAC memos and technical papers. These are the Amiga versions of the IBM execs PHYZLETT, PHYZMEMO and PHYZPUB. The final chapter is included for those readers who do not feel comfortable unless they know exactly what is going on, and where the files which control the default behavior of TEX, PhyzLett, PhyzMemo, etc. are located.

1.3 For Experienced Users

SLAC's AmigaTEX environment has recently been updated and improved. Experienced users may wish to glance through this document to familiarize themselves with newer features. Hopefully, this tutorial will prove useful to old and new users alike.

2. A Sample TEX Session

2.1 Notation

This chapter consists of several sample TEX sessions, so that you can try out AmigaTEX before getting into details. Text to be entered from the keyboard is set in a typewriter font and preceded by a > prompt. When you type this text *do not* type the > prompt.

2.2 A Simple Test

Before plunging into details, let us try out a sample session, to get a feeling for how things work. To begin, find the window which has the word WShell in its titlebar. Once you have found it, position your mouse pointer over the center of the window and then click the left button on your mouse once. This *activates* the window so that it is ready to accept keyboard input. Verify that the window is activated by typing

```
> e test.tex,
```

You should see the letters appear in the CLI window as you type them. When you are finished hit the carriage return key. The carriage return key is the big key located on the right hand side of the keyboard. It is labelled with a crooked arrow.

After you hit the carriage return another window will open. This is where we will create the file `test.tex`. The window should now be the active window. Test to be sure that this is true by typing the line

```
This is a test file.
```

The words you typed should have appeared in the new window. If not, then move your mouse pointer over the window and click the left mouse button to activate it. Once you are sure the window is active, type the line.

This is our first TEX file. To preview this file, hit the key labelled **F1**, located in the upper left-hand corner of the keyboard. Now, wait and watch what happens.

The first thing that will happen is that the window containing the TEX file will automatically resize itself. Next, another, smaller, CLI window will open towards the lower righthand corner of the screen. The TEX program will run in this window and display TEX messages. Finally, a third window will open in front of the other windows. This is the previewer window. Once the TEX program has finished processing the file the results will appear in the previewer window.

2.3 When Errors Occur

Assuming that you have been successful, let us now go back to the editor window and add some material to the file. To move the previewer window in front of, or behind, the other windows you click on the front/back gadgets. If you are using an Amiga 2000, or and Amiga 2500, the operation of the front/back gadgets will be a little different than for an Amiga 3000.

- **The Amiga 2000**
  If you are using an Amiga 2000 then you will see two small rectangles located in the upper right-hand corner of the previewer window, with two arrows located to their left. These rectangles are the front and back gadgets. If you click on the rightmost gadget it causes the window to move in front of all other windows. Clicking on the gadget to its left will cause the window to move behind all other windows. Since we want to reveal the window containing our test file, take your mouse, position the pointer on the left-hand gadget, and click on the left mouse button. If you did this correctly the editor window containing the line

    ```
    This is a test file.
    ```

  should now be visible.

- **The Amiga 3000**
  If you are using an Amiga 3000 then you will also two small rectangles, but only the rightmost gadget is for moving the window in front of or behind other windows. The gadget to its left is a zoom gadget, and it will cause the window to change its size. To move the previewer behind the other windows position the mouse over the right-hand gadget and click once.

Now that the file test.tex is in front of you, activate its window by clicking on it with the mouse, and then type in the following file.

```
Once upon a time, in a magical land called {\it Prydain}, there
lived a little boy named Taran.
Taran was a foundling, raised by the enchanter {\bf Dalben} and the
blacksmith Coll ap Colfrewr.
One day, Taran lost his way in the woods, with disastrous results.
\nodent
Bye. This file is to be TeX-ed now.
```

cursor is located at the beginning of the text hold down the Right-Amiga key, i.e. the key marked with an *A* which is located just to the right of the space bar, and while holding it, hit the key labelled **B**. You should see the cursor change color. Now, using the mouse, or arrow keys, position the cursor at some point past the end of the block of text you wish to preview. The entire block of text should now change color. When this happens hit **F4**. If you have done everything correctly just the hilighted material should be TEXed and sent to the previewer.

2.5 The Material Which Follows

You have now seen how simple it is to create a file and then TEX and preview it. There are, however, a few more things you really have to become familiar with before you can use the Amiga's TEX environment comfortably. First and foremost, you must understand the Amiga's filing system. If you do not learn how this works you will not be able to find the documents which you create. Next, you need to learn something about the features provided for changing files. For example, you want to learn how to delete and undelete lines and chunks of text, how to change one phrase to another, etc. You also want to learn the correct words for talking about the Amiga setup so that you can communicate effectively with other users. You will find an introduction to this material in the chapters which follow.

3. The Amiga Filing System and File Requester

3.1 The Filing System

The Amiga has its own method for organizing the files that it has stored either in memory, or on a floppy disk. Since this filing system is very different from that used by the IBM mainframe, IBM users should read this explanation carefully.

Below is a sample list of an SLAC IBM mainframe user's files. To many of you, this list will be very familiar; the filename is the name of the file, the filetype tells you what kind of file it is, and the *filemode* tells you on which disk the file is stored. The middle four columns — `Format`, `Lrecl`, `Records`, and `Blocks` — are descriptions of how the file has been stored; the last two columns tell you the date and time that the file was last changed.

```
AMANDA   FILELIST A0  V 108  Trunc=108 Size=60 Line=1 Col=1 Alt=1
Cmd  Filename Filetype Fm Format Lrecl     Records     Blocks   Date     Time
     FILREQ   BINARY   A1 F         320         140         44  8/16/90 13:48:24
     AMANDA   CONSLAST A1 F          80           2          1  8/21/90 18:00:58
     ERRATA   DVI      A1 F        1024           1          1  8/20/90 16:33:03
     PF       EXEC     A1 V          41          16          1  7/02/90 23:05:17
     PROFILE  EXEC     A1 V          47          39          1  7/02/90 23:04:32
     LASTING  GLOBALV  A1 V          32           2          1  8/24/90 10:52:54
     THY      SYNONYM  A1 F          80          13          2  7/02/90 23:04:54
     VLTMACS  TEX      A1 V          77         349         10  8/16/90 14:41:14
     WEDGE    TEX      A1 V          81          88          3  8/16/90 16:33:42
     VLTICON  TOPDRAW  A1 V          30         917         23  8/13/90 13:21:36
     PROFILE  XEDIT    A1 V          59          81          2  7/02/90 22:58:08

1= Help      2= Browse   3= Quit    4= XEDIT       5= Sort(date)  6= Sort(size)
7= Backward  8= Forward  9= FL /n  10= Refresh    11= Sort(type) 12= Cursor

====>

                                                          X E D I T  1File
```

When referring to an IBM file you use the contents of the first three columns, in the displayed order. For instance, to erase the first file on the list you would type `ERASE FILREQ BINARY A1`. Actually, `FILREQ` is the name of the file, the other two words describe the type of file, and where it is located. On the IBM one tends to blur this distinction and think of the entire expression `FILREQ BINARY A1` as the name of the file.

Let us now look at a comparable Amiga filelisting, obtained by typing `list sys:` in a CLI (an abbreviation for command line interface). A CLI is the name used for an Amiga window which accepts and displays input from the keyboard.

```
martin.tex              empty ----rwed 08-15-90 15:52:20
blinker.scp               184 ----rwed 07-20-90 09:57:26
myphyx.tex               2527 ----rwed 04-13-88 10:56:08
vltdocs                   Dir ----rwed Today    10:34:19
Interact                  Dir ----rwed Friday   12:01:36
schedules                 Dir ---arwed 09-09-88 17:04:14
usrc                      Dir ---arwed Friday   11:57:04
```

This listing looks quite different. The first thing you should notice is that you can't find a filemode listed anywhere. Filemodes aren't there for two reasons. First, because all of the files in this list share the same filemode— or at least the Amiga's equivalent of a filemode — the Amiga doesn't bother to list it. Second, on the Amiga, the disks where files are stored are called *devices*, not filemodes, and indicated differently.

A device can represent a partition on your hard disk, your temporary memory (this is similar to temporary storage disks on the mainframe), or a floppy disk. Whereas IBM filemodes are listed at the end of a file's description, the Amiga puts device names at the beginning of a file's description and follows the device name by a colon. All of the files on the above list share the same device name, which happens to be **SYS:**, so to fully describe the first file on the above list you would type `sys:martin.tex`.

The next difference, which you have probably noticed, is that an IBM name which would be written as `FILENAME FILETYPE` is written as `FILENAME.FILETYPE` on the Amiga. Thus, the first two filenames appearing in the leftmost column of the file listing, i.e. `martin.tex` and `blinker.scp`, would be called `MARTIN TEX` and `BLINKER SCP` on the IBM. Not only is there a difference in typing filenames on the

Amiga, but the terminology is different. On the Amiga, filetypes are known as *file extensions*. Furthermore, it is not really necessary to include a file extension in an Amiga filename, although it is considered good form to do so. This is just a trick for including information about the file in it's name. Once the file extension is part of a file's name, however, you *must* always include the file extension when referring to that file. One more difference between the Amiga and IBM is that filenames can be much longer than eight letters.

NOTE : Appending the extension `.tex` to TEX files is absolutely required if you want the WYSIWYG-like interface to work properly. More on this later.

There are other differences between the way in which files are listed on the IBM and Amiga. The last two columns of the Amiga filelist simply describe the date and time of the last changes made to each file, and the information in the third column from the right deals with how the files are stored, which is usually unimportant to most users. The entries in the second column from the left, on the other hand, give us crucial information. If the listing is for a file, then this column will contain a number which tells you how big the file is in bytes. However, as we see from the entry reproduced below, this column does not always contain a number.

```
vltdocs                    Dir ----rwed Today    10:34:19
```

This entry differs from the other entries in two ways. First, the *filename*—`vltdocs`—has no file extension. Second, the entry in the second column says "*Dir.*" What does this mean? Well, here is the most profound difference between the mainframe and Amiga filing systems, `vltdocs` *isn't a file, it's a directory.*

3.1.1 What Is A Directory?

We've already explained that Amiga device names are similar to IBM filemodes in that they indicate the disk where a file is stored. IBM file storage isn't structured any further than division according to filemodes. If you want to organize or categorize your files, you sort a filelist according to date, filetype, or the size of the file.

There are times when you want to store a set of files of varying type, size, etc. which have only one thing in common—subject matter. If you have two TEX preprints, a TEX macro file, several graphics files, and a Rexx exec, all of them related to your current project, it is convenient to store them all in one place. That way, each time you work on the project, you know immediately which files are related to it. On the IBM, you can't do this, because nothing in the file system allows for it. The Amiga, however, uses directories to make this possible.

If you picture your Amiga as a desk, and devices as drawers, then directories are infinitely stretchable file folders within those drawers. You can stuff as many files

as you like into such a folder; the folder will just expand to take up more of the space within the drawer. When the folder grows to fill the entire drawer, however, it cannot expand beyond the drawer's rigid walls, even though the folder itself is infinitely elastic. The same is true for a directory. As the number of files within a directory increases, the directory fills up more of the device's memory; if a directory takes up all of a device's memory, then the directory can no longer accept and store new files.

When you list the files on the device, each directory is listed, but the files in the directory are not shown. If you ask for a filelist of a particular directory, *then* all the files in that particular directory are shown. Making directories is a fairly simple task. For instance, say you wanted to create a directory in **SYS:** (**SYS:** is a device) which will store all files related to an upcoming SLAC conference. You would go into a CLI and type

```
> makedir sys:conference
```

Now you have a directory called `conference` within your **SYS:** device.

Directories can be subdivided into even smaller portions, known as subdirectories. Subdirectories, like directories, can be pictured as elastic file folders. When you stuff files into a subdirectory, the subdirectory stretches; because of this, the directory stretches as well. Again, both the subdirectory and the directory that contains it are bounded by the memory limitations of the device.

Subdirectories can have sub-subdirectories inside them, and these subdirectories can in turn contain sub-sub-subdirectories, and so on. To save time, we drop the extra "sub's" and just abbreviate sub-subdirectories, etc., as subdirectories. These directories and subdirectories must be included in file descriptions. All directories and subdirectories are placed between the device name and the filename, and the directory and subdirectories are separated from one another by slashes, like this:

```
sys:conference/guests/invitations/addresses.tex
```

If a subdirectory is to the right of another subdirectory, than the right-hand subdirectory is within the left-hand subdirectory. The device name, directory name, and subdirectory names are known collectively as the *path name* or *path specification*, while the name of the file itself is referred to as the *filename.*

Finally, to create a subdirectory, you go into a CLI and type

```
> makedir <device name>:<directory>/<subdirectory>...
```

Note that those portions of the above entry surrounded by ⟨ ⟩ are not part of the command itself; that is, you do not type them into the CLI. Rather, words surrounded by ⟨ ⟩ are known as *parameters.* Parameters indicate that you should type a certain kind of name or number at that point in the command. For instance, the ⟨device name⟩ in the above entry indicates that you type a device name, such as sys: or df0:, just after `makedir`. Furthermore, *you cannot create a directory*

and a correlated subdirectory simultaneously. In other words, in order to make a subdirectory, you must first have created the directory that is to contain the subdirectory. If you wanted to create sub-subdirectory c, for example, you would have to issue a sequence of `makedir` commands, like this:

```
> makedir sys:a
> makedir sys:a/b
> makedir sys:a/b/c
```

3.2 Using The WorkBench

To this point we have discussed everything from the point of view of the CLI. We have done this under the assumption that most of our readers are moving to the Amiga from the IBM mainframe, or an IBM PC, and so are more comfortable typing commands. The Amiga, however, has both a command line (keyboard) interface, and the graphical user interface, (GUI), similar to the interface familiar to Macintosh users. If you use this interface to manage your files, then you have to become familiar with the visual metaphors used to describe devices, directories and subdirectories, and files. Devices are indicated by displaying a stylized picture of a 3.5 inch floppy disk. This picture is displayed on the WorkBench screen, and the name of the device is displayed under the picture.

The visual metaphor, or icon, for a directory located on a disk, is a stylized picture of a drawer. To display the directories contained on a device one simply moves the mouse pointer over the disk icon and then clicks the left mouse-button twice in succession. If you do this rapidly enough a window will open and the drawers, directories, located on this disk will be displayed in the window. There are several things you should know about the graphical user interface.

- The window which opens when you double click on a disk icon *is not* a CLI window. You can verify this by typing something on the keyboard and noting that nothing happens.
- There is a difference between the way in which drawers are displayed under AmigaDOS 1.3 and AmigaDOS 2.0. AmigaDOS is the name given to the operating system which controls the Amiga computer. Recently this operating system has undergone a major upgrade. The version of the operating system that most of you who own Amiga 2000's have is called AmigaDOS 1.3. The newer and more powerful operating system is called AmigaDOS 2.0.

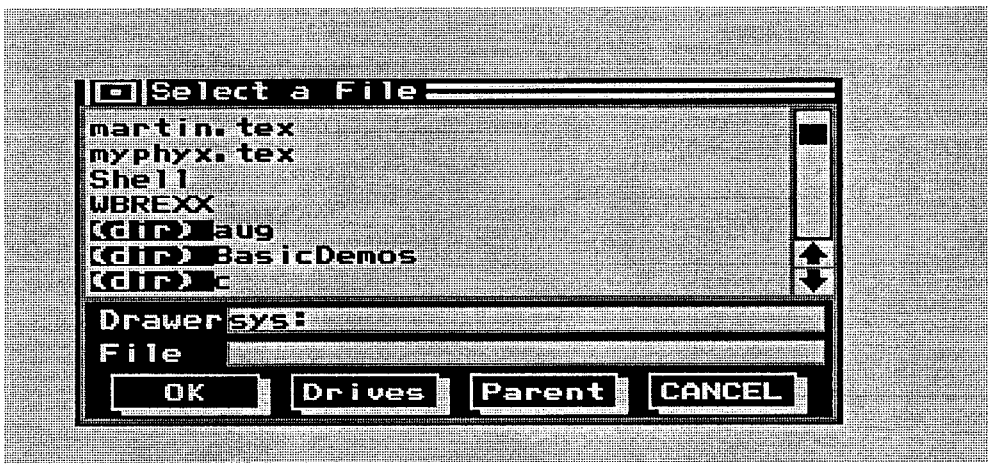
  Under AmigaDOS 1.3, drawers and files are displayed only if they have icons. These icons are the `.info` files which are listed in the CLI when you use the `list` command. When you make a directory by means of the `makedir` command, then no icon is created. This means that although you will be able to see this directory from the CLI by typing either the `dir` or `list`

commands, the directory will not be displayed as an icon on the WorkBench. To create a directory, or drawer, which will be displayed as an icon you must do it from the WorkBench. Read the manual which came with your Amiga to find out how this is done.

Under AmigaDOS 2.0 the GUI has been made much more powerful. In particular, it can display drawers and files which do not have `.info` files associated with them. Those of you who are fortunate enough to be using Amiga 3000's are already using AmigaDOS 2.0. The rest of you should upgrade your machines to AmigaDOS 2.0 when the option becomes available. The manual which comes with 2.0 does a very good job of explaining how to display and/or manipulate disks, files and directories from the WorkBench. Read it if you wish to work entirely, or mostly, from the WorkBench.

3.3 The File Requester

When you are inside the text editor, TxEd, you can open and save files. When you ask to load a file for viewing, or when you want to give TxEd the path and filename under which a file is to be saved, the file requester appears. The file requester allows you to specify the pathname and filename of the file you wish to load or save. It will look something like the picture below.



3.3.1 An Overview

The file requester is endowed with a plethora of gadgets, as are most Amiga requesters. Two of these gadgets are the long rectangular boxes at the bottom of the window, the first one labelled **drawer** and the second **file**. These are the string gadgets wherein you type the path name and filename of whatever file you want. Remember, we use the term *path of a file* to refer to both the **device:** and the

directory/subdir/subdir/.../ on that device, wherein a file resides. We use the term *filename* to mean the name of the file with no reference to its path. When we wish to talk about the combination **dev:dir/subdir/filename** we call it the full filename. Thus, for example, the full filename SYS:DEVS/KEYMAPS/USA1 has the path designation SYS:DEVS/KEYMAPS, and filename USA1.

3.3.2 The Drawer Gadget

The string gadget wherein you type the path designation is labelled **drawer**. This is of course a misuse of the word, if you think about the WorkBench notion of a drawer, which corresponds to a directory on a device and not to a full path. The second string gadget, into which you type the file's name, is labelled **file**. If you use your mouse to click on the drawer gadget, it is activated and you can then type the path for the file you wish to either create or access. When you then hit the carriage return, the names of all the files already stored in that drawer will appear in the rectangular region above the drawer gadgets, and the file gadget will become active. If there are a sizeable number of files in this directory it will take a few moments for the names of all files to be loaded.

NOTE : The drawer or path specification you enter *must already exist*, as directories *cannot be created from the file requester.* A nonexistent path specification causes the words `Directory Error` to appear.

3.3.3 Scrolling Around

If the designated path points to a directory which contains a large number of files, not all of the filenames can be visible at once. You can view them all, however, by *scrolling* through the list of names. That is what the *slider* (also called the scroll bar) on the right hand side of the requester is for.

The scroll bar is a gadget which works like a rectangular knob which slides in a track. To use it, place your mouse over the rectangular knob, hold down the left mouse button (this grabs the knob) and slide your mouse up or down. As you do this the list of file names in the directory will scroll past. In addition, when you first click on the knob, the list of files will be sorted alphabetically.

3.3.4 Doing It All With The Mouse

It is not really necessary to type the name of a file into the file gadget; it is possible to use the mouse to accomplish this task. To take advantage of this shortcut, move your mouse pointer over the name you wish to appear in the file gadget and click your left mouse button once. The name will automatically appear in the file gadget below. Now, in order to indicate that you have made your choice, click on the OK

gadget or hit the return key. You can also click twice on the filename displayed in the space above the drawer and file gadgets.

You can also use the mouse in the same way to search devices and directories. For example, if the file you are dealing with has a very long full filename, such as

```
SYS:documents/essays/english/cartonessay.tex
```

you probably don't want to have to type the path for this file into the drawer gadget. In such cases, you will want to use the file requester to do the job for you by scrolling through the list of devices, directories, and subdirectories, and clicking on names as they come up. One way to do this quickly is to type the name of the device (such as SYS:) into the drawer gadget and hit return. A list will appear in the region above the drawer and file gadgets. This list will contain some filenames in addition to names that are preceded by the highlighted letters **(dir)**. Names prefaced by **(dir)** are names of directories or subdirectories. When you click on a name that starts with the highlighted letters **(dir)**, this directory specification will appear in the drawer gadget. At this point another list will appear, this one of files and subdirectories within the directory you selected. Once again, if you click on the desired subdirectory its name will be appended to what is already in the drawer gadget, and the contents of this directory will appear in the scrolling region. In this way you can build up the path for the file you want to load simply by clicking on each subdirectory as you find it. Finally, when the name of the file you are looking for has appeared in the scrolling region, click on it and that name will appear in the file gadget. At this point you will have completely specified a full-filename and so can click on the OKAY gadget in order to proceed.

3.3.5 The Drives Gadget

It is not even necessary for you to type the name of the device into the drawer gadget. It is possible to use the mouse to scroll through a list of device names in the same way as you scroll through directory and file names. To do this position the mouse cursor over the body of the requester and either click on the right mouse button, or click the left mouse button when the pointer is over the **Drives** gadget located at the bottom of the requester. When you do this the names of all the devices on which files are stored will appear. By clicking on the name of a device it will automatically appear in the drawer gadget and the scrolling area will be updated to display the files and directories which are located on that device.

3.3.6 The Parent Gadget

Although we have discussed the use of the file requester in reverse: that is, we started with scrolling through filenames, then went on to scrolling through directories and subdirectories, and then went to scrolling through device names, when

you actually use the file requester, you will probably go in the opposite order. Going through these various stages is like peeling away the layers of an onion until you find the file you are looking for. Sometimes, you will choose the wrong directory or subdirectory. Clicking on the **Parent** gadget—one of the rectangular gadgets at the bottom of the requester—will get back the list of files that lay outside this directory. First the directory name will revert to the name of the parent directory and then the list of filenames and subdirectories in the parent directory will be redisplayed.

3.3.7 The Cancel Gadget

The last thing left to discuss is the CANCEL gadget. If you wish to abort the whole process you just started, click on the cancel gadget and the requester will disappear without doing anything.

An important note: If you type in an incorrect path specification and TxEd returns the words `Directory Error`, the list of files disappears from the requester. Don't panic, however, as TxEd hasn't really crashed. Click in the drawer gadget and type in an existing directory; the file requester will recover.

ASIDE : The Amiga allows you to pull a trick and assign whole path names to a 'logical' device by typing, for example,

```
> assign REXX: SYS:REXX
```

inside a CLI or by putting it in your startup-sequence. If you assign a path name to a logical device, then the requester will display this 'device' along with the other device names when you click the right mouse button. If you use a specific directory a great deal, this trick can be very useful.

4. Welcome to TxEd

4.1 Starting TxEd?

TxEd, like Xedit on the IBM, is a text editor. A text editor is a program which allows you to write, edit, and save the text files which you will then format using TEX. As with most Amiga programs, TxEd can be started from either a CLI window, or from the WorkBench.

To start TxEd from a CLI you simply type

```
> e
```

where `e` is the name of the TxEd program. It should be located in your **usrc** directory. If you have defined `x` as an alias for `e`, then you can also launch the editor by typing

```
> x
```

This way of working is popular with those of us who edit files both on the IBM mainframe and on the Amiga, since we then don't have to worry where we are.

4.1.1 WorkBench Under AmigaDOS 1.3

To start TxEd from the WorkBench will be difficult if you are running AmigaDOS 1.3, since the **usrc** directory probably doesn't have a drawer icon, and there is a good chance that you do not have an icon for the TxEd program itself. The easiest way to remedy this is to copy the **usrc** drawer icon and TxEd icon from the installation disk to your hard disk. Before doing this determine the name of your hard disk, or the partition on your hard disk, wherein the **usrc** directory is located.

To determine the name of your hard disk, or partitions on your hard disk, begin by examining the disk icons on your WorkBench. You should see icons labelled **fast:**, or **work:**, or **dh0**, or something to that effect. Alternatively, you can get a list of the names of devices where the **usrc** directory may be located by typing

```
> assign vols
```

or even just

```
> assign
```

which will produce a list of names under the header `Vols:`. Using either method you can then procede to locate **usrc** by typing

```
> dir devicename:
```

for each name on your list and searching the output for the directory **usrc**. You should begin by trying the names **sys:**, **fast:** and **work:**, if they appear on your list.

Once you have located the device on which **usrc** is located, let us assume it is **fast:**, then you should put your update disk into drive **df0:** and type

```
> copy df0:usrc.info to fast:
```

When the copy is complete you should click on the disk icon labelled **work** an see that the **usrc** drawer now exists. If you do not have a TxEd icon, one is provided on the update disk. Decide upon where you want this icon to go and copy the file **TxEd.info** to that directory and also the executable **TxEd**. Most likely you will simply type

```
> copy df0:TxEd.info to fast:
> copy df0:TxEd to fast:
```

4.1.2 WorkBench Under AmigaDOS 2.0

If you are running AmigaDOS 2.0, then running TxEd from the WorkBench is much simpler, as the new operating system can make up default icons for all drawers and programs which don't have one. In this case, to start TxEd running all you have to do is open the **usrc** drawer and then double-click on the default TxEd icon, which will be visible if you use the *show all* option.

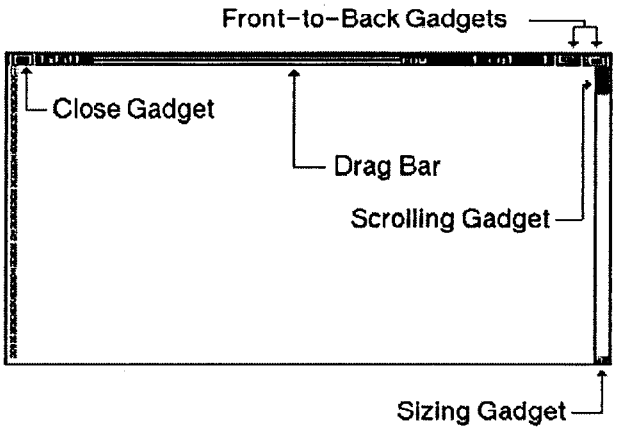
4.1.3 Running From The FastMenu

There is one more way to start TxEd running if you have standard SLAC configuration; that is to start TxEd from the fastmenu. The fastmenu is the small window full of gadgets which appears on the WorkBench screen when you turn your machine on. One of the gadgets in the fastmenu is labelled `Edit`. All you have to do is *single-click* on the `Edit` gadget and TxEd will start running.

4.2 TxEd Is Not XEDIT

Although both XEDIT and TxEd are text editors, TxEd's user interface is rather different from XEDIT's. First, as with almost all applications on the Amiga, TxEd does not usurp the entire Workbench screen; instead, it opens a resizable window. You can switch back and forth between the TxEd window and other Amiga windows at will by positioning your mouse pointer over the desired window and clicking the left mouse button. We will refer to the act of positioning the mouse pointer over a window and then clicking the left mouse button as *clicking on the window*. You can use the various gadgets included in the window borders, to resize the TxEd window and move it to the front or back of other windows. The

general layout of a TxEd window, together with its border gadgets is explained below.



Close Gadget — The close gadget, which is found at the upper left-hand corner of the window, is used to close the window. Click on the close gadget with your left mouse button, and the window will disappear. If you haven't saved the most recent changes to the file, a requester will appear, asking `Exit without saving changes?` If you click on the OK gadget, the window will still disappear, but if you click on the Cancel gadget, the window won't close.

Front-To-Back Gadgets — These two gadgets at the upper right-hand corner of the window are used for shoving a window behind other windows and for bringing a window to the front of other windows. The gadget on the left puts a window to the back of all the other windows while the gadget on the right brings a window out in front of all the others. Just single-click on either gadget with your left mouse button to make it work.

NOTE : If you are using an Amiga 3000, which runs the new operating system, the two gadgets at the top-righthand side of the window are different. The rightmost gadget is a front/back gadget. Clicking on this gadget once brings the window in front of all other windows, clicking on it a second time will push it behind all other windows. If you wish to use one click to push a window to the back, then hold down the Shift key while you click on the front/back gadget. The gadget to the left of the front/back gadget is a *zoom* gadget. When you click on this gadget the window will shrink in size. The functioning of the various window gadgets are well explained in the manual which comes with the 3000.

Scroll Bar — The left border of the window, as you'll notice, is made of two parallel lines. The lines represent slider tracks. Within these tracks is a rectangle—the slider, or scroll bar. After grabbing the scroll bar with your mouse (you do this by clicking on the slider with your left mouse button and continuing to hold the left mouse button down) you can slide the scroll bar up and down the tracks,

thus scrolling the contents of the window up or down. Some windows also have a horizontal scroll bar on their bottom border that lets you scroll the window's contents left or right.

Drag Bar/Title Bar — The bar that forms the top bar of the window is often known as the title bar, as it displays a line of text identifying the window. Flanked by the close and front-to-back gadgets, the title bar displays more than just the program's title. It displays the name of the file being edited and your cursor position. Notice, to the immediate left of the front-to-back gadgets, a space which says `row 1 col 1`. This is your cursor position—the first row, or line of text, and the first column. Each column is a character wide and the columns are numbered from left to right. Sometimes, when you issue a command, the filename will be replaced by a response, such as `Not Found`, or `Saving`, or `Done`. Furthermore, the striped middle segment of the title bar indicates the status of the window: fuzzy stripes indicate the window is inactive and crisp, clear stripes indicate that the window is active.* Note that under AmigaDOS 2.0 the title bar does not have the horizontal stripes and a window's status is indicated by changing the color of its borders.

The title bar is also known as the drag bar, because grabbing the bar with your mouse (see **Scroll Bar**) allows you to drag the entire window to a new location on the screen.

Sizing Gadget — The gizmo in the lower right-hand corner of the window is the sizing gadget. To employ this gadget, click on it with your left mouse button and hold the left mouse button down. While you are still holding down the left mouse button, you resize the window by moving your mouse. Moving your mouse down enlarges the window vertically, moving your mouse up shrinks the window vertically, moving your mouse to the right enlarges the window horizontally, and moving your mouse to the left shrinks the window horizontally. Moving your mouse on a diagonal towards the bottom right-hand corner of the screen enlarges the window in both directions, and moving your mouse on a diagonal towards the upper left-hand corner of the screen shrinks the window in both directions.

Menus and Menu Bars—Usually, you issue commands in TxEd by selecting these commands from the command menus. These menus drop down from the menu bar. If you hold down the right mouse button while the TxEd window is active, the menu bar will appear at the top of the screen. The menu bar displays the title of each of the window's menus. If you move your mouse (while still holding down the right mouse button) up to the menu bar and position your cursor over

* Only one window is active at any one time. When a window is active, that means that you can access its menus and give it keyboard input. The program controlling the window, however, is active at all times, even when its window is inactive.

one of the menu titles, the menu will drop down.

To select a menu option, position your cursor so that the desired menu option is highlighted. Then release the right mouse button. A keyboard sequence may be shown next to some of the menu options. Typing such a keyboard sequence has the same effect as selecting the associated menu option.

4.3 Opening A Specific File

When we started this chapter we said that to start the editor you simply type **e** or **x** in a CLI. If you do this, then the editor will open a window, but the window will be empty. This is the analogue, if you will, of reaching for a blank piece of paper and inserting it in your typewriter. At this point you are ready to begin typing. When you wish to save the results of your labor you will have to tell TxEd where to save the file and under what name. We will discuss how this is done when we discuss the **Project** menu.

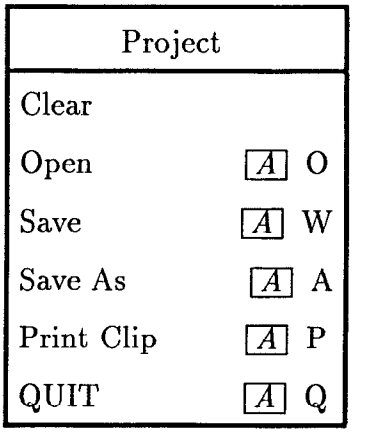
Often, we do not wish to begin with a blank sheet of paper. Rather, we wish to continue working on a file created some time ago. Of course, this can always be done by starting TxEd by typing **e** and then using the **Open** option in the **Project** menu to *load* the desired file. We will explain this process in a moment. There is, however, a way to have the editor open with a specific file already loaded. To do this go into a CLI and type

```
> x <device>:<directory>/<subdirectory><...>/filename
```

When the TxEd window opens the indicated file will be will be displayed.

4.4 TxEd Menus

4.4.1 The Project Menu



Clear This option completely clears the editor of all text. It does not erase the file on which you are working *unless* you save the blank screen. Do be careful, however, since the the **Clear** option does not alter TxEd's default filename when using the **Save As** option. In other words, if you clear the editor, type in a new file, and save that file using the **Save As** option (see below), TxEd will use the name of the previous file as the default entry in the file requester. Be sure to change this name before clicking on OK or you will overwrite the file which was originally in the editor. If you simply use the **Save** option then you will be told that the file doesn't have a name.

Open This option brings up the file requester. Type in the path name and file

name of the file you wish the editor to load. Then click on the **OK** gadget. The file will be loaded, unless it does not exist, in which case TxEd will bring up a message window informing you that it cannot find the directory or file you have specified. This message window has two gadgets, **Retry** and **Cancel**. Click on the **Cancel** gadget and verify the spelling of the file and path specifications you typed into the requester. Typos will make it impossible for TxEd to find the file.

Save This option *does not* bring up the file requester. It saves the file to the currently specified filename. If you aren't sure what the currently specified filename is, don't use this option, as you could end up overwriting a file you wanted to keep. This option can also be called using the keyboard sequence [A] W.

Save As When you select this option, the file requester appears. Type in the path name and file name to which you wish the file currently in the editor to be saved.

NOTE: *YOU CANNOT CREATE DIRECTORIES FROM THE FILE REQUESTER!* If you type in a nonexistent directory as part of the path name, TxEd will return an error message stating `Directory Error`. See Chapter 3 for advice on how to make a new directory.

Print Clip If you select this option, the file requester comes up, bearing the legend `SAVE/PRINT CLIPBOARD`. The **Drawer** gadget will be blank and the **File** gadget will contain the default device `Prt:`. What's the Clipboard? Well, it's the place to which text you delete or copy using the **Edit** menu options is saved. When you select the **Print Clip** option, the clipboard is, by default, sent to the printer device and printed. If you change the device shown in the **File** gadget before hitting return, then the clipboard will be saved to somewhere else.

QUIT This option closes the editor window and causes you to quit out of the program.

4.4.2 A Warning About Saving Files

It is important to understand the nature of the devices to which you save files. Devices which have names which begin with **DF**, such as **DF1:**, **DF2:** , ..., are floppy disk drives. On an Amiga 2000 or Amiga 2500 the names **DF0:** and **DF1:** are reserved for the two floppy disk drives located on the front of the computer. **DF0:** is the drive located at the extreme right of the box, and **DF1:** (if you have two drives) is located to its left. When you save a file to such a device it is written onto the floppy disk in that drive. Such storage is permanent; unless you overwrite the file, erase it, or destroy the floppy disk the file will be preserved forever. Names which begin with **DH**, such as **DH0:**, or **DH1:**, or names like **slow:**, **fast:**, **work:**, usually refer to hard disk drives. These are usually inside your machine. Hard disk drives are also permanent storage devices, in that files

saved on your hard disk survive until they are erased, or modified. In particular, files stored on your hard disk remain where they are, even when you turn off your Amiga. There is, however, another class of devices on which files can be stored, which does not provide permanent storage. These are the devices named **RAM:**, or **RAD:**, or **VD0:**.

RAM:, **RAD:** and **VD0:** are what are called virtual, or ram-disk, devices. These devices are really just regions of computer memory which are partitioned off by the operating system to be used for temporary storage of files. Information stored in these regions *disappears* when the power is turned off. This is why ram-disk devices are referred to as volatile storage, as opposed to permanent storage. Accessing a file stored in **RAM:**, or **RAD:**, or **VD0:** is much faster than accessing file stored on a floppy, or even a hard disk. For this reason, it is sometimes convenient to copy a file to one of these devices and work on the copy. HOWEVER, ALWAYS BE SURE TO SAVE A FILE CREATED ON A RAM-DISK DEVICE to a floppy or hard disk before shutting down your machine. Failure to do so will result in the loss of the current version of the file.

There are important differences between the **RAM:** device, and the devices called **RAD:** or **VD0:**. Even though all three devices are volatile storage devices, and disappear when the power is turned off, they do not have the same characteristics. The **RAM:** device does not have a fixed size, it will expand to fill as much of your computer's memory as is available. This is the good news! The bad news is that the contents of the **RAM:** device not only disappear if you shut off the power, but they also disappear if you have to reboot your Amiga with the power left on. Such a reboot is referred to as a *warm reboot* and is accomplished by holding down the Left-Amiga, Right-Amiga and Ctrl keys at the same time. Since no operating system is immune to problems caused by errant software, sometimes a warm reboot is necessary in order to reset the machine and shut off programs which are corrupting memory, or which refuse to run properly. When this happens, however, any files stored on the **RAM:** disk will be lost. For this reason, we strongly recommend that you store temporary editor files on the virtual devices called **RAD:** or **VD0:**, depending upon your machine. These virtual storage devices use a fixed amount of memory, so they are limited in what they can store, but the contents of either of these devices *almost always* survive a warm reboot. Almost always, because a particularly egregious program can corrupt the contents of even these devices and destroy their contents.

4.4.3 In Case Of Catastrophe

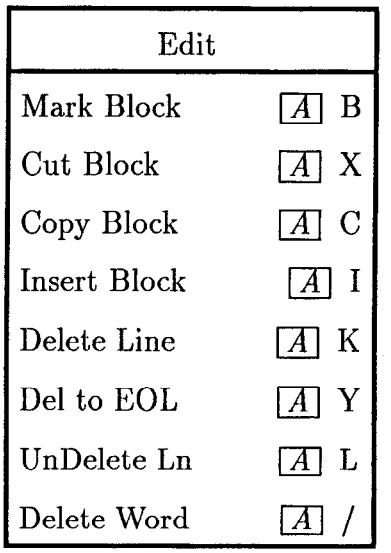
While we are talking about how one can lose a file, we should spend a few moments on the question of how TxEd saves *backups*. Each time you save the current changes

to a file, or TEX a file by hitting **F1**, TxEd saves a backup version of the file to the logical device **T:**. We refer to **T:** as a logical device, because it is not a separate physical device like **DF0:**, or **fast:**, but it is a directory on some physical storage device which has been *assigned* the name **T:**. To locate a backup file in the **T:** directory just type

```
> dir t:
```

and to load such a file into your editor just type **t:** into the drawer gadget in the file requester, or click on the **DRIVES** gadget and then scroll around until you find **T:**. The backup files will be saved with the same name, but with numbers attached to them. The highest number will, most likely, indicate the most recent backup.

4.4.4 The Edit Menu



Mark Block Selecting the **Mark Block** option puts your cursor into block-marking mode. The next time you click with your mouse, all the text between your original cursor position and the point where you clicked will be highlighted. The highlighted text constitutes a block, which can be operated on using the **Cut Block**, **Copy Block**, and **Insert Block** options (*see later*). Double-clicking with your mouse also puts your cursor into block-marking mode.

Cut Block If you have marked a block, the **Cut Block** option will remove the marked block of text from the editor while simultaneously saving the text block to the Clipboard. You can then retrieve this text block using the **Insert Block** option. If you have not marked a block, then TxEd will return an error message stating that a block has not been marked.

Copy Block If you have marked a block of text, the **Copy Block** option will save the block to the Clipboard without removing the block of text from the editor. You can then insert this copied block of text wherever you wish using the **Insert Block** option.

Insert Block Selecting this option allows you to insert the most recently cut or copied block at your present cursor position. To use this option, cut or copy a block using the **Cut Block** or **Copy Block** options, then position your cursor in front of the spot where you wish the block to be inserted. Then select the **Insert Block** option and voilà!

Delete Line Selecting the **Delete Line** option causes the line of text indicated

by your current cursor position to be deleted. Your cursor does not have to be at the front of a line for this option to work effectively.

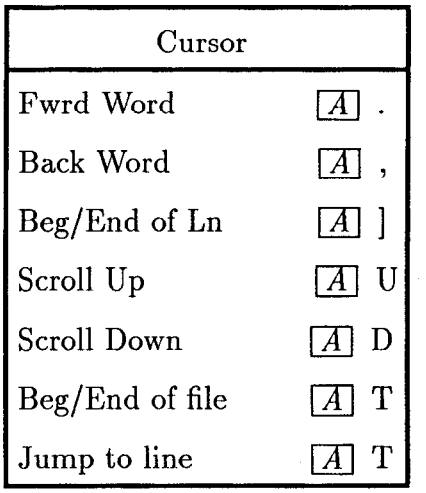
Del to EOL This option deletes all characters in a line that come *after* your cursor. If your cursor is at the front of a line, this option will delete the entire line.

UnDelete Line The **UnDelete Line** option re-inserts the last line deleted using the **Delete Line** option, or the last portion of a line deleted using the **Del to EOL** option. The line will be re-inserted at your current cursor position.

TxEd "remembers" the last line that you deleted until you *delete* a different line. As a result, the **UnDelete Line** provides a terrific way to make multiple copies of a line, as you can delete a line and then undelete it repeatedly.

Delete Word The **Delete Word** option deletes the word just following your current cursor position.

4.4.5 The Cursor Menu



Fwrd Word The **Fwrd Word** menu option, when selected, moves your cursor forward one word.

Back Word The **Back Word** menu option, when selected, moves your cursor backwards one word.

Beg/End of Ln This option, when selected, moves your cursor to the beginning or end of the line of text where it is currently positioned. Unless your cursor is already positioned at the beginning of the line, this option will move your cursor to the beginning of the line. If your cursor is already positioned at the beginning of the line, then selectingt this option will move your cursor to the end of the line.

Scroll Up Selecting this option causes the text in the editor to scroll upwards, that is, towards the beginning of the file.

Scroll Down Selecting this option causes the text in the ditor to scroll downwards, that is, towards the end of the file.

Beg/End of file Selecting this option causes your cursor to jump either to the beginning or end of the file. The text scrolls along with the cursor. If your cursor is already positioned at the beginning of the file, your cursor will jump to the end of the file. Otherwise, selecting this option will cause your cursor to jump to the beginning of the file.

Jump to line Selecting the **Jump to line** option causes the title space of the title bar to go blank and the word `Jump:` to appear instead. Type the number of the line to which you want your cursor to jump. Then hit return. Your cursor will jump to the specified line and the text will scroll accordingly. If you make a mistake while typing in the line number, use the backspace key to put the number back to zero and type the line number over again.

4.4.6 The Search Menu

Search Selecting the **Search** option causes a string requester to appear. Type in the word you want to search for into the space provided and then hit return. TxEd will move your cursor to the first incidence of the string *following* your present cursor position. If the string does not appear in the file, TxEd will post the message `Not Found` in the title space of the title bar. If you wish to continue searching for repeated incidences of the string, select the **Repeat** option or type [A] G.

Search	
Search	[A] S
Replace	[A] R
Repeat S/R	[A] G

If you want your search to be case-sensitive, that is, for TxEd to differentiate between capital and lower-case letters, then click on the small gadget in the left-hand corner of the requester that says `UC == lc`. The box will become highlighted and the text will now read `UC !== lc`. Furthermore, if you want TxEd to search for incidences of a string *before* your present cursor position, click on the small gadget in the right hand corner of the string requester. The text should change from `Forward` to `Backward`.

Replace The **Replace** option allows you to search for repeated occurrences of a particular string and to automatically replace that string with another string. For instance, if you want to change, in a particular file, all occurrences of the name Anna to the name Gwendolyn, you use the **Replace** option.

Selecting the **Replace** option causes a string requester similar to the **Search** string requester to appear. Enter the string that you wish to replace in the first string gadget. Then either hit return or click on the second string gadget (the one labeled `Replacement String` and enter the string with which you wish to replace the first string. When you have correctly entered your search and replacement strings, hit return. If the search string appears in the file, the following words will appear in title space of the title bar:

```
Replace?  (Y)es (N)o (G)lobal (Q)uit
```

Typing `Y` causes TxEd to replace the first occurrence of the string with the replacement string and move on to the next incidence of the search string. If there are no more incidences of the search string, TxEd will display the word `Done`. Otherwise,

the same set of options (Y, N, G, Q) will be displayed. Typing `N` causes TxEd to skip the current incidence of the search string and move on to the next one. Typing `G` causes TxEd to replace all occurrences of the search string with the replacement string. Typing `Q` causes TxEd to abort the search/replace operation.

Note: If you have specified `Forward` *on the string requester, TxEd will find only those incidences of the search string that occur after your current cursor position. If you have specified* `Backward`*, TxEd will find only those incidences of the search string that occur before your current cursor position. To be certain, therefore, that TxEd finds all occurrences of the search string, your cursor should be at the beginning or end of the file, depending on which option you selected.*

Repeat S/R This option causes your last **Search** or **Replace** operation to be repeated.

4.4.7 The Random Menu

Most of the items in the Random Menu are documented in the TxEd Manual. As these **Random** menu options aren't crucial, we'll dispense with a detailed explanation here. There are some **Random** menu options which were added via ARexx. These options are important and are listed below.

```
Send this to TeX [F1]
Send chunk to TeX [F4]
Print this as TeX
Print page range
Print chunk as TeX
VLT to front  [F8]
```

Send this to TeX causes the file currently displayed in the editor to be saved and sent to TeX . **Send chunk to TeX** allows you, if you have marked a block of text within your editor file, to send it through TeX by itself. **Print this as TeX** allows you to print the file you just TeX-ed. **Print page range** also prints the most recently TeX-ed file, but first it brings up a requester. You enter, separated by a space, the page at which you wish TeX to start printing your document and number of pages that you wish TeX to print. This option lets you print a small segment of a large document. **Print chunk as TeX** allows you to print a block of text that you have TeX-ed using the **Send chunk to TeX** option. *WARNING: The printing options work only if you have a printer attached to your Amiga.* Selecting **VLT to front** brings the terminal program, VLT, in front of the Workbench screen. The keyboard equivalents of these options, if they exist, are listed in the menu itself.

4.4.8 The User Menu

The options in this menu are fairly self-explanatory. They allow TxEd to behave, to a large extent, as if it was a word processor. There are menu options for starting and ending italics, boldface, and typewriter fonts, menu options for causing a paragraph to become narrower, etc.. When you select one of these options, the proper TEX command appears in the editor at your current cursor position.

5. TEX-ing and Previewing a File

5.1 Starting up TEX

Starting up TEX on the Amiga is a piece of cake. Go to the **User** menu of TxEd and select `Send this to TEX` , or hit the `F1` key. Your file, as it currently appears in the editor, will automatically be saved. Then two new windows will start up: the previewer and the error message window. The error message window will appear as a tiny window in the lower right-hand corner of the screen (you may enlarge it if you wish), while the previewer will open up in front of the TxEd window.

If all goes well and you have no errors in your file, then a beautiful facsimile of the TEX -formatted file will appear in the previewer. Since, however, TEX errors are a fact of life, we will begin by discussing what happens when TEX hits an error.

5.2 TEX Errors

When TEX encounters an error in your file several things happen at almost the same time. First, the TxEd window is brought in front of the previewer window and an error message appears in the error message window down in the corner. Soon thereafter another, smaller, TxEd window opens in order to display the TEX log file. At the same time the TxEd window is modified so that the cursor is positioned at the place in the file where TEX thinks the error is located. When you have located the error in your file, follow the posted instructions and click on the close gadget of the small message window in the upper left of the screen to make both that window and the log file disappear. At this point you should make your corrections and send the file to TEX again by hitting **F1**.

The error-checking frills are the result of ARexx execs. AmigaTEX itself responds to errors in the same fashion as TEX on the IBM. If you enlarged your error message window, you would see the same error messages, etc. that IBM TEX types to the screen when it encounters a difficulty. You can give the error message window the same limited kinds of input that TEX on the IBM will accept; for instance, if AmigaTEX can't find an input file, you can type the name of a different input file into the error message window. AmigaTEX will then input the new file and go on TEX-ing.

5.3 When You Are Finished

There are two ways to quit TxEd when you are finished TEX'ing. The first way is to simply move your mouse over the close gadget in the upper left-hand corner of the TxEd window and click on the close gadget. In this event the editor will close, but both AmigaTEX and the previewer will keep running. To stop these programs you will have to close each of them separately by clicking on their close gadgets.

There is a better way to close everything; that is to select the `QUIT` option from the `PROJECT` menu, or its keyboard equivalent [A] `Q`. If you do this, then the editor will close and at the same time it will tell the previewer and AmigaTEX to go away, at least if it is the only TxEd window which is communicating with AmigaTEX. Given the current setup, you can have several TxEd windows sending different files to AmigaTEX for processing and previewing. In the event that you have more than one window open and communicating with AmigaTEX you do not want to have AmigaTEX and preview closing down on you when you quit only one of the files. To handle this situation the last TxEd you quit from will close down AmigaTEX and Preview. In this way, the combination of TxEd, AmigaTEX and Preview, appear to function as a single program.

It is possible, however, that this cooperative behavior can fail to work properly. Suppose, for some reason, you quit one of the editor windows using the close gadget, and not the `QUIT` menu-option. In that case the mechanism for keeping track of who the last TxEd is gets all messed up. An ARexx exec has been provided to get everything back on track. If you find that the automatic closing down mechanism is not working properly, go to a CLI and type

```
> cleartexmasters
```

Once you have done that then things should procede normally.

5.4 Another Mode of Operation

People used to using TEX on the IBM have often gotten into the habit of scrolling past errors and then, using the log file, doing wholesale corrections of their TEX errors. The current ARexx macros do not support this mode of operation as the default setup. Using the default system, TEX stops running when an error is encountered, and you are thrown back into the editor. If, however, you wish to be asked what TEX should do when it encounters an error, there is a way to make this happen; go to a CLI and type

```
> doaskme
```

To go back to the default mode of operation you need only type

```
> dontaskme
```

5.5 The Previewer Window

When you have weeded all the TEX errors out of your file and everything runs smoothly, a facsimile of the formatted file will appear in the previewer. The previewer is a time-saving addition to TEX , because it allows you to view the formatted, TEX-ed version of your file *before* printing. Like the editor, the previewer opens its own which looks quite a bit different from the editor window. For a start, it has several new gadgets.

5.5.1 The Arrows

You'll notice, in the right hand area of the previewer's title bar, two gadgets which display arrows: a right-arrow and left-arrow, to be precise. These arrows allow you to move through the various pages of the TEX document you've just TEX-ed. Clicking on the right-arrow causes the previewer to display the next page of the document, while clicking on the left-arrow causes the previewer to display the previous page of the document. Striking the right- and left-arrow keys on your keyboard will have the same effect.

5.5.2 The Scroll Bars

The scroll bar that constitutes the right-hand border of the previewer window allows you to move up and down the page being displayed. The up- and down-arrow keys on your keyboard can also be used for this purpose. The scroll bar that constitutes the bottom border of the previewer window allows you to move right and left on the page being displayed. The right-left scroll bar is useful if you are previewing a document with oversized pages, or if you are in Zoom mode (*see below*).

5.5.3 The Zoom/Unzoom Gadget

To the immediate left of the arrow gadgets appears a gadget displaying two small rectangles. This is the Zoom/Unzoom gadget. Clicking on this gadget causes the text to become magnified, or zoomed. Clicking on this gadget again causes the text to be restored to normal size.

5.5.4 The Page Gadget

On the left-hand side of the title bar, to the right of the title space, where the document's filename is displayed, is a small space which shows the number of the page currently displayed in the previewer. If you click on this space, delete the number displayed there, type a new page number, and hit return, the previewer will display the page whose number you typed. If you type a page number which does not exist for the currently displayed document, the previewer will flash and display the last page in the document.

5.6 Previewer Menus

The previewer window has several menus of its own. As these menus are documented in the AmigaTEX manual, we won't take time here for a comprehensive tour. We will however indicate the sort of options which can be found in each of these menus, and hit upon a couple of special menu options which you should be aware of.

5.6.1 Project Menu

This is the first menu which appears on the far left of the menu bar when the previewer window is active and you hold down the right mouse button. One of the options in this menu allows you to modify the colors used in the preview window to display the text. If you make a change in this, and any other option, and wish to save it permanently, then select the `Save Configuration` option. The `Free Fonts` and `Free Picts` options are to be used if you get into a low memory situation and wish to recover some of the memory that the previewer is using to store previously displayed text and pictures. Choosing these options does not destroy your ability to preview previously displayed pages, all it does is slow down the rate at which they will come to the screen.

5.6.2 Unzoom and Zoom Menus

We already noted that the AmigaTEX previewer can be toggled between two display modes by clicking on the zoom gadget, which appears to the left of the arrow gadgets. The magnifications associated with each of these modes can be reset independently by selecting the appropriate options in the `Unzoom` and `Zoom` menus. Both the zoomed and unzoomed display support fourteen legible (or almost legible) magnifications, and a full-page display mode. If your Amiga does not have the necessary fonts installed to display text at a particular magnification, then a series of blocks of the appropriate size will be displayed instead. The full-page option causes a schematic presentation of the entire page to be displayed. Most of the time you will not be able to read the text in such a display, since you will not have fonts which are small enough, however this mode is very useful for looking at overall layout. It is also useful when making transparencies using **slidepack.tex**, because in that case you can read the full-page display.

5.6.3 Borders Menu

This menu contains several options which allow you to change how text is displayed in the previewer window. You can play with them and see how they change things. One option, however, is particularly interesting; that is, the `Dual Page` option. This option tells the previewer to display two pages at a time, side-by-side. Com-

bining the dual-page option with the full-page mode of display allows you to get a better picture of how a final printed document will look.

5.6.4 Display Menu

The options in this menu control whether or not the previewer opens on the Work-Bench screen, or on a custom screen. You should see that the `Interlace` option has a check-mark in front of it. Other options control the way in which included graphics will be displayed. No other items should have checks in front of them if you wish to run in the standard SLAC configuration.

5.6.5 Tracking Menu

This menu controls what the previewer is doing with respect to TEX. If the `Independent` option has a check in front of it then the previewer is not cooperating with TEX and files will not be automatically displayed. If, by some accident, you have saved your previewer configuration with the `Independent` option selected, then when you TEX a file from the editor you will be warned that the previewer is open but is not in condition to accept a file. You should then go to the previewer window and select the `Independent` option again, so as to make the check mark disappear. After you have done that save the configuration. The options which should be check marked by default are: `Track File`, `Auto Activate` and `Auto Front`. When these are selected the previewer will automatically jump in front of all other windows when it is ready to display the first page of a new file. Also, it will be the active window, so you do not have to click on it to move around in a file. In the `Track File` configuration you must use the mouse, or arrow keys, to display subsequent pages of the file. If you wish each page to be automatically displayed as it is processed, select the `Track Page` option.

6. The FastMenu PHYZOptions

The fastmenu is a small rectangular window that appears on your screen when you boot up. It has the customary window gadgets—a drag bar, a close gadget, and front-to-back gadgets. It does not have a sizing gadget, but it does have menus. The fastmenu looks a good deal like a list, with each name surrounded by a thin box. As a matter of fact, each boxed name is a gadget. Clicking once with the left mouse button on one of these gadgets starts up an ARexx exec.

The three gadgets which have the names **PHYZMemo**, **PHYZLett**, and **PHYZPub**, call special ARexx macros which allow you to interactively format a memo, letter, or paper using your mouse, instead of typing in all the commands by hand. Below is a brief description of these options and how they work.

6.1 PHYZMemo

Clicking on the **PHYZ-Memo** gadget will open a screen which allows you to fill in the blanks in order to create a TEX memo file. When the screen comes up the Date gadget will be active and will show the default value, Automatic. If you delete the default value and replace it with a different piece of text, or a string of numbers, then the replacement string will be included in the memo as the date. Make whatever changes you wish in the Date line, then either hit return to go the next line or use your mouse to jump directly to whatever line you wish. The other gadgets on this screen are described below.

- `To:` The text entered in this gadget will be included in the memo; the memo will use this text to indicate the recipient of the memo.
- `From:` The text entered in this gadget will identify the memo's sender.
- `Topic:` Enter a summary of the memo's topic in the set of gadgets which come under the `Topic` heading.
- `Copies:` Enter either Yes or No. If you enter yes, the command `\copies` will appear in your editor file. Follow the command by a list of people to whom you wish copies to be sent. For the list to have only one name to a line, follow each name by `\nextline`.
- `Filename:` Enter in this gadget the name of the TxEd file you want the exec to create. The default filename, which appears in the gadget, is vd0:memo.tex.

When the information in the template is complete, click on the **Continue** gadget. The screen will close and the editor will open. It will contain a TEX file with all of the correct commands for a memo. Move your cursor to the indicated position and type in the body of the memo.

If you have entered the name of a file which already exists in the `Filename` gadget then, when you click on `Continue`, a requester will appear. This requester will ask if you really want to overwrite the existing file. If you click on the **Replace** gadget, the existing file will be overwritten. If you click on the **Redo** gadget, the PHYZMemo screen will remain active. At this point you should change the name of the file which appears in the `Filename` gadget and click on continue once again.

To ABORT without saving a memo click on the close gadget in the upper left hand corner.

6.2 PHYZLett

Clicking on the PHYZLett gadget of the fastmenu will open a screen which allows you to fill in the blanks in order to create a TEX letter file. When the screen comes up the Bin gadget will be active. If you don't have a bin number, or don't want your bin number included in the letter, leave that gadget blank. Make whatever changes you wish in the Bin line, then either hit return to go the next line or use your mouse to jump directly to whatever line you wish.

Below the Bin gadget is the Date gadget, which will contain the default value of `Automatic`. This means that the computer will enter the date indicated by its internal calendar. If you delete the default Date value and replace it with a different piece of text, or a string of numbers, then the replacement string will be included in the memo as the date.

There are a set of gadgets which allow you to enter the name and address of the addressee. There is a `Salutation` gadget, in which you enter the desired salutation, such as "Dear Ellie" or "Cher Georges." There is a `Close` gadget, in which you enter your closing remark, such as "Sincerely yours," and a `Signature` gadget for your name as it will appear, typed, below your signature. (The exec leaves a space for your handwritten signature between the closing and the typed name.)

The `Copies` gadget allows for two possible options, yes or no. If you enter yes, the command `\copies` will appear in your editor file. Once in the editor follow the `\copies` command by a list of people to whom you wish copies to be sent. For the list to have only one name to a line, follow each name by `\nextline`. Below the `Copies` gadget is the `Filename` gadget, in which you enter the name of the TxEd file you want the exec to create. The default filename, which appears in the gadget, is vd0:letter.tex.

When the information in the template is complete, click on the CONTINUE gadget. The screen will close and the editor will open. It will contain a TEX file with all of the correct commands for a letter. Move your cursor to the indicated position and type in the body of the letter.

If you have entered the name of a file which already exists in the `Filename` gadget then, when you click on `Continue`, a requester will appear. This requester will ask if you really want to overwrite the existing file. If you click on the **Replace** gadget, the existing file will be overwritten. If you click on the **Redo** gadget, the PHYZLett screen will remain active. At this point you should change the name of the file which appears in the `Filename` gadget and click on continue once again.

To ABORT without saving a memo, click on the close gadget in the upper left hand corner.

6.3 PHYZPub

Clicking on the PHYZPub gadget of the fastmenu will open a screen which allows you to fill in the blanks in order to create a TEX file with the correct commands for formatting a paper. When the screen comes up the Date gadget will be active and will show the default value, Automatic. If you delete the default value and replace it with a different piece of text, or a string of number, then the replacement string will be included in the memo as the date. Make whatever changes you wish in the Date line, then either hit return to go the next line or use your mouse to jump directly to whatever line you wish.

There are a set of gadgets in the upper left hand corner of the window which have box gadgets to the left of the string gadgets. These gadgets are used to set spacing, etc. and are self-explanatory. Click on the box to the left of the string gadget to cycle through the possible options for that gadget.

To add multiple authors to the paper, fill in a name in the Author gadget and click on the Add Author gadget. This will clear the text from the Author gadget and position you for the next author. Carry out this procedure once for each author.

As usual, you type the name of the file that you want the exec to write into the `Filename` gadget. Note that the `Filename` gadget on this screen isn't indicated as such. It is the gadget at the very bottom of the screen and contains the default filename `vd0:paper.tex`.

When the information in the template is complete, click on the Write File gadget. The screen will close and the editor will open. It will contain a TEX file with all of the correct commands for a paper. Move your cursor to the indicated position and type in the body of the paper.

If you have entered the name of a file which already exists in the `Filename` gadget then, when you click on `Write File`, a requester will appear. This requester will ask if you really want to overwrite the existing file. If you click on the **Replace** gadget, the existing file will be overwritten. If you click on the **Redo** gadget, the PHYZPub screen will remain active. At this point you should change the name of the file which appears in the `Filename` gadget and click on continue once again.

To ABORT without saving the paper, click on the close gadget in the upper left hand corner.

7. Where To Find Things

7.1 TEX Environment Variables

There are many ways in which the user can change the actions which AmigaTEX takes by default. One of the ways to do this is to change one of AmigaTEX's environment variables. To get a complete list of all of the environment variables used by AmigaTEX read Tom Rokicki's manual. There are, however, two environment variables which the Amiga user should know about.

The first such variable, `texformat`, controls which macropackage AmigaTEX will load when it starts up; the standard setting is

```
> texformat=phyzzx
```

If you wish to change this to plain.fmt, or lplain.fmt (for latex), then you issue a command of the form

```
> set texformat=lplain
```

if you are running one of the early releases of AmigaTEX, or

```
> setenv texformat lplain
```

note that there is no = sign. Under AmigaDOS 2.0 you should follow it with

```
> copy env:texformat to envarc:
```

if you want the setting to be permanent. If you want to permanently change this default for AmigaDOS 1.3, then you must edit the file **env:texdefaults**

The second important environment variable, `texinputs`, controls where AmigaTEX looks for files. A typical setting is

```
> texinputs=.,rad:,ram:,tex:inputs,tex:samples,tex:personal
```

i.e., a string of devices and directories followed by commas. This line is pretty self-explanatory, except for the period at the beginning, which tells AmigaTEX to first look in the current directory. To change the search order, or to add directories to this search order, you type either

```
> set texinputs=.,dir1,dir2,dir3,...,dirn
```

or

```
> setenv texinputs .,dir1,dir2,dir3,...,dirn
> copy env:texinputs to envarc:
```

Once again, to make a permanent change under AmigaDOS 1.3 you must edit the file **env:texdefaults**.

7.2 The TEX: Directory

In discussing the environment variable `texinputs` we referred to directories like **tex:inputs**. Usually **tex:** is assigned to a directory on your hard disk (or floppy disk, if you have no hard disk). The **tex:** directory has several subdirectories which you should know about.

- **tex:inputs** This directory contains plain.tex, as well as the usual PHYZZX files phyzzx.tex, myphyx.tex, newphyz.tex, phyzzx.fonts, phyzzx.local, and nul.tex, which also exist on the mainframe. Of course, it also contains the files tables.tex and slidepack.tex. In addition there is a file called amiga.graphics. This file is needed if you want to preview included graphics files on your Amiga. For a longer explanation of how this works read the file PICTURES TEX which exists on the U-disk on the IBM mainframe.
- **tex:macros** This is the directory which contains the **.fmt** files used by AmigaTEX. You should find that it contains **phyzzx.fmt** and **plain.fmt**. There is also a version of LaTEX available and a file **lplain.fmt** can be obtained from Marvin.
- **tex:pk** and **tex:fonts** These two directories contain information related to TEX fonts.

Feel free to make other subdirectories on the **tex:** device in which to store your personal files.

7.3 The REXX Directory

Reprogramming TxEd's menus, tying TxEd together with AmigaTEX and Preview, and creating the menus PHYZMemo, PHYZLett and PHYZPub, are all done with Amiga REXX (ARexx). The average user has no reason to worry either about where these execs are located, or what they look like. Some of you, however, suffer from an insatiable curiosity, and are only happy if you know how everything is done. This write-up is not the place for an extensive discussion of these matters; more complete documentation of this aspect of SLAC's Amiga setup has yet to be written. In this chapter we will simply tell the interested reader which files control which aspects of the setup, and where these files are located.

Before plunging into specifics we can say that *all* ARexx execs are stored in a special directory found on one of your permanent storage devices. Since the names of the hard-disks on different machines can differ, you should refer to this directory using its assigned device name, **rexx:**

7.3.1 TxEd's Menus

As with XEDIT on the mainframe, TxEd has the analogue of a profile exec which is automatically executed everytime TxEd starts up. This file is called

`rexx:startup.txed`

and a typical file looks like

```
/**rexx:startup.txed
*
**/
SIGNAL ON ERROR
OPTIONS results
ARChive 7
COLor 0
WRap 1
TAB 3
MENU CLEAR 5 0
MENU CLEAR 0 5
RM 71

menu 0 'Q' '"Quit "' "TxEdStuff/DoQuit"
menu 5 ' ' '"Begin numbered points"' ,
    "'INSERT ""\pointbegin"";""txedstuff/newline""'"
menu 5 ' ' '"Next point          "' ,
   "'INSERT ""\point"";""txedstuff/newline""'"
menu 5 ' ' '"Begin quotes        "' "'INSERT ""``""'"
menu 5 ' ' '"End quotes          "' "'INSERT ""''""' '"
menu 5 ' ' '"Begin inline math   "' "'INSERT ""ä'"
menu 5 ' ' '"End inline math     "' "'INSERT ""ä'"
menu 5 ' ' '"Begin superscript   "' "'INSERT ""^{""'"
menu 5 ' ' '"End superscript     "' "'INSERT ""}""'"
menu 5 ' ' '"Begin subscript     "' "'INSERT ""_{""'"
menu 5 ' ' '"End subscript       "' "'INSERT ""}""'"
menu 5 ' ' '"Begin display math  "' ,
    "' ""txedstuff/newline"";INSERT ""$"";""txedstuff/newline""'"
menu 5 ' ' '"End display math    ,
   "' "'INSERT ""$"";""txedstuff/newline""'"
menu 5 ' ' '"Begin block paragraphs "' ,
   "'INSERT ""{\parindent 0pt"";""txedstuff/newline""'"
menu 5 ' ' '"End block paragraphs  "' ,
   "'INSERT ""}"";""txedstuff/newline""'"
menu 5 ' ' '"Begin italics "' "'INSERT ""{\it""'"
menu 5 ' ' '"Begin bold face"' "'INSERT ""{\bf""'"
```

```
menu 5 ' ' '"Begin typewriter "' "'INSERT ""{\tt""'"
menu 5 ' ' '"End special font "' "'INSERT ""}""'"
menu 5 ' ' '"Center a line of text "' "TxEdStuff/centerline"
menu 5 ' ' '"Left adjusted line    "' "TxEdStuff/leftline"
menu 5 ' ' '"Right adjusted line   "' "TxEdStuff/rightline"
menu 5 ' ' '"Begin narrow paragraph "' ,
   "'INSERT ""{\narrower"";""txedstuff/newline""'"
menu 5 ' ' '"End narrow paragraph   "' ,
   "'""txedstuff/newline"";INSERT ""\smallskip}"";""txedstuff/newline""'"
menu 5 ' ' '"TeX reference          "' "TxEdStuff/ref"
menu 5 ' ' '"Referee Report         "' "TxEdStuff/refrep"
menu 5 ' ' '"PhyzLett               "' "PhyzStuff/phyzlett.txed SLAC"
menu 5 ' ' '"PhyzMemo               "' "PhyzStuff/phyzmemo.txed"
menu 5 ' ' '"PhyzPub                "' "PhyzStuff/phyzpub.txed"
menu 4 ' ' '"send this to TeX  [F1]"' "TxEdStuff/txedtexify"
menu 4 ' ' '"Send chunk to Tex [F4]"' "TxEdStuff/viewchunk"
menu 4 ' ' '"Print this as Tex      "' "TxEdStuff/timatex"
menu 4 ' ' '"Print piece as Tex     "' "TxEdStuff/piece"
menu 4 ' ' '"Print chunk as Tex     "' "TxEdStuff/printchunk"
menu 4 ' ' '"VLT to front      [F8]"' "TxEdStuff/vltfront"
menu 4 ' ' '"Stop TeXing File       "' ,
   "'if showlist(""p"",""AmigaTeX"") then address ""AmigaTeX"" ""Abort""'"
menu 4 ' ' '"Turn Off TeX           "' ,
   "'if showlist(""p"",""AmigaTeX"") then address ""AmigaTeX"" ""Exit""'"
macro 1 "TxEdStuff/txedtexify"
macro 2 "TxEdStuff/toerror"
macro 4 "TxEdStuff/viewchunk"
macro 8 "TxEdStuff/vltfront"
macro r "TxEdStuff/ref"
macro b "MatchBrak"
macro c "ColCut"
macro p "ColPaste"
macro s "SetToggle"
macro t "Toggle"
if showlist('p',"FINDTXEDHOST") then address FINDTXEDPORT REFRESHLIST

exit 0

ERROR:
   exit 0
```

The ARChive, COLor, WRap, TAB and MENU commands are explained in the TxEd manual. By setting ARChive 7 we tell TxEd to save an unlimited number of backups to the **T:** directory and to consecutively number them. If your hard

disk runs out of space you should clean out this directory occasionally. Your **rexx:** directory should have a file called **purge.rexx** which cleans out all but the most current backup of each file. The remaining commands which configure menu options require understanding how REXX and TxEd work. If you are interested in modifying these examples and need help get in touch with Marvin Weinstein (NIV at SLACVM) or Willy Langeveld (WGLP09 at SLACVM), or even better, post your problems to the Amiga CONSPIRE conference, or new slac.users.amiga Netnews group.

7.3.2 TxEd Macros

The macros which allow you to match brackets, braces, parens, etc., are contained in the files

- rexx:txedstuff/colcut.txed
- rexx:txedstuff/colpaste.txed
- rexx:txedstuff/toggle.txed

If you do not have these macros in your **rexx:txedstuff** directory then you are not completely up to date. Get in touch with Marvin to acquire them.

7.3.3 TxEd-TEX Integration

There are several files which control the interaction between TxEd and AmigaTEX they are

- rexx:startup_tex.rexx
- rexx:txedstuff/doquit.txed
- rexx:txedstuff/newline.txed
- rexx:txedstuff/ref.txed
- rexx:txedstuff/refrep.txed
- rexx:txedstuff/leftline.txed
- rexx:txedstuff/rightline.txed
- rexx:txedstuff/vltfront.txed
- rexx:txedstuff/txedtexify.txed
- rexx:txedstuff/centerline.txed
- rexx:txedstuff/piece.txed
- rexx:txedstuff/timatex.txed
- rexx:txedstuff/viewchunk.txed

If you run into a problem with these execs that you can't handle, or if you are missing execs which you would like to have, get in touch with Marvin or Willy, or even better post your problems to the Amiga CONSPIRE conference, or new slac.users.amiga Netnews group.

7.3.4 PHYZMemo, PHYZLett, PHYZPub

Finally, all of the ARexx execs which control the menus for letters, memos and publications, are found in the directory **rexx:phyzstuff**. The files you should find there are

- rexx:phyzstuff/phyzpub.txed
- rexx:phyzstuff/phyzlett.txed
- rexx:phyzstuff/phyzmemo.txed
- rexx:phyzstuff/phyzlett.rexx
- rexx:phyzstuff/phyzmemo.rexx
- rexx:phyzstuff/phyzpub.rexx

These are quite complicated ARexx execs, however you should not have to tamper with them. The only things you might want to change have to do with the default strings which appear when you create a letter, memo, etc. The simplest way to do this, if your execs are up to date, is to go to a CLI and type

```
> makedir env:phyzstuff
```

This creates a directory wherein the default values of the strings which will appear in the different menus are kept. To set these values you issue the following commands from the CLI: For memos

```
> setenv  phyzstuff/memoto          name of person
> setenv  phyzstuff/memofrom        name of person
> setenv  phyzstuff/memofile        name of file
> setenv  phyzstuff/memocopies      list of people
> setenv  phyzstuff/mainmemotopic   text
```

For letters

```
> setenv  phyzstuff/mysalutation    Dear Whoever
> setenv  phyzstuff/mysignature     Very Truly Yours,
> setenv  phyzstuff/letterfile      vd0:letter.tex
> setenv  phyzstuff/mybin           80
```

NOTE: If you are working on an Amiga 3000 or running AmigaDOS 2.0 then you need to do two more things; namely, issue the commands

```
> makedir envarc:phyzstuff
```

```
> copy env:phyzstuff to envarc:phyzstuff
```