ZFC
Keizersgracht 684/3
1017 ET Amsterdam
The Netherlands
Phone: +31 20 4 208 248
Email: `zfc@zfc.nl`
WWW: `http://www.zfc.nl/`

Note on support and maintenance. The **Edith** package is no longer under development for the Atari series. Version 1.221P is the final version. Development in general is continued for **UNIX** computers in the form of **Edith Professional for X11**.

The development of **Edith** for the **ATARI** series has had benefit from the help and support of a large group of friends, family, collegues at the universities of Utrecht and Leeds, the Centre for Mathematics and Computer Science in Amsterdam and many users of the internet. Thank you all very much.

This manual was produced using **Edith**, **XIMG-Snapshot**, pixel-oriented **ATARI** drawing packages, **XV** and LaTeX.

Contents

1 Introduction

Thank you for buying an **Edith** manual. We hope you will enjoy using the program; a powerful but easy-to-use editor should increase your productivity in a pleasant way, and the manual will, without doubt, tell you things you couldn't have discovered from within the program. Unfortunately, the **Edith** Pro package is no longer maintained for the **ATARI** series. This means that your present copy, V1.221P, is the last ever to be available.

Although any comments you have will be highly appreciated, and we will consider them in the design of future systems of **Edith** for the X Window system, fixes to the present **ATARI** version are out of the question.

This manual is, from this point on, identical to the one issued with the commercial edition of V1.221P.

1.1 System Requirements

Edith can be used on any **ATARI ST/TT/Falcon**, **Medusa** or **Eagle** computer running a single or multi tasking operating system (**TOS**) with version numbers between 1.0 and 4.2, and is likely to run without problems if your computer is of a later design. There are no special requirements for memory or screen resolution. 512 kilobytes of memory are sufficient for basic editing, and **Edith** will adapt itself to screen resolutions as limited as ST Low Resolution (320 × 200 pixels).

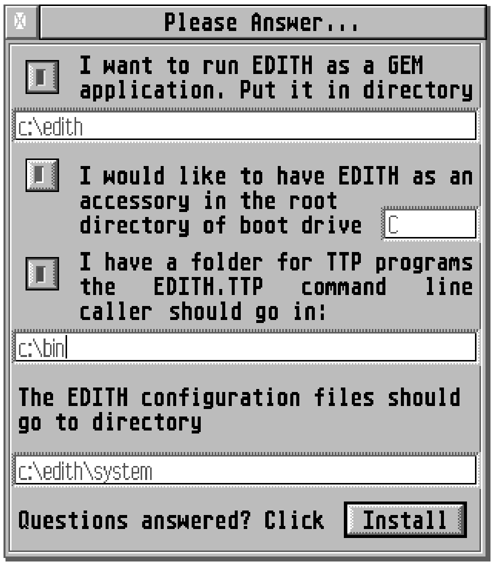
Edith takes advantage of **MiNT** and **MultiTOS**.

1.2 How to Install Edith

Always make a copy of the original **Edith** disk. If the disk gets damaged, you will have to order a new disk (even though this is at a reduced price, it will cost you time and money).

To use **Edith** on a *floppy disk based system*, simply use the copy of the original disk. If you want to use it as a **GEM** application, double click on `EDITRESS.APP` in the folder `A:\EDITH`. If you want to use **Edith** as a desk accessory, make a copy of the original **Edith** disk, move the files `A:\EDITH\EDITRESS.APP` and `A:\EDITH\EDITRESS.ENV` to the root directory of the disk. Then rename `EDITRESS.APP` to `EDITRESS.ACC`, and reboot your computer while the new disk is in drive `A:`.

To install **Edith** on a *hard disk*, put the copy of the **Edith** disk in the disk drive and double click on `INSTALL.PRG`. In the dialog form which appears, choose for either or both the accessory and application versions by selecting the appropriate boxes. Select the third box (for `EDITH.TTP`) if you are frequently using **GEMINI** or **MultiTOS**. If necessary, change the directories where **Edith** will be installed. Then click **Install**. During installation, `INSTALL.PRG` may ask you a few questions via alert boxes. After you have answered all these questions, and there were no errors, an alert box will tell you that **Edith** has been successfully installed on your hard disk.



NOTES.

1. Never change the name of the files `EDITRESS.APP` and `EDITRESS.ACC`, as otherwise, **Edith** may not function properly.[1]

2. Although `INSTALL.PRG` will install the **Edith** fonts for you, the fonts must be added to the `ASSIGN.SYS` file manually. The fonts are not strictly necessary, but certainly enhance the appearance of **Edith**'s user interface.

1.3 How to Run Edith from your Hard Disk

After you have installed **Edith** using `INSTALL.PRG`, the editress is ready for use. If you have chosen for the desk accessory version, its name will show up in the *Desk* menu after you have re-booted your computer, and you can activate it by selecting its name in the *Desk* menu, or by pressing the Control Shift Shift keys simultaneously. If you have chosen for the **GEM** application version, it has been copied to the directory you specified in the installation procedure.

1.4 Running EDITRESS.APP from the ST/STf Desktop

In general, **Edith** is started by double clicking on `EDITRESS.APP` from the directory where it was installed. Read your Atari manual if you want to install **Edith** as an application for certain types of files.

1.5 Running EDITRESS.APP from NEWDESK

If you are in possession of a more recent version of the Atari operating system (such as supplied with the **Mega STE**, **TT** and **Falcon** series), you can drag `EDITRESS.APP` onto the desktop. You can drag and drop a file onto `EDITRESS.APP` or its desktop icon, which will then automatically pop up in a text window. When you install **Edith** as an application to certain file extensions, choose *default path: application* and *parameter: full path* in the desktop's *Install Application* dialog.

[1] Of course, you may temporarily change `EDITRESS.ACC` to `EDITRESS.ACX` without problems.

1.6 Using MultiTOS

When **Edith** is running under **ATARI**'s multi tasking system **MultiTOS**, there will be no menu bar and desktop, but there will always be at least one **Edith** window or icon open; the file selector and text windows will always have a menu bar. Windows can be iconified by clicking on the window *smaller* which appears at the right hand side of all windows[2]. Also, you can drop desktop icons onto **Edith**'s windows to load or insert texts. See also section 1.9 below.

1.7 Using GEMINI and the Edith Accessory

The **Edith** accessory and **GEMINI**[3] are a very powerful combination. You can create an icon for the accessory; you can drop **GEMINI** icons onto the **Edith** icon or onto **Edith**'s windows to load or insert texts. You can drop icons in **Edith**'s clipboard window onto some of **GEMINI**'s icons and windows as well. You can type text into the console window even when one of **Edith**'s windows is on top of it. **GEMINI**'s windows are automatically updated when **Edith** has written to your disk or hard disk. See also the next section.

1.8 Initial Command Line Options

The following command line options are recognized when `EDITRESS.APP` is started. (*Advanced*)

`-nodesk` Don't install a desktop and menu bar. This is automatically switched on in the **Edith** accessory and when the screen is too small for the menu bar and function keys (ST Low and TT Low resultions).

`-noprev` Don't copy the desktop pattern of the calling application. This is automatically the case when **MiNT** or **MultiTOS** is running.

`-pr[eferences]` **file** Specifies the default preference file. If this is not a full file name, it should be in the **Edith** system directory.

`-di[mensions]` **file** Specifies the default dimensions file.

`-co[lours]` **file** Specifies the default colours file.

`-ke[ys]` **file** Specifies the default keyboard parameters file.

`-ku[rzels]` **file** Specifies the default kurzel file.

`-i[conic]` Open all windows as icons, and don't install desktop and menu bar. Under **MultiTOS**, desktop and menu bar are never installed. Repeated use of `-i` can be used to switch iconic mode on and off.

Following the initial parameters, you can specify any number of files (or *run-time parameters*, as explained on page 17.

1.9 Using the command line caller EDITH.TTP

When you are using a multi tasking system or you have **Edith** installed as an accessory, you will want to be able to open new files without starting a second copy of the program. (*Advanced*)

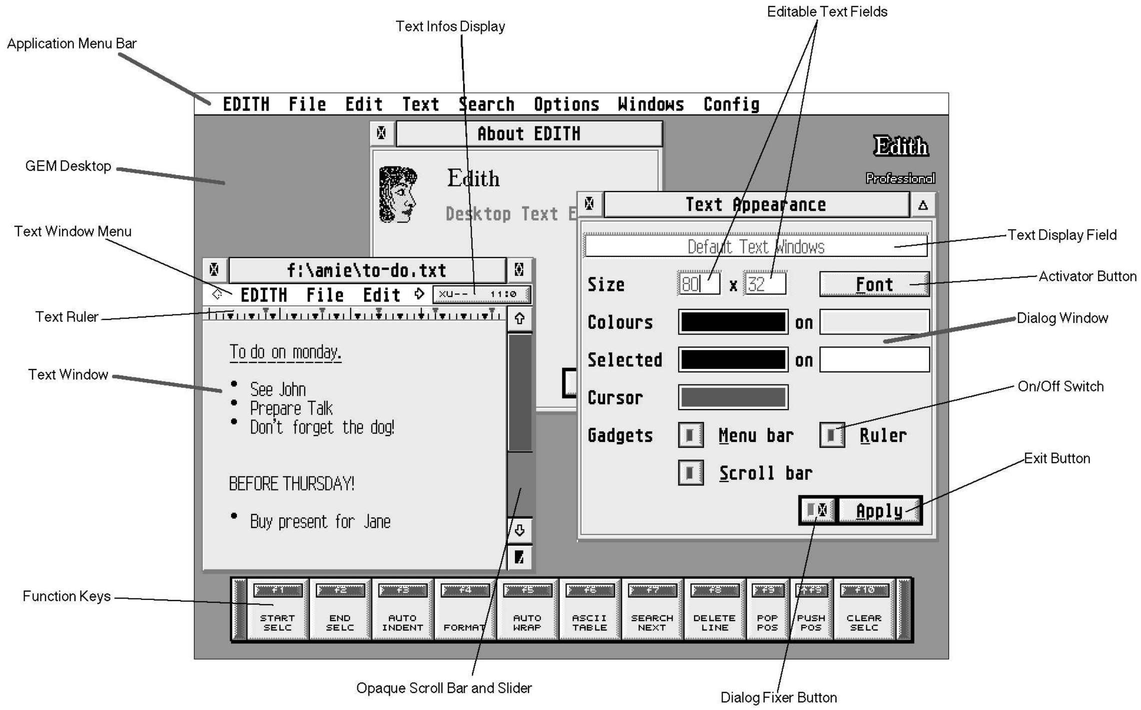
When you have selected the `EDITH.TTP` box at installation, a special program `EDITH.TTP` is copied to your hard disk; it is normally put in `C:\BIN`. This program allows you to open text files from the **Mupfel** console window or a MINIWIN/shell window, when a copy of the Editress is already running (for example, as a desk accessory).

NOTE. The command line caller `EDITH.TTP` makes use of the environment variable `EDITHSYSTEM` which contains the name of the directory containing **Edith**'s configuration files. This variable is defined in the `EDITRESS.ENV` file. The environment definitions in `EDITRESS.ENV` are automatically generated by the installation software. The installation software also tries to find any configuration scripts for **GEMINI** and **MiNT**,

[2]Except for *modal* windows; see page 12 for an explanation.

[3]**GEMINI** is an alternative desktop for **ATARI TOS** computers, recommended for any **ATARI** user who has a hard disk and preferably 2 megabytes of internal memory. It is available from most **FTP** sites (such as `ftp.uni-paderborn.de`) and the **MAUS** mailbox.

and will offer to patch those. If it does not inform you about **GEMINI** or **MiNT** configuration files, the software may have failed to find these. In that case, you must copy the definition of the `EDITHSYSTEM` variable from your `EDITRESS.ENV` to your `MUPFEL.MUP` and/or `GEM.CNF` files manually.



The **Edith** desktop.

2 The Graphical User Interface

2.1 The Desktop

By default, **Edith** has a *desktop* and *menu bar*, like most other **GEM** applications. Most **Edith** commands are available in the menu bar, some only in dialog boxes or through keyboard combinations. The desktop contains a graphical representation of the function keys. Clicking on one of the function key buttons is the same as pressing the corresponding function key.

In very low resolutions (less than 80 columns in the standard font, such as ST Low and TT Low), or when one of the command line options `-i` (iconic) or `-nodesk` is specified, there will be no desktop and menu bar.

2.2 Text Windows

Edith allows editing of six texts simultaneously. On the screen, a text is represented by a text window or an icon.

2.2.1 The Layout of Text Windows

A text window consists of a *text pane* showing a part of the text, and a number of *gadgets*. These gadgets can be switched on and off in the *Text Appearance* dialog; see page 42. The gadgets are

▷ A *menu bar*, which contains precisely the same menus as the desktop menu. This is especially useful when running **Edith** as an accessory, or in iconified mode. When the window is not wide enough to hold all menu entries, there will be two arrows which let you scroll through the menu.

▷ A *text information box*, showing:

- The *status* of the text; **-** if the text was not changed since last reading it from or writing it to disk, **X** if it has been irrestorably changed, and **/** when it was changed, but by a single *Undo* operation it can be restored in its original form.
- The possibility to *Undo* the last operation; **U** when *Undo* is possible, **-** when it is not.
- The presence of a *selection*; **S** when there is a selection, **-** if there is no selection. This is a useful option, since a selection in an **Edith** text need not be inside the part of the text which is displayed in the window.
- Editability: a **-** if the file can be edited, an **R** if the text is read-only. (Help pages and text buffers.)
- The type of newlines in a text: **d** for *Dos*, **m** for *Macintosh*, or **u** for *Unix*.
- The position of the text cursor (*row* and *column*).

▷ A *text ruler*, showing the columns of the text window, and two sets of tabulator stops (see page 20).

▷ The *text* and a *scroll bar*. As is standard in **GEM** applications, the size of the slider in the scroll bar gives an indication of how big a part of the text is displayed inside the window. The scroll bar, when operated with the *left* mouse button, lets you scroll through the text *opaquely*, i.e., the text moves up and down immediately as you move the slider. If you hold the **Shift** key before pushing the left mouse button, scrolling will be *proportional*, i.e. the text moves up/down 1 line as you move the mouse down/up 1 pixel. When you are using the *right mouse button* instead, the scroll bar will behave as is normal under **GEM**, i.e. it will be in *rubber box* style.

2.2.2 Types of Text Window

There are six different kinds of text window (the *text window classes*): *Default* and *Alternative* windows are for normal text editing, but the colour, size, font and text ruler settings for these types may be different. *Binary* text windows allow basic editing of non-text files in a hexadecimal format compatible with the **C** programming language. A *Text Buffer/Trash Can* window shows the contents of the trash can or one of the text buffers. The *Note Pad* window is for general use. Finally, there are *Manual Pages*.

2.3 Dialogs

Communication between you and **Edith** always takes place through **GEM** windows. While many older **ATARI** applications cause other programs (such as desk accessories) to be stopped when the user is answering a dialog, in a multi-tasking environment this is no more satisfactory.

Modal and Permanent Dialogs; the dialog fixer

Edith's *Dialog windows* can be *permanent* or *modal*. **Edith** will use a *modal dialog* whenever there is vital information which needs to be provided before **Edith** can proceed. This can for instance be the file selector for saving a text file, or an alert box. A modal window does not have a closer box, and the menu bar and all other **Edith** windows are disabled until the modal dialog is finished (by selecting, e.g., the **OK** button). When answering a modal dialog, windows of *other* applications can still be used.

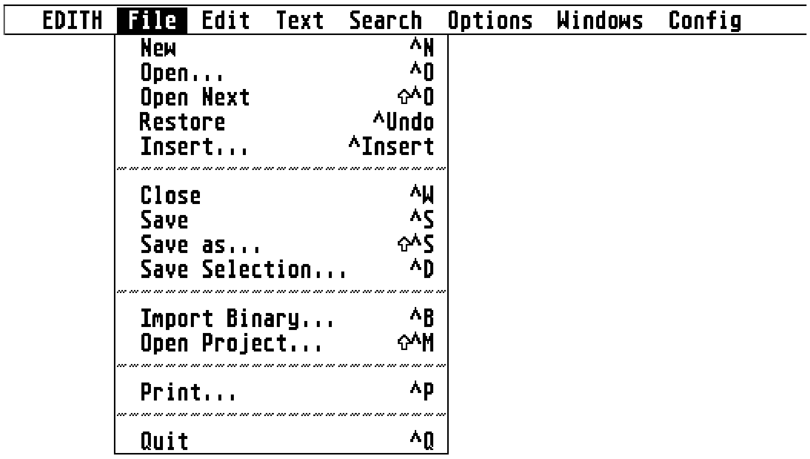
A *permanent* or *nonmodal* dialog window can remain open while you are working with other windows. The picture on page 9 shows how you can simultaneously work with dialog windows and text windows. If a permanent dialog has an **OK** button, then to the left of this button you will usually find the *dialog fixer* (see the illustration on page 9). If the fixer is selected, then selecting **OK** will also cause the dialog window to be closed. If it is not selected, the dialog window will remain open until you click on the window closer, or select another exit button, like **Cancel**. Fixing/unfixing a dialog can also be done by selecting *Options/Fix Dialog* in the menu bar.

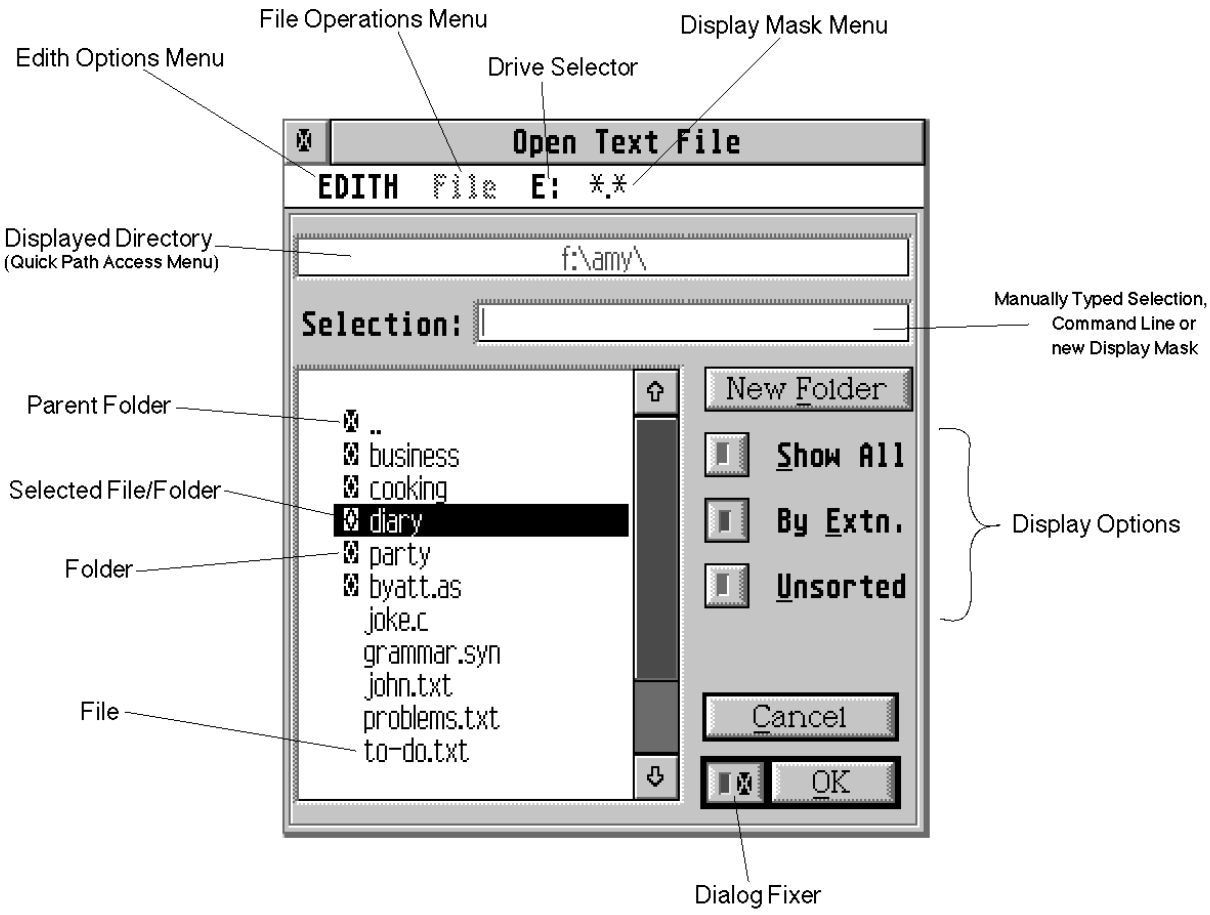
2.4 Using the Keyboard

Almost everything you can do with the mouse is also possible from the keyboard. All keyboard alternatives are listed in appendix B. However, for most of these options, you don't need this manual, as they can also be found in the program. *Menu shortcuts* are shown in the menus. These are usually combinations using the **Control** (indicated by a ^ symbol) and **Shift** (indicated by a $\Uparrow$ symbol) keys. Buttons in dialogs can usually be operated by a combination using the **Alternate** key. One of the letters in such a button is underlined; combining that letter with the **Alternate** key then activates the button.

2.5 The 3D interface

When sixteen or more colours are available, **Edith** will automatically show up in colourful 3D style. When using **MultiTOS** or on a **Falcon** computer, the scroll bars will always be in 3D style, regardless of the number of available colours. The appearance of **Edith**'s 3D buttons can be set (varying from light grey to dark grey) in the *GEM Options* dialog **Control** **1**.





The *File* menu and the file selector.

3 Loading and Saving Text Documents

This section describes all methods **Edith** offers to load and save files, in increasing order of complexity. A beginning user will be satisfied with the usual way of selecting files in the File Selector (sections 3.2 and 3.3). Advanced users will want to exploit the possibility to drag and drop files (section 3.4) or use the command line options (section 3.6).

3.1 The File Selector

Edith has a built-in File Selector which uses a window and does much more than the standard **TOS** Item Selector. A beginning user however may prefer to use the familiar **TOS** Item Selector instead. Therefore the choice between the two selectors can be set in the *GEM Options* dialog (see page 41).

In general use, the **Edith** File Selector behaves very much like the standard Item Selector. To select a file, double click on its name; select it and click on **OK**, or type its name and press **Return**. To create a new file, type a new name and press **Return**. To enter a folder, click on its name. The following things which are also possible in the standard Item Selector are different in the **Edith** File Selector:

Closing a directory There are two ways to go up in the file hierarchy. The usual closer box is not present,[4] but clicking on the directory `..` has the same effect. In addition to this, the box which displays the name of the directory you are viewing activates a pull down menu with the name of a number of useful directories, for quick access.

Changing Drive The drive identifier in the File Selector's menu bar activates a pull down menu with the drive identifiers `A:` through `Z:`.

Changing the Displayed Selection To limit the number and type of files displayed, a file mask (normally `*.*`) can be set. The mask is displayed in the File Selector's menu bar. Click on the mask to open a pull down menu with eight standard file masks, and the option *Edit*. Use *Edit* to change the eight standard masks according to your preferences. Finally, the mask can be changed by typing a wildcard expression into the *Selection* field and pressing *Return*.

Apart from the usual wildcards `*` and `?`, Edith supports character ranges such as `[a--zA--Z]`. See also page 32.

The following new options are provided:

It runs in a Window The **Edith** File Selector runs in a window, like the other **Edith** dialogs. Normally, the File Selector is a *modal* window, except when it is used to *open text or binary files*. The *non-modal* File Selector can remain open while you are editing texts (see **Modal and Permanent Dialogs**, page 12).

Especially useful when using **Edith** as a Desk Accessory, the *non-modal* file selector provides the *Edith* menu with access to general **Edith** options from the Application Menu Bar.

The File Menu The File Selector's *File* menu provides the options

File Info Gives information about a file.

Delete File After your confirmation, deletes a file.

Touch File Touches a file (sets the file's *time stamp* to the current time and date). (*Advanced*)

New Folder Allows you to create a new folder.

Show All If this option is selected, hidden files are shown. (*Advanced*)

By Extn., Unsorted Like in the **TOS** desktop, you can choose to display files sorted by name, by their extension, or unsorted. Whichever option is selected, directories are always listed first.

[4] Experience shows that this will result in confusion as the file selector *window* already has a closer box

Using the Keyboard Most options in the File Selector are also accessible without using the mouse. The cursor keys and standard cursor options such as *home*, *page down* etc. can be used to browse through the files. Further key combinations are

Alternate I File Information

Delete Delete File

Alternate T Touch File

Alternate D Enter the identifier of a New Drive

Alternate F New Folder

Alternate S Show All on/off

Alternate E Sort by extension on/off

Alternate U Unsorted on/off

Tab Complete File Name (only when files are sorted by name)

Typing full Command Lines When *opening text files*, you can type a *full command line* into the *Selection* field, so in one rush, without touching the mouse, you can open several files at once. See page 17 for more about command lines.

3.2 Opening Files

To load a normal text file, choose the *Open...* Control O option in the *File* menu, and select the file to be opened in the File Selector. When the *Alternate* key is held down while selecting *Open...*, or while selecting the OK button in the File Selector, the text will be opened in an alternative text window (see page 11).

To load a binary text file, i.e. anything which is not a simple text in **ASCII** format, select *Import Binary...* Control B from the *File* menu, and choose a file in the File Selector. More about editing binary files is found in chapter 9, page 37.

To create a new, untitled text document, select *New* Control N from the *File* menu. If the Alternate key is held down while selecting *New*, the new document will be opened in an alternative text window. To create an empty editor for *binary* files, use *Import Binary...* and type the name of a file which does not yet exist.

3.3 Closing and Saving Files

After editing a text, it can be closed by clicking on the closer box, or selecting *Close* Control W in the *File* menu. When a text window is closed, and it has been changed since it was last read from or written to a file, **Edith** will automatically ask you whether the changes must be saved.[5]

When editing a large text, it is wise to write it to a file at regular intervals, as you may accidentally throw away text which later turns out to be very useful, or worse, there may be a power breakdown just after you typed a thousand lines of brilliant text![6]

Use *Save* Control S in the *File* menu to save a file. This option is only available when the text has been changed since it was last written to disk. Use *Save As...* Shift Control S to save the text to a new file with a different name.

[5]The option *Close Top* Control U in the *Windows* menu has the same effect, but will apply to the top window instead of the active window; see page 41. The key combination Control U has been chosen compatibility with existing, mostly German, software. Unfortunately, this could not be maintained for the Control W combination which used to stand for *cycle windows* (German *wechseln*). The keyboard shortcut layout may change in the future as developers on the internet are currently working on a new standard keyboard layout proposal.

[6]Of course, even regularly saving your file does not guarantee that you will never need to recover lost data. For ultimate safety in text editing, Edith offers a *trash can* (see page 26) and an automatic *carbon copy* function (page 43).

3.4 Dragging & Dropping Files

If you drop one or more files onto the **Edith** executable or its icon in the **TOS**[7] desktop, the editress is started, and instead of opening the file selector, it will open those files.

When you are using **GEMINI**, and have **Edith** installed as an accessory, there is a number of additional options. You can drag the accessory file `EDITRESS.ACC` onto the desktop. Then, you can drop files onto that icon to edit them. Since **Edith** is then already running, the files are loaded very quickly. When an **Edith** text window is open on the **GEMINI** desktop, you can drag icons onto the text window; these files will then be inserted into the text. If you drag a folder icon onto a text window the File Selector will be opened, where you can select a file to insert. Dragging files to other windows, or the menu bar of a text window opens them.

When you are using **MultiTOS**[8], the same operations are available; apart from this you can also drag pieces of text to windows of the **TOS** desktop, you can drag pieces of text into **Edith** windows (such as selections from **MINIWIN**).

3.5 The Input Queue

Edith can open at most six texts windows. If you attempt to open more than six texts, an alert box will ask you whether the remaining files should be *queued*. This means that **Edith** will remember their names, and when you close all text windows, she will remind you that those texts are still to be opened. You can then choose to open them straight away, let them wait, or abandon them.

When there are files in the input queue, you can open the next file by selecting *Open Next* **Shift** **Control** **O** in the *File* menu.

NOTE. Files from the input queue will always be opened in default text windows.

3.6 Loading Files by Command Line

The following (*run-time*) command line options are supported at startup, from the *open text* File Selector or from the command line caller `EDITH.TTP`. (*Advanced*)

`-i[conic]` All files following this switch will be opened as icons.

`-a[lternative]` All files following this switch will be opened in alternative text windows.

`-b[inary]` All files following this switch will be opened in binary mode.

`-no[tepad]` Open the note pad window

`-tr[ash]` Open the trash can window

`-cl[ipboard]` Open the clipboard window

`-ab[out]` Open the *About Edith* dialog

`-n[ext]` Open the next file in the input queue, if there is one.

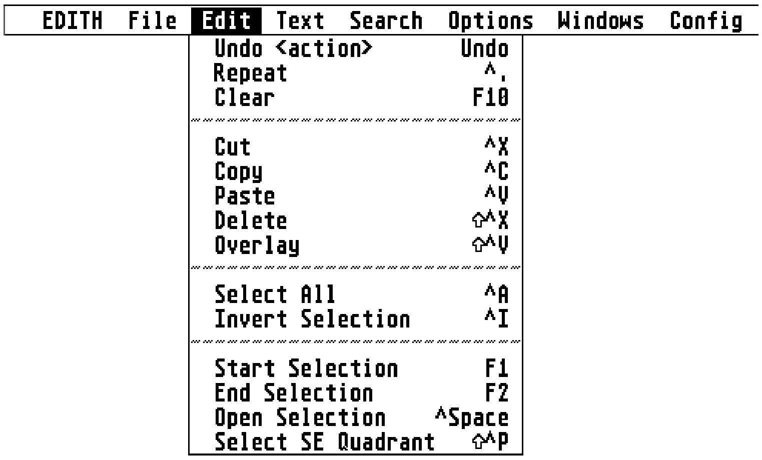
`-p` **file** Process project **file**. Everything following this option on the command line is ignored.

file [+ line-no] Open **file**. If **line-no** is specified, the cursor will be placed at line **line-no**, otherwise it will be on the first line of the text.

3.7 Projects

A *project file* is a text file containing a command line. Project files may be a convenient option if you are regularly editing the same set of files. A project file can contain any number of file names, which will be queued if there are not enough windows available when the project is opened. All *run-time* command line parameters can be used in a project file. (*Advanced*)

When the option *Save open texts* in the *At Exit* dialog is selected. A project file is automatically generated when leaving **Edith**.



The *Edit* menu

[7] **TOS** versions 2.0 and up only.
[8] You will need a very recent version of **MultiTOS**, probably **TOS** 5.0 or higher

4 Basic Text Editing

This chapter describes the basics of text editing with **Edith**, i.e. how you move the cursor, type and remove small bits of text.

4.1 Moving Around

Browsing with the Mouse

The fastest and easiest way to browse through a large text is to use the *slider* right of the text (see the picture on page 9). When you click on the slider with the left button, and move the mouse up and down while holding down the left mouse button, you will find yourself sliding through the text as fast as your the **GEM** system in your computer allows. For more precise movement, click on the arrow buttons above and below the slider, use hold down a **Shift** key while moving the slider. To jump to a point in the text without sliding through other parts, use the right mouse button instead.

Positioning the Text Cursor

The *text cursor* is the thin, curled vertical line[9] blinking in a text window. It marks the point where the text you type will be inserted. You can move the cursor by clicking anywhere in the text window with the left mouse button, or by using the arrows on the **ATARI** keyboard. The following key combinations are available for moving through the text:

⇑ Move the cursor one line up

⇓ Move the cursor one line down

⇒ Move the cursor one character to the right; if the cursor is at the end of a line, it will move to the beginning of the next line.

⇐ Move the cursor one character to the left; if the cursor is at the beginning of a line, it will move to the end of the previous line.

Shift ⇑ Move the cursor one page up, where a *page* is precisely the number of lines visible in the text window.

Shift ⇓ Move the cursor one page down.

Shift ⇒ Move the cursor to the end of the line.

Shift ⇐ Move the cursor to the beginning of the line.

Control ⇑ Move the cursor to the top of the text window.

Control ⇓ Move the cursor to the bottom of the text window.

Control ⇒ Move the cursor one word or inter-word space to the right. A *word* is a sequence of letters, digits and underscores. An *inter-word space* is any sequence of other characters. So when you press this combination repeatedly, the cursor will stop at both the beginning and the end of each word.

Control ⇐ Move the cursor one word or inter-word space to the left.

Shift **Control** ⇑ Move the cursor up a quarter of a page.

Shift **Control** ⇓ Move the cursor down a quarter of a page.

Clr/Home Move the cursor to the beginning of the text.

---

[9] Or a rounded box, when **Edith** is in *replace* mode.

Shift **Clr/Home** Move the cursor to the end of the text.

Shift **Control** **Clr/Home** Move the cursor to the middle of the text.

NOTES.

1. In addition to this, the numerical keypad can be configured to act like an **MS-DOS** keyboard with *num lock* switched off. See page 44.

2. The **Clr/Home** combinations will, in most cases, push the original text position onto the *position stack*. This will cause a small black triangle to appear on the scroll bar, marking the position of the cursor before it jumped to the beginning/end of the text. See page 33 for more.

4.2 Line-oriented Editing

Line-oriented operations are for typing regular text and small-scale editing not involving the manipulation of large blocks.[10] The most important line operation is of course typing text, which is normally inserted at the point indicated by the text cursor. The following other operations are available from the keyboard

0–**9** **a**–**z** **A**–**Z** **!** **$** **%** **&** ***** **etc.** when typed are directly inserted into the text.

Backspace Delete the character left of the cursor. If the cursor is at the beginning of a line, append the current line to the previous one.

Delete Delete the character right of the cursor. If the cursor is at the end of a line, append the next line to the current one.

Shift **Backspace** Delete everything left to the cursor.

Control **Backspace** Delete the word or inter-word space left of the cursor.

Control **Delete** Delete the word or inter-word space right of the cursor.

Insert Toggle between *insert* and *replace* mode. In insert mode, text typed will be inserted into the text; in replace mode, typing one character will replace the character originally at the cursor position.

Control **.** (*Repeat*, *Edit* menu) Repeat the last sequence of operations on one line.

F10 (*Clear*, *Edit* menu) Clear, among other things[11], the buffer that stores the sequence for **Control** **.** .

4.3 Indentation

Edith has a sophisticated system for indentation that requires a little more understanding than a standard system using *tabs*. There are two sets of tab stops. Those in the *first* set, the *true tab stops* are normally at positions 8, 16, 24, 32 etc. and are shown in the text ruler as the upper row of triangles. Whenever a *tab character* (ASCII code 9) appears in a text, a blank space will be displayed stretching until the next true tab stop. Those in the *second* set are called the *editor* tab stops, and are normally at positions 0, 4, 8, 12, 16 etc. These are displayed in the text ruler as the lower row of triangles.

NOTE. The right margin, true tabs and edit tabs can be at any horizontal position smaller than or equal to 256. The minimal distance between two tab stops is 2. The minimum value for the right margin is 10.

[10] A formal distinction between line-oriented operations and structure-oriented operations must also be made, because Edith considers a sequence of small changes to a single line as one action. See chapter 8.

[11] The *selection* and the *position stack*.

The TAB key

In most text editing systems, the **Tab** key simply inserts a tab character (ASCII code 9) into the text, creating a space stretching out until the next *true tab* stop. **Edith** allows an alternative mode of operation, where the **Tab** key inserts a number of blanks (ASCII code 32) into the text, stretching out until the next *editor* tab stop.

The advantage of the second mode (tab key generates edit tabs) is that text files using true tab characters are displayed correctly,[12] even while the **Tab** key generates a smaller level of indentation which is much more convenient for editing texts and program source code. The tab key mode can be selected in the *Editor Options* dialog **Control** **4** (*Options* menu). See also page 43.

The ESCAPE key (back-indent)

The **Escape** key acts very much as the *opposite* of the **Tab** key. When it is pressed, **Edith** will look back into the text to find the first line whose indentation is smaller than that of the current line, and then moves the cursor back to that level, deleting all space in between.

The **Escape** option was added as a compensation for the fact that a set of space characters generated by the **Tab** key cannot be deleted, like a true tab character, by simply hitting the **Backspace** key once.

NOTE. When the **Escape** key is pressed, and there is no white space immediately left of the cursor, the *Kurzel expansion* machanism is engaged instead. See **Kurzels**, page 47.

4.4 Special Characters

Apart from letters, digits and punctuation symbols, the **ATARI** character set contains a number of special characters (such as arrows, close, open pictograms, the **ATARI** symbol and mathematical symbols), accented letters, a Hebrew alphabet, and some Greek characters. The easiest way to get these characters is by using the *ASCII Table* dialog **Shift** **Control** **A**, in the *Options* menu. In this dialog, you can select one of the open text windows and insert a special character into the text by clicking on it with the left mouse button. It also shows the ASCII codes of the characters as you move the mouse over the character set.

A quicker method is the following: in a text or dialog window, hold down the **Alternate** key, enter the ASCII code of the desired character, and release **Alternate**.

Accented characters can also be entered using the *dead key* feature. The available combinations are shown in the figure on the next page.

NOTE. Some of these combinations may be different on non-UK keyboards.

4.5 Adding and Removing Lines

The following other basic operations, which are not line-based, are available:

Return Insert a new line, and move everything right of the cursor to that line. If the *auto indent* function is on, the amount of indentation on the original line will be copied to the new one. See also page 43.

Shift **Insert** Same as **Return**, but the cursor is not moved to the new line.

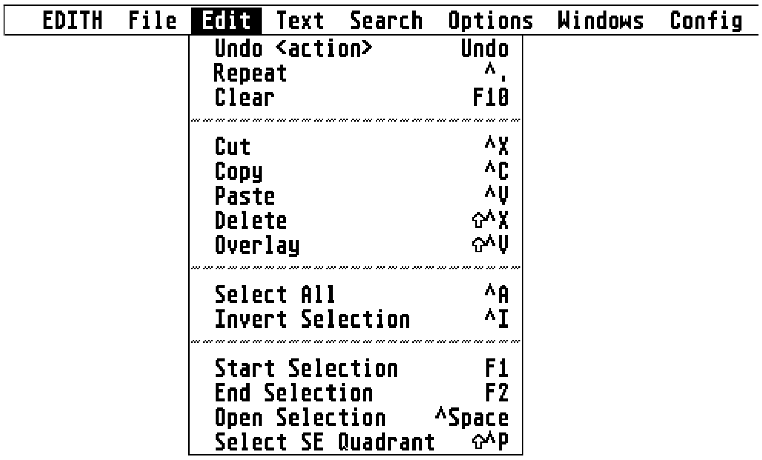
Control **Y** (*Cut Line*, *Text* menu) Delete a line, and copy it, as in the *Cut* operation (see chapter 5).

Shift **Control** **Y** or **F8** or **Shift** **Delete** (*Delete Line*, *Text* menu) Delete a line, as in the *Delete* operation (see chapter 5).

[12] The distance of 8 columns between true tabs is more or less standard. As a general rule, never distribute a text file which is supposed to be displayed with a true tab size other than 8. This is why we strongly urge **Edith** users to use editor tabs rather than true tabs. True tabs, especially if not spaced at intervals of 8 columns, are a very undesirable personalization of your text files.

Combination	followed by	yields
Control + ,	**c**	ç
	C	Ç
Control + **Shift** + ”	**a**	ä
	e	ë
	i	ï
	j	ĳ
	o	ö
	u	ü
	y	ÿ
	A	Ä
	J	Ĳ
	O	Ö
	U	Ü
	Space	¨
Control + ’	**a**	á
	e	é
	i	í
	o	ó
	u	ú
	E	É
	Space	´
Control + **0**	**a**	å
	A	Å
Control + \	**!**	¿
	?	¡
	8	∞
	c	Hebrew ‘c upside down’
	e	e upside down
	P	¶
	U	∩

Combination	followed by	yields
Control + **Shift** + ^	**a**	â
	e	ê
	i	î
	o	ô
	u	û
Control + **Shift** + _	**a**	ª
	o	º
	L	Pound symbol
	Y	Yen symbol
Control + ‘	**a**	à
	e	è
	i	ì
	o	ò
	u	ù
	A	À
Control + |	**c**	Dollar cent symbol
Control + ˜	**a**	ã
	n	ñ
	o	õ
	A	Ã
	N	Ñ
	O	Õ



The *Edit* menu

5 Selections and Text Buffers

Methods for manipulation of larger or more complex collections of text make use of the concepts *text selection* and *text buffer*.

A *text selection* is an arbitrary number of highlighted fragments in a text window. When a selection is present, all structure-oriented operations will be confined to the selected fragments.

A *text buffer* contains a piece of text divided into an arbitrary number of fragments. **Edith** has three text buffers available to the user: the *cut/paste text buffer*, the *delete/text operation buffer* and the *all purpose text buffer*. The contents of the text buffers can be viewed at any time by opening the *Clipboards* panel **Shift** **Control** **B** from the *Options* menu.

5.1 Making Selections

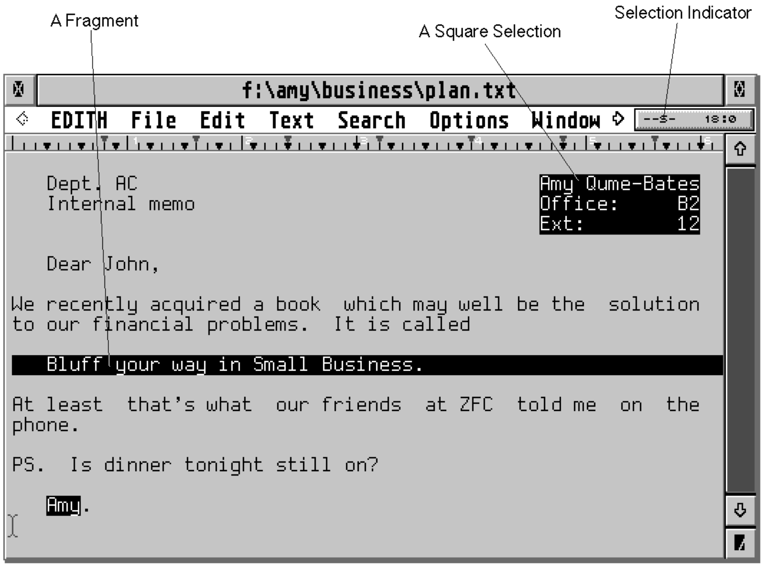
The **Edith** selection system is a generalization of that of traditional cut/paste-based applications. If you are familiar with an existing graphical editing system (any editor package inspired on the **Apple Macintosh** for example), then in many ways **Edith** will behave as you expect, but there will be some differences.

5.1.1 Using the Mouse

The easiest way to select a fragment of text is to move the mouse over the fragment while holding down the left mouse button. The fragment will be highlighted as you move the mouse, and in the text infos display in the top right corner of the text window, the letter **S** will appear to indicate that there is a selection. A double click, or a click at the cursor position selects a word. A triple click, or a double click at the cursor position selects a line.

When the **Shift** key is held down while making a selection, a fragment is *added* to the selection. There is no limit to the number of fragments simultaneously selected.

When the *right* mouse button is used, a *rectangular selection* (called *square selection* in the picture below) is created, i.e. a column of text, as opposed to the normal, contiguous selections.



Example: types of selection.

5.1.2 Other Selection Facilities

In addition to the facilities for making selections with the mouse, the following options are offered in the *Edit* menu:

Select All **Control** **A** Select the entire text as a single contiguous fragment.

Invert Selection **Control** **I** Select precisely those parts of the text which are not selected in the current selection.

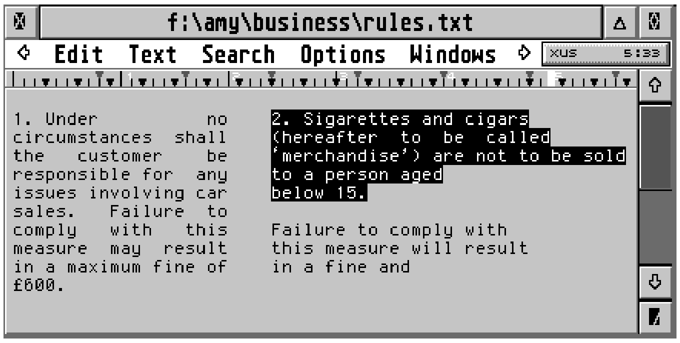
Clear **F10** Removes, among other things,[13] the selection.

Start Selection **F1** Create a selection starting at the text cursor. If there is no previous selection, this selection stretches out until the end of the text.

End Selection **F2** Create a selection ending at the text cursor. If there is no previous selection, it starts at the beginning of the text.

Open Selection **Control** **Space** Creates a selection starting *and* ending at the text cursor. While typing text or moving the cursor, the end of the selection will follow the cursor position, until an *End Selection*, or until an operation is performed upon the selection.

Select SouthEast Quadrant **Shift** **Control** **P** Select the text in the south-east quadrant under the cursor, until there is a line which does not stretch as far as the current cursor column. An example of a south-east quadrant is shown in the picture below.



A SouthEast quadrant

5.2 Cut and Paste

The three most elementary operations on selections and text buffers are *cut*, *copy* and *paste* in the *Edit* menu:

Cut **Control** **X** Copy the selection into the cut/paste text buffer and cut it out of the text.

Copy **Control** **C** Copy the selection into the cut/paste text buffer.

Paste **Control** **V** Insert the cut/paste text buffer into the text as a single fragment (and select the inserted text).

NOTE. In conventional cut/paste based text editors, typing text or selecting *paste* when there is a selection will cause the selected text to be replaced with the new data.[14] The **Edith** selection system is essentially different, in that

1. One can still move the cursor while there is a selection, and text can be typed both inside and outside selected fragments.

2. More than one contiguous fragment can be selected at a time.

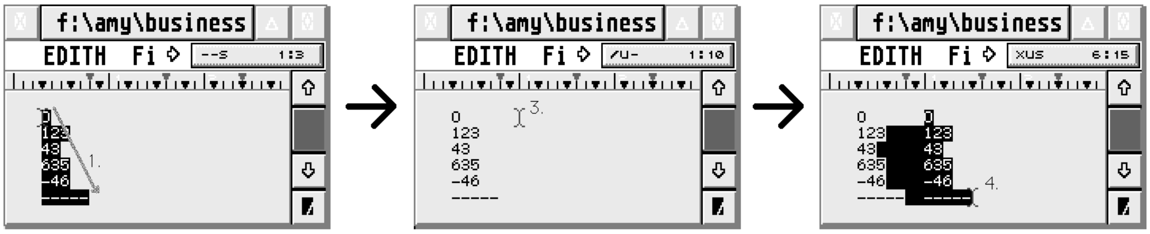
[13] The *repeat* buffer and the *position stack*

[14] In the conventional system, a selection is thought of as a "very wide cursor" dragged over a piece of text. (This conventional method is inconsistent—think of the action of the *backspace* and *delete* keys. For those keys it does matter whether the cursor has width 0 or width $k > 0$).

3. A selected piece of text is never removed when a letter is typed or *paste* is selected.[15]

5.3 Paste vs Overlay

In addition to the normal *paste* option, **Edith** offers the *overlay* function. While *paste* inserts a whole text buffer into the text at a single point, *overlay* inserts it line by line, merging it into the text which is already there. This is especially useful for manipulating columns of data such as numbers, as illustrated in the picture below.



1. Move mouse while holding down right mouse button 2. Copy (Ctrl + C) 3. Move mouse to desired position 4. Overlay (Shft + Ctrl + V)

Using the *overlay* operation.

5.4 Cut vs Delete

In addition to the *cut* operation, selecting *Delete* Shift Control X in the *Edit* menu also removes the selection, but copies the selection into the *delete/textop* buffer instead of into the *cut/paste* text buffer. This allows the user to copy a piece of text, keep it in the cut/paste text buffer while deleting text, and use it later for a *paste* operation.

NOTE. Conventional cut/paste-based editors usually also provide a *delete* operation, but unlike in **Edith**, deleted data is lost forever. In **Edith** the delete/textop buffer can be retrieved from the *Clipboards* dialog (see the section 5.6 below).

5.5 Multiple Selections and Fragmented Buffers

Just like a selection, a text buffer can be divided into a number of fragments. While *paste* and *overlay* always operate on the whole text buffer, other functions take the fragment structure of a text buffer into consideration. When a multiple selection is copied using *Copy* Control C or *Cut* Control X, the copied text in the cut/paste text buffer will have the same fragment structure as the copied selection. In addition to the conventional *paste* operation, **Edith** offers the *Replace* operation (*Search* menu or Control H), which swaps the contents of the text selection and the cut/paste text buffer. If the number of fragments in selection and buffer is equal, selection and text buffer are replaced fragment-wise. If the number of fragments does not match, the whole text buffer is substituted for every fragment in the selection. This *Replace* operation is convenient both for quick manual replacement and automatic multiple search and replace (see page 31, and the picture below).

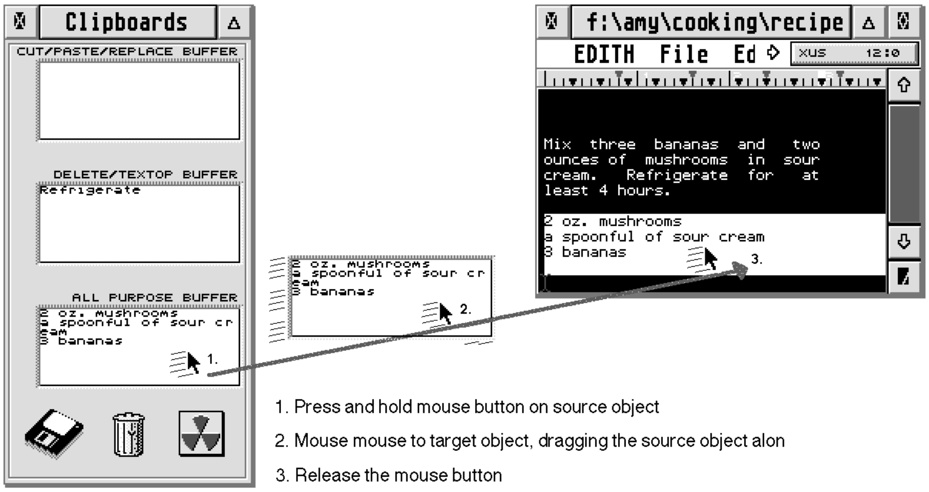
[15]The preset macro Control F4 "`MultiPaste`", attempts to emulate the old-fashioned **Macintosh** style *Paste* operation, where pasting removes all selected text. The macro consists of *Replace* followed by *Copy*.

In a recent discussion on the internet, many **ATARI** developers agreed that the **Macintosh** based system introduces dangerous combinations without much added functionality. For example, it is fairly easy to hit *Select All* by accident and then any single letter will replace the *entire text*. One can safely call the action of replacing 100 lines of text by one letter unrealistic, and any editing system needs to take account of such situations. In **Edith**, we have adopted a rather radical change of perspective.

Manual replacement using multi-fragment selections.

5.6 The Clipboard Panel

The *Clipboard* dialog, available from the *Options* menu or from the keyboard as **Shift** **Control** **B**, allows visual manipulation of the text buffers. It shows the contents of the three text buffers, and three icons: a *disk icon*, the *trash can* and the *shredder*. The contents of each text buffer are displayed in a small font. You can pick up one of the objects in the clipboard dialog by selecting it with the left mouse button, moving it over another object while holding down the left mouse button, and releasing the button—as illustrated in the picture below.



A drag and drop operation.

The following such *drag and drop* operations are available:

Text Buffers can be dragged onto the disk icon (save buffer to a file), the trash can (delete buffer and put it in the trash can), the shredder (remove a text buffer forever), or a text window (paste into text).

the Disk Icon can be dragged onto a text buffer (load buffer), a text (select a file for insertion into the text).

the Trash Can can be dragged onto the shredder (empty trash).

selection After selecting text in a text window, click inside the selection with the left mouse button and hold the button down. A box will appear showing the contents of the selection. While holding down the left mouse button, move the buffer to the clipboard dialog to copy it into one of the text buffer, or move it to another text window to insert the selected text.

When **Control** is held down during drag and drop, the selection is removed after it is copied. When the box is dragged onto the trash can or shredder, it is always removed.

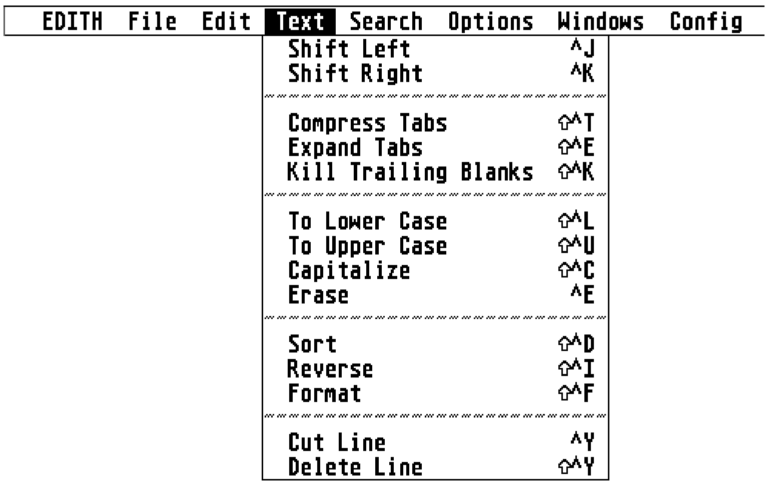
The following *double click* operations are available:

Text Buffers Open text buffer into a window for editing.

Trash Can Open the trash can into a window for retrieving previously discarded text. Note that if you wish old data to be automatically copied to the trash can, you must select this option in the *editor dialog* (see page 43).

The *trash can* icon may be *slim* (the trash can is empty), *fat* (it contains text) or its lid may be open, which means that data has been moved into it a short while ago. While the lid is open, moving additional data to the trash can will take less time than when it is closed. The time it takes for the lid to close can be set in the *File Options* dialog (see page 43).

When the **GEMINI** desktop is visible, buffers and icons can also be dragged onto **GEMINI** icons. The trash can and shredder icons act precisely the same as the icons in the clipboard dialog. Drive and folder icons act as the disk icon in the clipboard dialog, except that the file selector is opened with a preset directory.



The *Text* menu.

6 Text Manipulation

A selection of useful operations for manipulation of text files is found in the *Text* menu. All these operations operate on the text selection. If there is no selection, a default scope is automatically selected. The operations are also valid on multiple fragment selections.

Shift Left [Control] [J] *Default scope: 1 line* Shift the selected text left over one tab distance, or as far as possible. (Depending on the tab settings—see page 43).

Shift Right [Control] [K] *Default scope: 1 line* Shift the selected text right over one tab distance.

[Shift] [Control] [<] *Default scope: 1 line* Shift the selected text left over one column (remove one space character in the indent).

[Shift] [Control] [>] *Default scope: 1 line* Shift the selected text right over one column (add one space character to the indent).

Compress Tabs [Shift] [Control] [T] *Default scope: whole text* Replace as many space characters as possible by tabs (ASCII code 9).

Expand Tabs [Shift] [Control] [E] *Default scope: whole text* Replace all true tabs by sequences of space characters.

Kill Trailing Blanks [Shift] [Control] [K] *Default scope: whole text* Remove all blank space at the end of lines

To Lower Case [Shift] [Control] [L] *Default scope: one word* Replace all letters by their lower case equivalent. E.g., `LaForge` would be converted to `laforge`.

To Upper Case [Shift] [Control] [U] *Default scope: one word* Replace all letters by their upper case equivalent. E.g., `LaForge` would be converted to `LAFORGE`.

Capitalize [Shift] [Control] [C] *Default scope: one word* Capitalize words. E.g., `The king WENT to sPAIN` would become `The King WENT To SPAIN`.

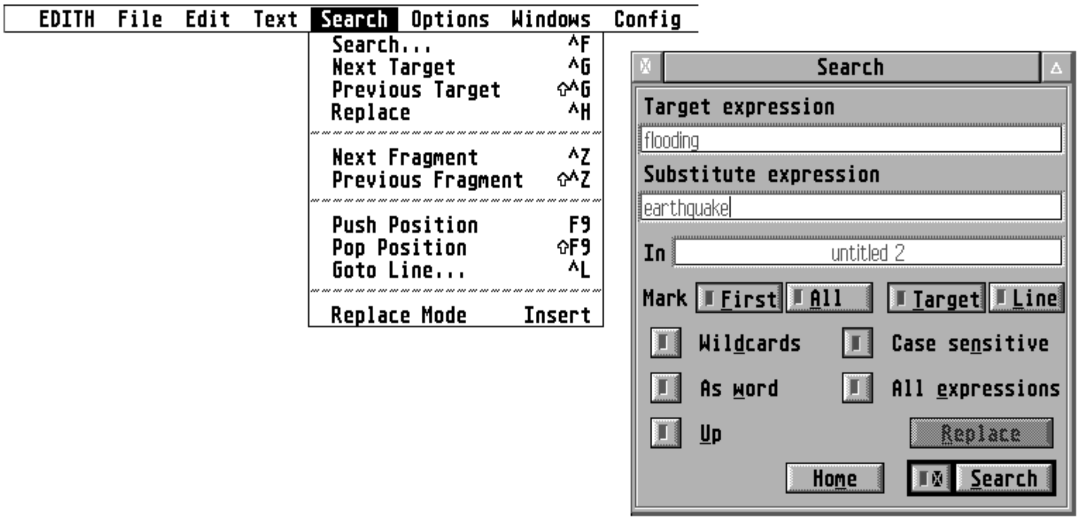
Erase [Control] [E] *Default scope: one word* Fill the selection entirely with space characters, removing any text that was there. This is especially useful for editing text arrays, pictures, or diagrams.

Sort [Shift] [Control] [D] *Default scope: South East Quadrant* Sorts the selected text fragment by fragment, line by line.

Reverse [Shift] [Control] [I] *Default scope: South East Quadrant* Reverses the selected text fragment by fragment, line by line.

Format [Shift] [Control] [F] **or** [F4] *Default scope: South East Quadrant* Formats the selected text, where an empty line is used for paragraph separation. The options for formatting can be set in the *Text Ruler* and *Editor Options* dialogs described in chapter 11.

Undo is valid for all these operations. The original data is always stored in the *Delete/Textop* text buffer, so a buffer copied using *Copy* or *Cut* survives these operations. (See chapter 8). (*Advanced*)



Search menu and search dialog.

7 Searching, Wildcards and Automatic Substitution

The search and replace functions in **Edith** are not like the ones found in most text editors. For example, there is no single operation for combined search and replace. As in the rest of the application, the ability to mark any number of text fragments is exploited in order to get a flexible and unconventionally safe search and replace system.

7.1 Searching

To search in a text window, select *Search...* Control F from the *Search* menu. Type the text to search for in the *Target Expression* field. Select the right parameters for searching (as described below) and click the Search button or press Return. To continue searching, select *Next Target* Control G from the *Search* menu.

The following search options are provided:

First/All When *All* is selected, all occurrences of the target expression in the text are selected; when *First* is selected, only the first target found is selected.

Target/Line When *Line* is selected, **Edith** will select whole lines instead of just the targets.

Wildcards When this option is selected, **Edith** will take wildcards into account (see page 32).

As Word When this option is selected, occurrences of the target expression will only be selected when they are *not* part of a longer word. (e.g., searching for *boat* would detect the occurrence in *the boat over there*, but not in *speed-boat*.

Up When *Up* is selected, successive searching of single targets is done bottom-up, left to right.

Case Sensitive When this switch is *off*, a target will also be selected if part of it is in a different case. E.g., when searching for `boat`, the target `bOaT` will be recognized.

All Expressions **Edith** can store four combinations of a target expression and a substitute expression at once. While only one of these combinations is visible in the search dialog, the others can be accessed through a pull-down menu activated by clicking on the *Target Expression* field with the left mouse button. When the *All Expressions* switch is on, **Edith** will search for all four target expressions at once.

7.2 Substitution

The substitution features offered in the *Search* dialog are no more than an interface to the *Replace* Control H function in the *Search* menu. *Replace* substitutes the contents of the cut/paste text buffer for the text selection. If the number of fragments in the selection is equal to the number of fragments in the text buffer, then the fragments are replaced one by one. If the number of fragments is different, then the whole buffer is substituted for each fragment in the selection. There are two ways to exploit this feature for substitution.

Manual Substitution By hand, replace the first item with the new text. Copy it into the cut/paste text buffer using *Edit/Copy* Control C. Then select all other items, and select *Search/Replace* Control H. This is illustrated in the picture on page 25.

Automatic Substitution Open the *Search* dialog, and enter the target and substitute strings. Select the *All* switch in the search dialog. Select *Search* or press Control G. All targets are selected, and the substitutes have automatically been copied in the cut/paste text buffer. Now select *Replace* or press Control H. The targets are substituted.

NOTE. In conventional editors, it is possible to give a *search and replace by "nothing"* command. In **Edith**, this is not possible; but the same can be achieved by *Search* followed by *Cut*.

7.3 Tagged Wildcard Expressions

(*Advanced*)

For searching targets in a text and matching files in the File Selector, **Edith** provides *tagged wildcard expressions*.[16] [17] These are a logical extension of the **TOS** and **UNIX** wildcard system for files, and strongly resemble common expression styles in **UNIX** programs such as **VI**. *Wildcard expressions* allow you to search for strings which match a general pattern, but which are not literally the same. *Tagging* allows you to use the unknown parts of the matched strings in a substitution.

7.3.1 Target Expressions

Generally, the target expression is a word a sequence of words you are searching for, but with special signs for *unspecified* characters or strings. For example, searching with the expression

```
Hello, *!
```

would yield all strings like

```
Hello, John! Hello, Mary!
```

The 'unspecified string' `*` in the expression `Hello, *!` is called a *wildcard*. If a string fits into an expression pattern we also say that the expression *matches* the string, or the string *matches* the expression. The part of the string corresponding to the wildcards are said to be *matched* by the wildcard. Wildcards allowed in **Edith** target expressions are

`?, !` Match any single character, so `Hello, J?n` will match `Hello, Jen` as well as `Hello, Jan`.

`[` *characters* `]` Matches any tokens in *characters*, where *characters* is a list of single characters or ranges such as `f-r`. I.e. `Hello, J[a-e]n` will match `Hello, Jen` and `Hello, Jan` but not `Hello, Jon`.

`[`^*characters*`]` Matches any tokens *not* in *characters*. So searching for `Hello, J[^a-e]n` will match `Hello, Jon` but not `Hello, Jan` or `Hello, Jon`.

`&, *` Match any string of zero or more characters. So `Jo*n` will match `Jon`, `Joan` and `Jonathan`.

^, `$` Match, respectively, the beginning and the end of the line; `^Hello, John$` will match `Hello, John` if it occupies a whole line, but will not match `Why Hello, John!`.

\(, \) Match zero characters (i.e. these are ignored as far as searching is concerned). A part of the expression between `\(` and `\)` is called a *tagged expression*.

character Matches *character*. A useful option to search for the special characters `?, !, ^, *, &, $, [, ], -` and \ literally. Character should not be a bracket `(` or `)`.

7.3.2 Substitute Expressions

Strings matched by the wildcards in a search expression can be used in the substituted string. The following special characters can be used in the substitute expression:

`?` stands for the *last* character matched by the wildcard `?`.

`!` stands for the last character matched by the wildcard `!` or a range pattern **[(**^**)***characters***]**.

`*` stands for the last string matched by the wildcard `*`.

`&` stands for the last string matched by the wildcard `&` or a *tagged expression*.

[16] Sometimes confused with *regular expressions*, which are more powerful.

[17] The wildcard system for searching in text files is only applied when it is switched on in the *Search* dialog.

Examples of a search and replace action using tagged wildcards.

Target expression:	`\(Hello, *\)!`
Substitute expression:	`Mary saw *. She said: '&'.`
Text:	`Hello, Ann! Hello, Frank!`
Replacement:	`Mary saw Ann. She said: 'Hello, Ann'. Mary saw Frank. She said: 'Hello, Frank'.`

Target 1:	`ex*`
Substitute 1:	`im*`
Target 2:	`e[^x]*`
Substitute 2:	`in!*`
Search Dialog:	*All Expressions* switched on
Text:	`express, export, evade, evolve`
Replacement:	`impress, import, invade, involve`

7.4 The Position Stack

Most text editors offer some kind of facility for *marking* positions in a text, allowing you to memorize the current cursor position, move around in the text, alter some parts elsewhere, and jump back to where you were editing. Since one usually uses these markers to temporarily edit text somewhere else, and then to jump *back* to the original point, **Edith** offers a marking facility in the form of a *stack* of text positions.

When *Push Position* **Shift** **F9** is selected from the *Search* menu, the current cursor position is saved on the stack. This can be done up to four times. When *Pop Position* **F9** is selected, **Edith** will jump back to the *last* position saved, following the *last-in-first-out* or *stack* principle. Repeated use of *Pop Position* will cause **Edith** to *cycle* through the positions which have been saved to the stack. In order to keep use of this function simple, the stack can only contain up to four text positions, which are indicated on the scroll bar by small black triangles. The saved positions are removed by selecting *Clear* **F10** from the *Edit* menu.[18]

The functions *Text Home* **Home**, *Text End* **Shift** **Home**, *Text Middle* **Shift** **Control** **Home** and *Goto Line* **Control** **L** automatically push the original cursor position on the stack, so you never need to be afraid to accidentally hit the **Home** key and forget where you were editing.

7.5 Browsing through the Selection

Much like the *Next Target* **Control** **G** and *Previous Target* **Shift** **Control** **G**, **Edith** offers the operations *Next Fragment* **Control** **Z** and *Previous Fragment* **Shift** **Control** **Z** for browsing through multiple selections. These operations can be useful for browsing through the targets found by a *Search All* operation, or for finding selected text in a large file (note that unlike in conventional editors, **Edith** allows the selection to remain present even while editing text elsewhere; the selection indicator in the text information box (see page 11) only indicates whether there is a selection or not, and not where to find a selection when there is one).

[18] *Clear* also clears the selection and the buffer for the *Repeat* operation.

8 Recovering Lost Information

Most **ATARI** editors only allow recovery of data lost at the very last operation performed. More powerful editors in the **UNIX** world remember long sequences of operations, but recovering information is tedious and these programs consume large amounts of computer memory. **Edith** offers a combination of a one-stage *Undo* operation and a *trash can* for recovering very old data, which is both user-friendly and kind to small computer architectures.

8.1 Undo

The effect of all **Edith** operations[19] can be reversed immediately after they have been performed, by selecting *Undo* from the *Edit* menu (or by pressing the **Undo** key). However, for some operations this depends on the contents of one of the text buffers, and the presence of selections to indicate the position of the original data in the text. So removing a selection or a copy operation in a different text window may cause the *undo* function to fail.

NOTES.

1. While *Undo* mostly applies to a single operation, after typing text (basic line-oriented editing) on one line, *Undo* will restore the line as it was when you started changing it (as opposed to only restoring the very last character typed, word deleted, etc.).
2. Because *Replace* (*Search* menu, **Control** **H**) is a single operation, even multiple search and replace actions can be reversed by pressing **Undo**. Other **ATARI** text editors usually only undo the last substitution made.

8.2 The Trash Can

If you wish to be sure *never* to lose important data, open the *Editor Options* **Control** **4** dialog from the *Options* menu, and switch on *Move old buffers to trash file*. When that option is enabled, *vital data* in the text buffers (e.g. data cut out of a text and never pasted back somewhere else) will automatically be written to the *trash can file* when new data is copied into the buffer. It often happens that you are deleting pieces of text, to discover a minute later that you should not have deleted that information. Usually by that time the data is lost, because the cut/paste text buffer already contains new text. Then simply open the trash can by selecting *Open Trash Can* **Control** **T** from the *Options* menu, or by double clicking on the trash can icon in the clipboards dialog. The text you just deleted will be somewhere in the trash can window. You can now select the text, copy it and paste (or drag and drop) it back into the text.

At regular intervals,[20] the trash can will get full; to empty the trash can, open the *clipboards* dialog **Shift** **Control** **B** from the *Options* menu, and drag the trash can icon onto the shredder. The 'fat' trash can icon will become slim, to indicate that it is empty.

The trash can file is stored in the file `editress.jnk`, either in the directory defined in the environment variable `TRASHDIR` or if this variable is not defined, in the **Edith** system directory. (*Advanced*)

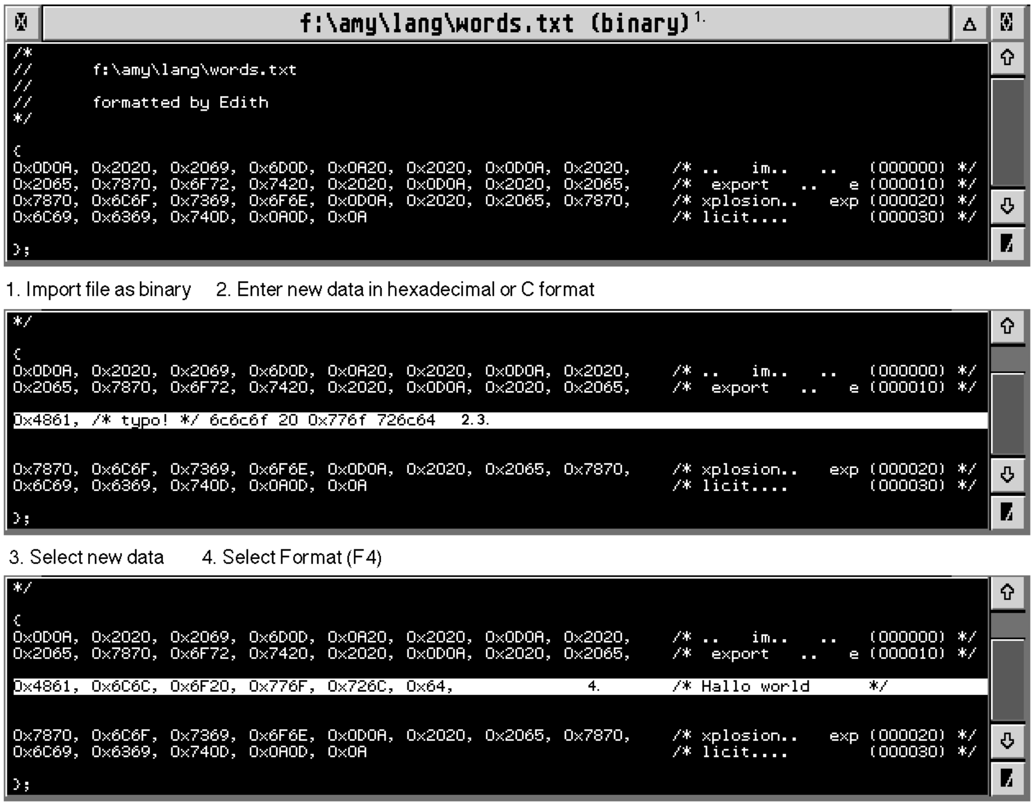
[19] Except substitutions by external TTP filters
[20] But not all that often, experience has shown.

9 Editing Binary Files

Quick editing of binary files can be done directly from within **Edith**. A file can be loaded in binary format by choosing *Import Binary* **Control** **B** from the *File* menu, and selecting a file. (*Advanced*)

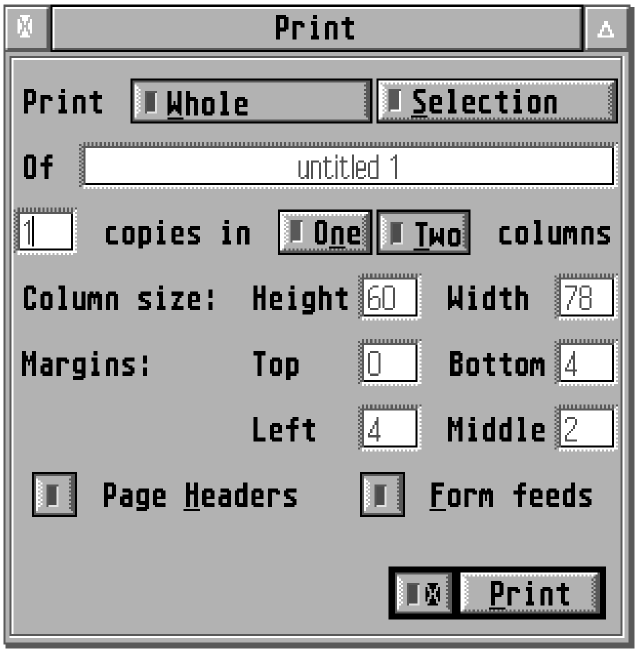
Binary files are displayed in a format close to the **C** programming language. New data can be entered either in **C** format, or by simply entering hexadecimal values. To save a binary file, select *Save As...* **Shift** **Control** **S** from the *File* menu. An alert box will pop up allowing you to choose to save the **C**-representation of the binary (including any text between comments you added during editing), or its binary contents.

Within a binary text window, the *Format* operation (*Text* menu, **Shift** **Control** **F** or **F4**), will reformat the data into aligned **C** format, with the ASCII representation between **C** comments. Format will discard any other **C** comments.



Editing a file in binary mode.

NOTE. When an error is discovered in the **C** syntax, a format or save operation will be aborted. In case of a save operation, this means that the file has *not* been saved. In case of a format operation, a message will appear in your text window. Press **Undo** to recover the old, unformatted text.



The printer dialog.

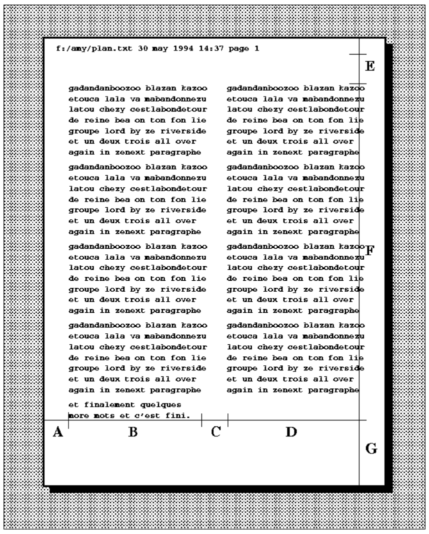
10 Printing Text

Edith has a simple built-in printer facility. It can only send text directly to the printer in ASCII format, but is reasonably configurable and supports two-column printing.

The *Printer* dialog **Control** **P** can be opened from the *File* menu. It allows you to choose a text to be printed, specify a number of copies, one-column or two-column mode, and a number of printing parameters, *viz.* the height and width of the text (or the columns) on a page, the *top margin*, i.e. the number of blank lines written before printing an actual page of text, the *bottom margin*, i.e. the number of blank lines written below a page (except the last page), the *left margin* in columns and the *middle margin* between the two columns in double column printing mode. It also allows you to choose between a real bottom margin and a form feed command. Finally, **Edith** can add a page header line to printer output, specifying file name, date, time and page number.

NOTES.

1. The **Edith** printer routine is not prepared for multi tasking, i.e. you cannot use the editor while printing.
2. Printing is aborted by pressing both **Shift** keys simultaneously.
3. Remember to use a *condensed font* when printing in two-column mode. Otherwise, the right column will probably not be printed at all.



A Left margin

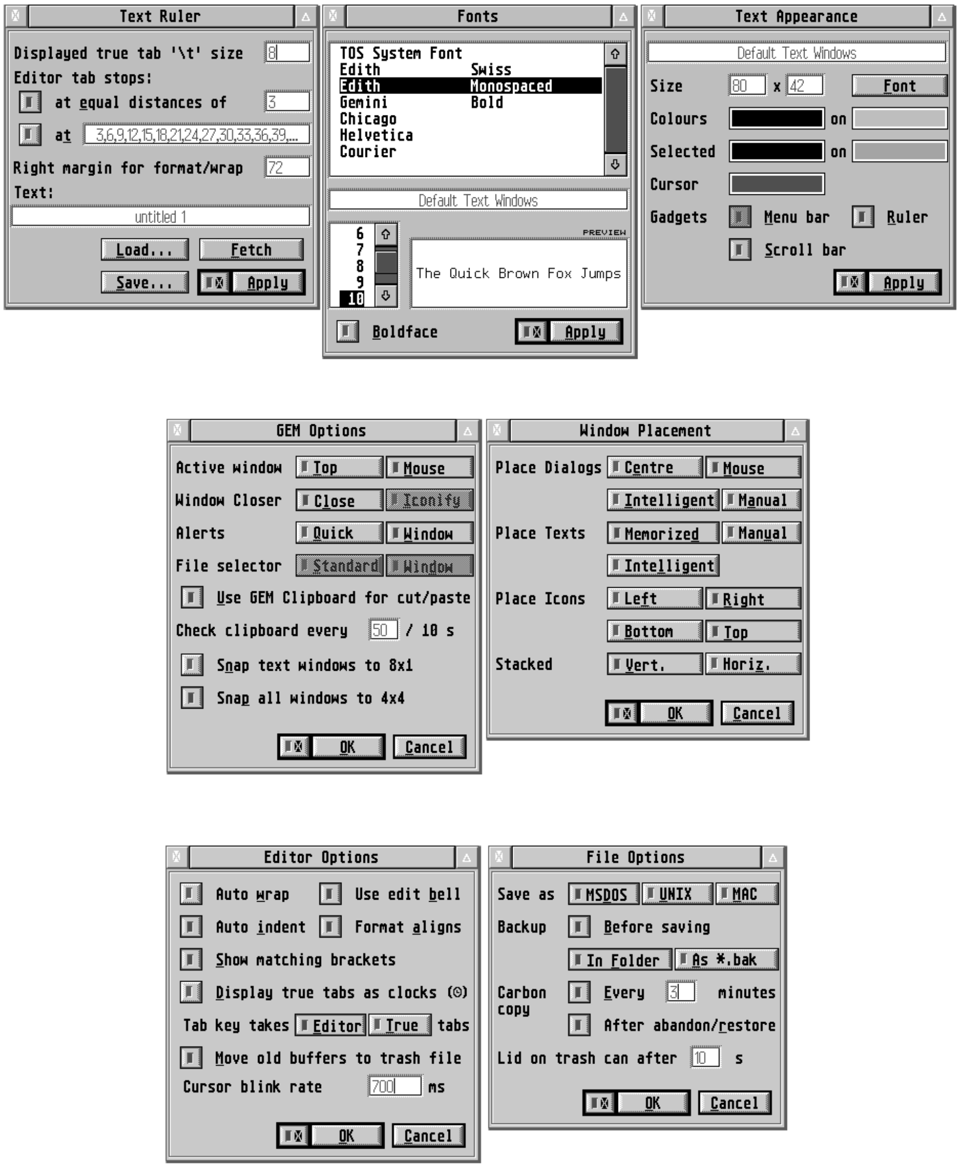
B = D Column width

C Middle margin

E Top margin

F Column height

G Bottom margin (only used when form feed mode is off)



The options dialogs.

11 Making Things Match Your Personal Taste

The *Options* menu gives access to a wealth of parameters to change the behaviour of the editor, the windowing system and the treatment of files.

11.1 Text Rulers

The *Text Rulers* dialog **Shift** **Control** **R** lets you set/copy ruler options for individual text windows and the general settings for the three text window classes *Default*, *Alternative* and *Binary*.

Use the *pull-down menu* in the middle of the dialog to select a text window or default text window class for which to change parameters.

The dialog lets you set the tab stops and the right margin for text formatting. The tab system is explained on page 20.

Click the **Fetch** button to copy a text window's ruler information into the dialog window, or copy the values for a text window class into the dialog. Click **Apply** to copy the dialog window's information into a text window or set the values for a text window class.

Rulers can be loaded or saved by clicking the appropriate buttons and entering a file in the file selector. The files `DEFAULT.RUL`, `ALTER.RUL` and `BINARY.RUL` in the system directory are used for **Edith**'s default setup, and are automatically saved when this is selected in the *At Exit* dialog.

NOTE. This dialog will only let you set the *editor tab stops* at a fixed distance. To set editor tab stops individually, click the left mouse button on the ruler in a text window. Clicking on the ruler with the right mouse button will move the right margin indicator. The **Fetch** button allows you to copy a ruler from a text window into the *Rulers* dialog.

11.2 Fonts

The *Fonts* dialog **Shift** **Control** **H** lets you change 9 sets of font information: six sets for each type of text window, and 3 sets for the appearance of the dialog windows.

A new set is selected using the pull down text display in the middle of the dialog. A font can then be selected by browsing through the font names and sizes using the scroll bars and selecting a name and size. The appearance of the font is shown in the *Preview* field. After selecting font type and size, click **Apply** to set the font.[21] [22]

Texts can be displayed in both monospaced and proportional fonts. When a text is displayed in a proportional font, the character cell width is the width of the lower case character `x`. **Edith** tries to put characters as close as possible to their original position in the text by stretching or shrinking blanks between $\frac{1}{2}$ and $1\frac{1}{2}$ times the width of the `x` character, so that most of the alignment (especially tabs) in a text is respected. Right aligned text however will have a slightly ragged right margin, because words are aligned to the left. (*Advanced*)

11.3 GEM Specific Options

General parameters specific to the **ATARI GEM** system are set in the *GEM Options* dialog **Control** **1**.

- ▹ In standard **ATARI** applications, the *Active Window* is always the *Top* window, i.e. when you type text on the keyboard, it appears in the top window. When the **Mouse** switch is selected instead, text will be inserted in the window right below the mouse pointer.[23]
- ▹ Normally, when you click the closer button of a window, it is closed. When *Window closer: Iconify* is selected, it is iconified instead. Holding down **Shift** while clicking the closer will then close the window.

[21] Please remember that before changing to a different font set, *Apply* must be clicked or else the font information in the dialog will be lost.

[22] When you are using a system font with an alternative size (as possible in e.g. **Geneva**), you will have to change the Edith standard font manually.

[23] When **WINX** or **MultiTOS** is used, applications which understand the **Venus-Accessory Protocol** (see the **GEMINI** documentation) may even respond to key input in background windows when an **Edith** window is on top. However, this may also cause uncautious programs to crash.

- **Edith**'s *alert boxes* are normally presented in a window, so that other applications or accessories can still be used when an Edith alert is on the screen. The **Quick** switch brings back the old type of alert boxes.[24] The *Alert Bell* switch determines whether **Edith** should sound the bell when an alert pops up.
- **Edith**'s *file selector* normally makes use of a window. When the **Standard** switch is selected, the standard **TOS** item selector is used instead. This option is *not* available when **MultiTOS** is being used.
- When the *Use GEM clipboard as cut/paste buffer* switch is on, the **GEM** clipboard is used as the *cut/paste text buffer*. This means that you can cut and paste pieces of text to and from other applications.
- Finally, windows can be *snapped* for extra display performance and to avoid distorted dot patterns when moving windows which contain a scroll bar. The patterns in the scroll bar will normally be distorted when you move a window. This is a problem of **ATARI GEM** which can also be solved by using **WINX** by Martin Osieka.[25]

11.4 Window Placement

The way windows and icons are placed can be configured in the *Window Placement* dialog **Control** **2**.

For *windows* the following options are provided: (separate selections of these options are available for text windows and dialog windows)

Center windows are always put in the middle of the screen.

Mouse windows are put at such a position that the mouse pointer is inside a window when it is opened.

Manual every time a window is opened, the user can place it anywhere on the screen by moving a rubber box contour of the window. The window is opened after the left mouse button or the **Return** key is pressed. When the **Undo** key, or the *right* mouse button is pressed instead, the window is not opened.

Intelligent The window is aligned to the other windows at a place on the screen where it covers the least amount of space in other **Edith** windows.

Icons are always placed automatically starting in a corner of the screen, going either in vertical or horizontal direction. Both the corner and the direction can be set in this dialog.

11.5 Text Appearance

The appearance of text windows (except for fonts) is set in the *Text Appearance* dialog **Control** **3**.

Use the pull-down field at the top of the dialog to choose the text window class for which you wish to change the appearance parameters. The parameters cannot be set for text windows individually.

The text colours can be selected through pull-down menus activated by clicking on the coloured rectangles.

The *ruler* and *scroll bar* can be switched off independently for all text window classes. The *menu bar* can *not* be switched off when **MultiTOS** or the desk accessory version of **Edith** is used.

The **Font** button provides quick access to the *Fonts* dialog.

NOTES.

1. Remember to select **Apply** before changing to a different text window class through the pull-down menu, if you want your changes to be preserved.
2. The default colour and dimension parameters are saved into separate configuration files for each number of colours (2, 4, 16 and 256) and each resolution.

[24] Quick alert boxes do not react to keyboard shortcuts except **Return**.

[25] This program is available on most **FTP** sites and bulletin board systems.

11.6 Editor Options

The *Editor Options* dialog **Control** **4** contains switches for miscellaneous text editor parameters.

Auto Wrap Select this option when you are editing plain text (and better not when writing a computer program). When this option is active, words are automatically moved to the next line when they exceed the right margin. *Auto Wrap* can be quickly switched on and off using the function key **F5**.

Edit bell When this option is selected, actions which do not have a clear visible effect will make the system bell sound.

Auto indent Especially useful for programming. when this option is active **Edith** will automatically indent new lines precisely as much as the previous line. This option also changes the behaviour of the **Escape** key in text windows. When the *Auto Indent* function is on, **Escape** will cause the cursor to jump back to the previous level of indentation. When *Auto Indent* is off, **Escape** will cause the cursor to jump back all the way to the beginning of the line. Auto indent can be quickly switched on and off using the function key **F3**.

Count as Indent Here you can specify a up to 10 special characters (such as the comment characters in a programming language, e.g. `#` and `/*`) which are to be considered as indentation for purposes of auto-indent, back indent **Escape** and text formatting.

Format aligns When this option is selected, the *Format* command (*Text* menu, **Shift** **Control** **F** or **F4**) will align paragraphs to the right margin by adding extra space between words.

Show matching brackets Like in the **UNIX** editor **VI**, when this option is selected, and a closing bracket `)`, `}` or `]` is typed, the cursor will jump back to the corresponding opening bracket, or the bell will sound when the bracketing is inconsistent.

Display true tabs as True tab characters (ASCII code 9) can either be displayed as a white space stretching out until the next true tabulator stop (the upper row of triangles on the text ruler), or as a clock symbol, as some older **ATARI** editors do.

Tab key generates This option is the core of **Edith**'s *Tab/True Tab* system. In most editors, the tab key simply inserts a true tab (ASCII code 9) into the text. This is the case when *Tab key generates true tabs* is selected.

When *Tab key generates editor tabs* is selected, the tab key will generate a number of blanks (ASCII code 32), stretching out until the next *editor* tab stop. See page 20 for more information about tabs.

Move old buffers to trash can file When this option is selected, data in a text buffer is automatically saved to the *text data trash can* when it is *vital* and is about to get overwritten. **Edith** has a quasi-intelligent way of deciding whether data is vital, based on the likelihood that the text in a buffer is still available somewhere else. For more information see page 35.

Cursor blink rate The rate at which the text cursors blink can be set in milliseconds. If the number of milliseconds is set to zero, the cursor will not blink at all.

11.7 File Options

Some miscellaneous parameters for saving files can be set in the *File Options* dialog **Control** **5**.

▷ When a file is saved, the line transitions can be converted to one of following conventions:[26] (*Advanced*)

MS-DOS	carriage return, line feed (**ATARI** standard)
UNIX	line feed
Macintosh	carriage return

[26] All of these conventions are recognized when a text is loaded.

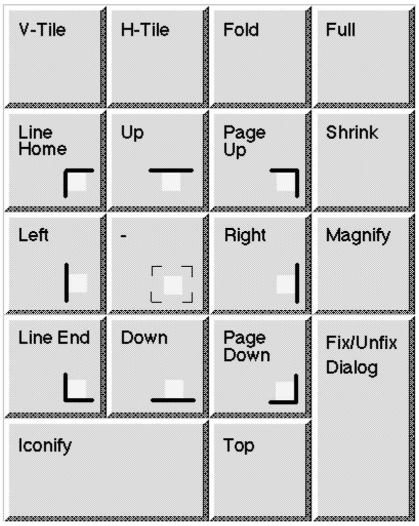
If however, the *As Is* switch is selected, newlines remain as they are. This also makes saving faster.

- ▷ You can choose to write a *Backup* file before saving any file. The backup file either has the same name as the original file, with the extension '`.BAK`', or is saved in a folder `BACKUP` by its original name.
- ▷ You can choose for *Carbon Copy files* to be written automatically at regular intervals, set in minutes. They can also be written when you abandon the contents of a window when you close or restore it. Carbon copies are a safety measure. Suppose for some reason your computer system has crashed while you were editing important text. If a carbon copy of that text has been written recently, and the text is opened again later, **Edith** will automatically offer to use the carbon copy instead of the original file. Also, when you decided to abandon a text and later found the changes valuable anyway, the carbon copy can provide a solution.

11.8 Interface Properties

The *Interface Properties* dialog **Control** **7** provides switches for some miscellaneous keyboard, mouse and dialog appearance options.

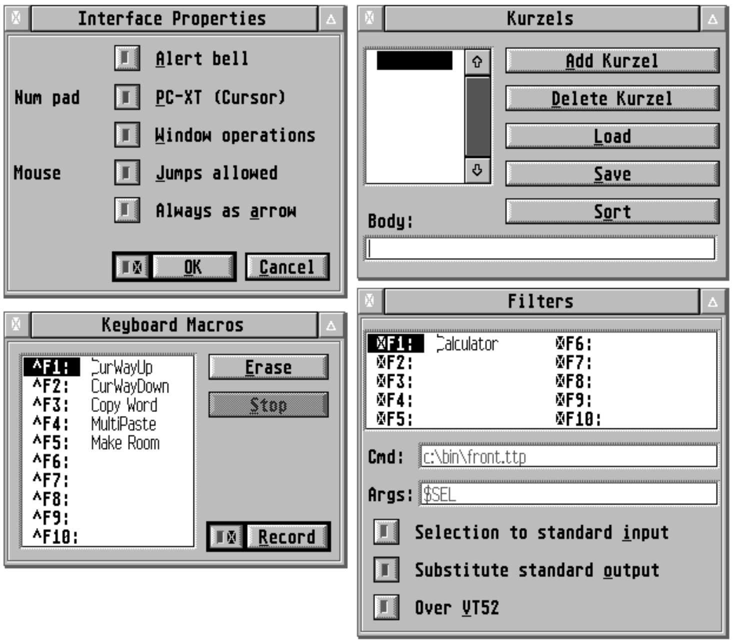
- ▷ When *Alert Bell* is selected, the bell sounds whenever one of **Edith**'s alert windows pops up.
- ▷ The darkness of **Edith**'s buttons can be set to one of three levels (only valid for resolutions with more than 16 colours).
- ▷ When *Num Pad: PC-XT* is selected, the digit keys on the numerical keypad behave like they do on an IBM keyboard when *num lock* is switched off. (They move the cursor.)
- ▷ When *Num Pad: Window Options* is selected, the non-digit keys on the numerical keypad are assigned to window operations. Some of these operations are in the *Windows* menu. The picture below shows the operations assigned to the numerical key pad when both the *PC-XT* and Window Operations switches are **ON**.



The lay-out of the Numerical Keypad when both *PC-XT* and *Window Operations* are switched on.

- ▷ If *Window Operations* is selected, combining **Control** and a digit on the numerical keypad moves a window to a corner (1, 3, 7, 9), the edge (2, 4, 6, 8) or the middle (5) of the screen.

- ▹ When *Mouse jumps allowed* is selected, Edith will try to automatically move the mouse around so that the user need not get his hands off the keyboard. This option is not available under **MultiTOS** and **Geneva**.
- ▹ When *Mouse: Always as Arrow* is selected, Edith will only use the arrow and busy bee mouse forms. Also, the mouse will never remain hidden when it is inside a text window.



The keyboard options dialogs

12 Advanced Keyboard Options

The upper half of the *Config* menu gives access to a number of dialog panels for advanced keyboard configuration options.

12.1 Macros

A *keyboard macro* is a sequence of key strokes recorded like you record music on a cassette tape. **Edith** allows you to record up to 10 such sequences, which are activated by pressing **Control** and a function key.

To start recording a macro, open the *Macros* dialog **Control** **6** from the *Config* menu. Then select one of the function keys, type a name to identify the sequence (this is not necessary), and press **Record**.

Top a text window and enter the sequence. To stop recording, select *End Macro* **Control** **6** from the *Config* menu, or click **Stop** in the macro dialog. While recording, the mouse will take the shape of an audio cassette. From then on, you can repeat the sequence by pressing the selected function key, while holding down **Control**.

NOTES.

1. A macro consists of up to 16 keystrokes. A macro cannot call another macro.
2. A macro will stop playing when the **Control** key is released.
3. Macros are loaded and saved together with other keyboard settings by *Load Keys* and *Save Keys*.
4. Out of the box, **Edith** comes with a number of preset macros:
   (a) `CurWayUp` Moves the cursor way up, until you release **Control**.
   (b) `CurWayDn` Moves the cursor way down.
   (c) `Copy Word` Selects and copies the word the cursor is inside of.
   (d) `MultiPaste` An emulation of **Macintosh** type paste. Consists of *Replace* followed by *Copy*.
   (e) `Make Room` Inserts a number of blank lines, and then moves the cursor back up to where it was.

12.2 Kurzels

Kurzels were introduced in Tom Quellenberg's Shareware Text Editor **QED**[27], and have since be adopted in other **ATARI** based packages. A kurzel consists of an abbreviation and a body. When the abbreviation is typed in a text window, followed by **Escape**, it is expanded into the full body. If the special character ˜ appears in the body, this is where the cursor will be placed afterwards.

Examples

1. Business.

   kurzel: `us`

   body: `Johnson, Blake and Hammersmith Ltd.`

   Here, a kurzel eliminates the need to remember your company's name. Simply type `us` followed by **Escape** to insert your business name into the text.

2. The **C** programming language.

   kurzel: `main`

   body: `void main(int argc, const char* argv[])`

[27] The shareware package **QED** is available on many **FTP** sites and bulletin board systems.

kurzel: `itoa`

body: `itoa(~, s, 10)`

Show how a kurzel serves to remember things a programmer should not need to worry about.

3. The LaTeX document formatting language.

   kurzel: `ds`

   body: `\documentstyle[din_a4, times, epsf]{~}`

   With this kurzel definition at hand, instead of having to type the whole LaTeX header

   ```
   \documentstyle[din_a4, times, epsf]{article}
   ```

   the user will only have to type

   ```
   ds <ESCAPE> article
   ```

Editing Kurzels

Open the *Kurzels* dialog **Control** **8**. Select an abbreviation using the mouse and the scroll bar, or create a new abbreviation using the **Add Kurzel** button. Then type the body in the field at the bottom of the dialog.

NOTES.

1. An abbreviation can be at most 8 characters long.
2. Click on the **Sort** button after adding a number of kurzels. The abbreviations will then be sorted alphabetically. It is then wise to save the kurzels to a file by clicking the **Save** button and typing a file name. The kurzels in the file `DEFAULT.KRZ` will be automatically installed the next time **Edith** is started.
3. The **Escape** key also has the *back indent* function, when what is directly left to the cursor is white space. See page 20. This means that a kurzel cannot end in a blank (ASCII code 32).
4. The file format for the **Load** and **Save** options in the kurzel dialog is identical to that of later versions of the editor package **QED**.
5. To erase all kurzels, click **Load**, and type the name of a file that does not exist.
6. The **Edith** disk contains a number of example kurzel files. One for LaTeX, and one for the **DeskJet** printer.

12.3 Filters

Filters are an experimental feature of **Edith**, and are not quite in concordance with **ZFC**'s user-friendliness guidelines. This is subject to improvement in later versions.

A *filter* is a text based program (a **TOS** or **TTP** program) called by **Edith**. Up to 10 filters can be installed; they are activated by pressing **Alternate** and a function key from inside a text window.

Two filters are automatically installed for you.

- ▷ The `front` filter, installed on **Alternate** **F1**, is a calculator. When you select an numerical expression, for example

  ```
  (20 + 40) ^ 2
  ```

and press **Alternate** **F1**, a window pops up with the result of the expression (3600). The `front` program is included on the **Edith** disk with the kind permission of its author **Eero Tamminen**. It is freeware. Please read the rules for copying and distribution in the file `A:\FRONT\FRONT.MAN` on the disk.

▷ The `quote` filter inserts a prefix before each line in the selection. It can be used for answering electronic mail. For example, when the following text is selected:

```
Hello Fred,

How are you?

John.
```

and you hit **Alternate** **F2**, the selection is replaced with

```
> Hello Fred,
>
> How are you?
>
> John.
```

▷ The `colsort` filter sorts lines based on their contents starting at a given column. In the default configuration, **Alt** plus **3**, **4** or **5** activate different forms of column sorting (standard, reverse and by numerical value). The column position for sorting is indicated by placing the cursor before activating the filter.[28]

▷ The `textinfo` **Alt** **F6** filter gives information about the selected text. This information includes things like the number of lines, words and bytes, the type of newlines, and the general type of the file.

▷ The `spell` **Alt** **F7** filter checks the words within the selected text with a *dictionary file*. This is by default the supplied "toy dictionary of English". Any words in your text which are *not* in the dictionary are listed in a new window. This is a rudimentary form of spelling checking which is very popular in the **UNIX** world. It is often much more convenient than interactive spell checking.

▷ The `rmdup` **Alt** **F8** filter removes duplicate lines in the selected text.

It requires a little experience to install your own programs as a **TTP** filter.

To install a filter, open the *Filters* dialog **Control** **9** from the *Config* menu, select a function key, and enter a name to identify its use (this is not strictly necessary). Then click on the *Cmd* field, and select the filter program in the file selector. Type its arguments in the *Args* field. (*Advanced*)

The command line for filters can contain environment variables, plus:

`$IN`	The name of a file containing the selection
`$OUT`	A file whose contents will be substituted for the selection
`$INOUT`	A file which will serve as `$IN` and `$OUT`
`$ENTER`	A box pops up; what you type in the box is substituted for this variable
`$SEL`	The first line of the selection, literally

Examples:

```
command: banner.ttp
arguments:  $ENTER
substitute stdout: yes
```

Makes a banner after the user typed text into a dialog box.

[28] Remember here that in **Edith**, you can move the cursor inside a selected text block.

```
command: banner.ttp
arguments: $SEL
substitute stdout: yes
```

Replaces the first words in a selection by a ‘bannered’ version.

```
command: vi.ttp
arguments: $INOUT
over text terminal: yes
```

Opens the **VI** text editor on the text buffer, and substitutes the changes made by **VI**.

```
command: date.ttp
substitute stdout: yes
```

Inserts the date and time into the text.

When a filter is run over a text terminal, its output may flash over the screen too quickly. Hold down a **Shift**, **Alternate** or **Control** key to hold the VT-52 terminal screen after the filter has terminated.

13 Loading and Saving your Preferences

All **Edith** behaviour parameters as described in the previous chapters can be saved in files. As many **ATARI** owners do not always work in the same screen resolution, and because **Edith** has so many configuration parameters, the parameters have been divided into small groups.

Loading and saving parameters is done through the entries in the lower half of the *Config* menu. Parameter files can be anywhere on your disk, but are normally put in the **Edith** *system directory*. Parameter files can have any name, but one should be aware that some names are used to designate the *default* configuration which is automatically set when **Edith** is started.

NOTE. After installation by `INSTALL.PRG`, there are no configuration files. **Edith** will use built-in default configurations if some of the configuration files are missing.

Auto save at exit

When the application `EDITRESS.APP` is left, the configuration files, or a limited selection of these, can be automatically saved. The settings for this automatic save option are set in the *At Exit* dialog **Shift** **Control** **Q**. This dialog also allows you to save that selection of configuration files straight away, which is convenient if you are using the desk accessory version.

Configuration Groups

The preferences are stored in the following six groups of configuration files.

Preferences Printer, file selector, search, window options, window placement, editor options, file options, at exit parameters, mouse options. Loaded and saved through the entries in the *Config* menu. Default configuration file is `DEFAULT.PRF`.

Dimensions Window dimensions, text window components, fonts, text window positions. Loaded and saved through the *Config* menu. Default configuration file hv`.DIM` where h is the number of horizontal pixels and v is the number of vertical pixels. E.g., the dimensions file for ST high resolution is `640400.DIM`.

Colours Text window and cursor colour. Loaded and saved through the *Config* menu. Default configuration file is `DEF`n`.COL` where n is the number of colours. For example, the default colours file for ST low and TT medium resolutions is `DEF16.COL`. Note that a maximum of 256 colours is supported. **Edith** never actually changes the **VDI** palette, so using other colours than the first 16 may result in surprises when **Edith** is used in combination with other applications.

Keyboard General keyboard parameters, filters, macros. Loaded and saved through the *Config* menu. Default configuration file is `DEFAULT.KEY`.

Rulers Editor tabs, true tab distance and right margin. Loaded/Saved from the *Rulers* dialog panel. Default files are `DEFAULT.RUL`, `ALTER.RUL` and `BINARY.RUL`.

Kurzels Loaded and saved from the *Kurzel* dialog. Default configuration in `DEFAULT.KRZ`.

A Menu Glossary

A.1 EDITH menu

About Edith... Show information about **Edith**.

Desk Accessories The rest of this menu provides access to desk accessories and other applications under multitasking systems. The `EDITH` menu in text windows only contains the *About Edith* entry.

A.2 File menu

New Control N Open an untitled, empty text window. Add the Alternate key for an alternative window.

Open... Control O Open **(page 16)** the file selector **(page 15)** for a new text. Add Alternate for an alternative window.

Open Next Shift Control N Open the next file in the input queue **(page 17)** in a default text window.

Restore Control Undo After your confirmation, reload the text as it was last saved. A carbon copy of the text as it was before reloading is written when *carbon copy before abandon* is set in the *File Options* dialog **(page 43)**.

Insert... Control Insert Open the file selector **(page 15)** for a buffer to insert into the text.

Close Control W Close window. If changes have been made since the text was last read from/written to a file, **Edith** will ask you to choose one of the following options: *Abandon* Alternate A, *Save* Alternate S and *Save As*... Alternate V. When a text is *Abandon*ed, **Edith** can automatically write a *carbon copy* of the text **(page 43)**.

Save Control S Save the text. Only available if the text has a valid file name (e.g. not `untitled 1`), and it is not read-only.

Save As... Shift Control S Open the file selector for a file name. Then save the text under that name. After saving, **Edith** will ask the user whether the window should go on existing carrying the new or the old name.

Save Selection... Control D Open the file selector **(page 15)** for a file name, then save the selected text under that name.

Import Binary Control B Open the file selector for a file opened in binary editor mode. See page 37.

Open Project Shift Control M Open the file selector for a project file to be executed. See page 18.

Print... Control P Open the *Print* dialog. **(page 39)**.

Quit Control Q Quit. Only valid when **Edith** is used as a **GEM** application. Warning: **Edith** does not ask for a confirmation (but will ask you what to do with the files that have been changed).

A.3 Edit menu

Undo Undo Undo the changes made by the last operation. **(page 35)**.

Repeat Control . Repeat the last sequence of line-editing operations performed on one line. **(page 20)**.

Clear F10 Clear the text selection, the position stack and the buffer for the *Repeat* command.

Cut Control X Remove the selected text after copying it into the cut/paste text buffer.

Copy Control C Copy the selected text into the cut/paste text buffer.

Paste Control V Insert the cut/paste text buffer into the text at the point indicated by the cursor. Unlike in most other editors, the selected text will not be replaced with the pasted text.

Delete Shift Control X Remove the selected text after copying it into the delete/textop buffer.

Overlay Shift Control V Insert the cut/paste buffer into the text line by line, left-aligned, merging it into the present text.

Select All Control A Select the entire text

Invert Selection Control I Select precisely those parts of the text which were *not* selected

Start Selection F1 Create a contiguous selection starting at the cursor position.

End Selection F2 Create a contiguous selection ending at the cursor position.

Open selection Control Space Create an *empty* selection, whose end point travels along with cursor as you move it down in the text, or when you are typing, until you select *End Selection* F2 or perform an operation on the selection.

Select SouthEast Quadrant Shift Control P Select the paragraph immediately right and below the cursor. See page 24.

A.4 Text menu

Shift Left Control J *Default scope: 1 line* Shift the selected text left over one tab distance, or as far as possible. (Depending on the tab settings—see page 43).

Shift Right Control K *Default scope: 1 line* Shift the selected text right over one tab distance.

Shift Control < *Default scope: 1 line* Shift the selected text left over one column (remove one space character in the indent).

Shift Control > *Default scope: 1 line* Shift the selected text right over one column (add one space character to the indent).

Compress Tabs Shift Control T *Default scope: whole text* Replace as many space characters as possible by tabs (ASCII code 9).

Expand Tabs Shift Control E *Default scope: whole text* Replace all true tabs by sequences of space characters.

Kill Trailing Blanks Shift Control K *Default scope: whole text* Remove all blank space at the end of lines

To Lower Case Shift Control L *Default scope: one word* Replace all letters by their lower case equivalent. E.g., `LaForge` would be converted to `laforge`.

To Upper Case Shift Control U *Default scope: one word* Replace all letters by their upper case equivalent. E.g., `LaForge` would be converted to `LAFORGE`.

Capitalize Shift Control C *Default scope: one word* Capitalize words. E.g., `The king WENT to sPAIN` would become `The King WENT To SPAIN`.

Erase Control E *Default scope: one word* Fill the selection entirely with space characters, removing any text that was there. This is especially useful for editing text arrays, pictures, or diagrams.

Sort Shift Control D *Default scope: South East Quadrant* Sorts the selected text fragment by fragment, line by line.

Reverse Shift Control I *Default scope: South East Quadrant* Reverses the selected text fragment by fragment, line by line.

Format Shift Control F **or** F4 *Default scope: South East Quadrant* Formats the selected text, where an empty line is used for paragraph separation. The options for formatting can be set in the *Text Ruler* and *Editor Options* dialogs described in chapter 11.

Cut Line Control Y remove the current line after copying it to the cut/paste text buffer.

Delete Line Shift Control Y or F8 or Shift Delete remove the current line after copying it to the delete/textop buffer.

A.5 Search menu

Search... Control F Open the search/replace dialog **(page 31)**

Next Target Control G Search again (search next item in *mark first* mode).

Previous Target Shift Control G Seach again, but in the opposite direction to what's set in the *Search* dialog **(page 31)**.

Replace Control H Swap the selection and the cut/paste text buffer. If the number of fragments in the selection and the buffer are equal, substitution is fragmentwise; otherwise, the entire contents of the buffer are substituted for each fragment in the selection.

Next Fragment Control Z Move the cursor to the beginning of next fragment of the text selection

Previous Fragment Shift Control Z Move the cursor to beginning of the previous fragment of the selection.

Push Position Shift F9 Push the current cursor position onto the position stack. **(page 33)**.

Pop Position F9 Pop a pair of co-ordinates from the position stack and move the cursor to that point. **(page 33)**

Goto Line... Control L Enter a line number, then move the cursor to the beginning of that line.

Replace mode Insert Toggle between insert and replace mode.

A.6 Options menu

Clipboards... Shift Control B Open the clipboard panel. **(page 26)**

ASCII Table... Shift Control A Open the ASCII table panel. **(page 21)**

Text Ruler... Shift Control R Open the text rulers dialog. **(page 41)** If opened from a text window's menu, copy its ruler parameters into the dialog.

Fonts Shift Control H Open the font dialog. **(page 41)**

GEM Options... Control 1 Open the GEM specific options dialog. **(page 41)**

Window Placement... Control 2 Open the window placement dialog. **(page 42)**

Text Appearance... Control 3 Open the text appearance dialog. **(page 42)**

Editor Options... Control 4 Open the editor options dialog. **(page 43)**

File Options... Control 5 Open the file options dialog. **(page 43)**

At Exit. . . Shift Control Q Open the at exit dialog(page 51)

Interface. . . Control 7 Open the *Interface Properties* dialog.(page 44)

Help Help Open a help page with information about the active window.

Note Pad Shift Control N Open the note pad window.

Open Trash Can Control T Open the trash can into a text window.(page 35)

Fix Dialog *num* Enter Fix or unfix a dialog.(page 12).

A.7 Windows menu

The keyboard shortcuts on the numerical keypad can only be used when these are switched on in the *Interface Properties* dialog. See page 44.

Full/Old Size *num* * Bring window to its full size or when it already covers the whole screen, back to its original size.

Iconify *num* 0 Replace a window by its icon.

Top/Bottom *num* . Top a window, or bring it to the bottom when it is already on top. Useful as a keyboard shortcut.

Close Top Control U Close the top window. *Note* this is not always the same as *Close* as the latter will close the *active window*.

Close All Shift Control W Close all windows.

Cycle Texts Shift Tab Top the next text window.

Cycle Dialogs Control Tab Top the next dialog window.

V-Tile *num* (Arrange the text windows in such a way that they fit exactly on the whole desktop, vertically stacked.

H-Tile *num*) Arrange the text windows in such a way that they fit exactly on the whole desktop, next to eachother.

Fold *num* / Fold all other text windows below the active text window.

Smaller Font *num* – Zoom out on the text by taking a smaller size of the same font.

Larger Font *num* + Zoom in on the text by taking a larger size of the same font.

Resize. . . Control R Enter a new size for a text window. Examples of patterns allowed are: `80x40`, `60x`, `x30`. When you just type nothing and immediately hit Return or click OK, the default size for the window class is set.

A.8 Config menu

Rec Macro. . . Control 6 Open the *Keyboard Macros* dialog.**(page 47)**.

End Macro Stop macro recording.

Kurzels. . . Control 8 Open the *Kurzels* dialog.**(page 47)**

Filters. . . Control 9 Open the *Filters* dialog.**(page 48)**

Load Preferences Open the file selector to load a general **Edith** preferences file.**(page 51)**

Load Dimensions Open the file selector to load a text dimensions parameter file.**(page 51)**

Load Colours Open the file selector to load a text colours parameter file.**(page 51)**

Load Keys Open the file selector to load a keyboard parameter file.**(page 51)**

Save Preferences Open the file selector to save a general **Edith** preferences file.**(page 51)**

Save Dimensions Open the file selector to save a text dimensions parameter file.**(page 51)**

Save Colours Open the file selector to save a text colours parameter file.**(page 51)**

Save Keys Open the file selector to save a keyboard parameter file.**(page 51)**

B Keyboard Glossary

A complete list of all keyboard combinations recognized by **Edith**, with the following annotations: a combination marked with (∗) is only recognized when *Window Operations* is switched on in the *Interface Properties* dialog (page 44); a combination marked (†) is only recognized when *PC-XT* is switched on.

F1 *Edit* menu, *Start Selection*

F2 *Edit* menu, *End Selection*

F3 Toggle *auto indent*(page 43) on/off.

F4 *Text* menu, *Format*. Also Shift Control F.

F5 Toggle *auto wrap*(page 43) on/off

F6 *Options* menu, *ASCII Table*...

F7 *Search* menu, *Next Target*

F8 *Text* menu, *Delete Line*. Also Shift Control Y and Shift Delete

F9 *Search* menu, *Pop Position*

Shift F9 *Search* menu, *Push Position*

F10 *Edit* menu, *Clear*

Escape Delete white space to the left of the cursor as far as possible until the beginning of the line (when *Auto Indent* is off), or until the previous level of indentation in the text, calculated from the lines above (when *Auto Indent* is on). When instead of white space there is a kurzel (abbreviation) is directly left to the cursor, expand it. See pages 43 and 47.

Shift Escape Insert an escape character (ASCII code 27) into the text.

Tab Insert a tab or a number of spaces into a text, depending on the settings for the Tab key in the *Editor Options* dialog Control 4. See pages 20 and 41.

Shift Tab *Windows* menu, *Cycle Texts*

Control Tab *Windows* menu, *Cycle Dialogs*

Alternate Control Tab Switch to another application under **MultiTOS** or **Mag!X**.

Shift Control " A dead key(page 21).

Control , Dead key.

Control ‘ Dead key.

Control \ Dead key.

Control — Dead key.

Control | Dead key.

Control ˜ Dead key.

Control **ˆ** Dead key.

Control **.** *Edit* menu, *Repeat*

Shift **Control** **<** Shift left line or selection over one column

Shift **Control** **>** Shift right line or selection over one column (on some keyboards, where $<$ and $>$ are on the same key, **Shift** must be omitted)

Control **Space** *Edit* menu, *Open Selection*

Backspace Remove one character to the left of the cursor, or when there is no such character, append the current line to the one immediately above.

Shift **Backspace** Remove everything left to the cursor

Control **Backspace** Remove the word or inter-word space left to the cursor

Delete Remove one character right to the cursor. If there is no such character, append the next line to the current one.

Shift **Delete** Delete line. Also **F8** and **Shift** **Control** **Y**

Control **Delete** Delete the word or inter-word space right of the cursor

Alt **Control** **Delete** Reboot your computer. Only available on **TOS** versions 1.4 and higher.

Return Insert a new line, and move everything right of the cursor along.

Shift **Shift** Break. Aborts any of the following operations: printing, searching, overlay, repeat, and the operations in the *text* menu. Macro play is aborted when the **Control** key is released.

Help *Options* menu, *Help*

Insert *Search* menu, *Replace mode* on/off.

Shift **Insert** Move what's right of the cursor to a new line below the current one (as **Return**, but the cursor does not move)

Control **Insert** *File* menu, *Insert. . .*

Clr/Home Move the cursor to the top of the document

Shift **Clr/Home** Move the cursor to the bottom of the document

Shift **Control** **Clr/Home** Move the cursor to the middle of the document

⇑ Move the cursor one line up

⇓ Move the cursor one line down

⇐ Move the cursor one column to the left. If it is at the beginning of a line, move to the end of the line immediately above

⇒ Move the cursor one column to the right. If it is at the end of a line, move to the beginning of the next line

Shift **⇑** Move the cursor up one page

Shift $\Downarrow$ Move the cursor down one page

Shift $\Leftarrow$ Move the cursor to the beginning of the line

Shift $\Rightarrow$ Move the cursor to the end of the line

Control $\Uparrow$ Move the cursor to the first line of the window

Control $\Downarrow$ Move the cursor to the last line of the window

Control $\Leftarrow$ Move the cursor one word or inter-word space to the left

Control $\Rightarrow$ Move the cursor one word or inter-word space to the right

Shift **Control** $\Uparrow$ Move the cursor up a quarter of a page

Shift **Control** $\Downarrow$ Move the cursor down a quarter of a page

num **0** $^{(*)}$ *Windows* menu, *Iconify*.

num **1** $^{(\dagger)}$ Move cursor to end of line. Also **Shift** $\Rightarrow$

num **2** $^{(\dagger)}$ Cursor down.

num **3** $^{(\dagger)}$ Page down. Also **Shift** $\Downarrow$

num **4** $^{(\dagger)}$ Cursor left.

num **6** $^{(\dagger)}$ Cursor right.

num **7** $^{(\dagger)}$ Move cursor to beginning of line. Also **Shift** $\Leftarrow$

num **8** $^{(\dagger)}$ Cursor up.

num **9** $^{(\dagger)}$ Page up. Also **Shift** $\Uparrow$

Control **num** **1** $^{(*)}$ Move window to lower left corner of screen

Control **num** **2** $^{(*)}$ Move window to bottom of screen

Control **num** **3** $^{(*)}$ Move window to lower right corner of screen

Control **num** **4** $^{(*)}$ Move window to left edge of screen

Control **num** **5** $^{(*)}$ Move window to middle of screen

Control **num** **6** $^{(*)}$ Move window to right edge of screen

Control **num** **7** $^{(*)}$ Move window to upper left corner of screen

Control **num** **8** $^{(*)}$ Move window to top of screen

Control **num** **9** $^{(*)}$ Move window to upper right corner of screen

Alternate **num** **1**–**9** Hold down **Alternate**, type a decimal ASCII code, and release **Alternate**. The character with that code is inserted (both in text and dialog windows).

num **(** $^{(*)}$ *Windows* menu, *V-Tile*

num **)** (*) *Windows* menu, *H-Tile*

num **/** (*) *Windows* menu, *Fold*

num ***** (*) *Windows* menu, *Full*

num **–** (*) *Windows* menu, *Smaller Font*

num **+** (*) *Windows* menu, *Larger Font*

num **Enter** *Options* menu, *Fix Dialog*. In a text window, **Enter** acts the same as **Return**

num **.** (*) *Windows* menu, *Top* or *Bottom*

Control **0** Dead key. See page 21.

Control **1** *Options* menu, *GEM Options...*

Control **2** *Options* menu, *Window Placement...*

Control **3** *Options* menu, *Text Appearance...*

Control **4** *Options* menu, *Editor Options...*

Control **5** *Options* menu, *File Options...*

Control **6** *Config* menu, *Rec Macro...*

Control **7** *Options* menu, *Interface...*

Control **8** *Config* menu, *Kurzels...*

Control **9** *Config* menu, *Filters...*

Control **A** *Edit* menu, *Select All*

Shift **Control** **A** *Options* menu, *ASCII Table...*

Control **B** *File* menu, *Import Binary...*

Shift **Control** **B** *Options* menu, *Clipboard...*

Control **C** *Edit* menu, *Copy*

Shift **Control** **C** *Text* menu, *Capitalize*

Control **D** *File* menu, *Save Selection...*

Shift **Control** **D** *Text* menu, *Sort*

Control **E** *Text* menu, *Erase*

Shift **Control** **E** *Text* menu, *Expand Tabs*

Control **F** *Search* menu, *Search...*

Shift **Control** **F** *Text* menu, *Format*. Also **F4**

Control G *Search* menu, *Next Target*

Shift Control G *Search* menu, *Previous Target*

Control H *Search* menu, *Replace*

Shift Control H *Options* menu, *Fonts...*

Control I *Edit* menu, *Invert Selection*

Shift Control I *Text* menu, *Reverse*

Control J *Text* menu, *Shift Left*

Shift Control J Join Lines

Control K *Text* menu, *Shift Right*

Shift Control K *Text* menu, *Kill Trailing Blanks*

Control L *Search* menu, *Goto Line...*

Shift Control L *Text* menu, *To Lower Case*

Control M *File* menu, *Save As...* Also Shift Control S

Shift Control M *File* menu, *Open Project...*

Control N *File* menu, *New*

Shift Control N *Options* menu, *Note Pad*.

Control O *File* menu, *Open...*

Shift Control O *File* menu, *Open Next*

Alt Control O as *File* menu, *Open...*, but texts will be opened in *alternative* text windows.

Control P *File* menu, *Print...*

Shift Control P *Edit* menu, *Select SouthEast Quadrant* (think of as *Select Paragraph*)

Control Q *File* menu, *Quit*

Shift Control Q *Options* menu, *At Exit...*

Control R *Windows* menu, *Resize...*

Shift Control R *Options* menu, *Text Rulers...*

Control S *File* menu, *Save*

Shift Control S *File* menu, *Save As....* Also Control M.

Control T *Options* menu, *Open Trash Can...*

Shift Control T *Text* menu, *Compress Tabs*

Control **U** *Windows* menu, *Close Top*

Shift **Control** **U** *Text* menu, *To Upper Case*

Control **V** *Edit* menu, *Paste*

Shift **Control** **V** *Edit* menu, *Overlay*

Control **W** *File* menu, *Close*

Shift **Control** **W** *Windows* menu, *Close All*

Control **X** *Edit* menu, *Cut*

Shift **Control** **X** *Edit* menu, *Delete*

Control **Y** *Text* menu, *Cut Line*.

Shift **Control** **Y** *Text* menu, *Delete Line*. Also **F8** and **Shift** **Delete**

Control **Z** *Search* menu, *Next Fragment*

Shift **Control** **Z** *Search* menu, *Previous Fragment*

C Glossary of Terms

Active window In normal **ATARI** applications this is the top window. **Edith** however can also send keyboard input to the window below the mouse pointer, even when it is in the background. In that case, that is the *active window*. The treatment of the active window can be set in the *GEM Options* dialog Control 1 from the *Options* menu. See page 41.

Alternative text window A normal window for editing text. Text window appearance and ruler settings default to those for *Alternative text windows*.

ASCII *American Standard Code for Information Interchange* refers to the 256 character **IBM** compatible character set. A file in ASCII format is a text which contains plain text without any additional information such as fonts, text rulers or pictures.

Auto indent When this mode is on, the editress will automatically indent lines created when you press the Return key. See page 11.6. Switched on/off by the menu item in the *Search* menu, or by F3 .

Auto wrap When this mode is on, a word will automatically be moved to the next line when it exceeds the right margin.

Backup A copy of a file made just before a new version is written to disk. See page 43.

Binary A file that contains arbitrary data (not text). See page 37

Block We don't use this term, as it is ambiguous—in some applications it means *Selection*, in others it means the contents of a *Text Buffer*.

Buffer A piece of text, divided into fragments, stored in computer memory, but not part of a text window. **Edith** has three text buffers, whose contents are displayed in the *Clipboards* panel Shift Control B (*Options* menu). See page 26.

Button font The font used inside buttons in dialog windows.

Carbon copy A copy of a file saved automatically at regular intervals or when reloading/abandoning a text. See page 43.

Clipboard The **GEM** clipboard contains text that can be used by different applications. **Edith** allows you to use the clipboard as the cut/paste text buffer. (*Editor Options* dialog Control 4 , *Options* menu). See page 43.

Default text window A normal window for editing text. Text window appearance and ruler settings default to those for *Default text windows*.

Dialog (window) A window through which you are communicating with a program. Not a text window.

Dialog fixer The small button left to the OK button in dialog windows. If it is selected, the dialog will be closed whenever OK is touched. Otherwise, it will stay open.

Editor tab A number of blanks (ASCII code 32) generated by pressing the Tab key.

Editor tab stop A triangle in the lower row of triangles in a text ruler. These tab stops are used when *tab key generates edit tabs* is selected in the *Editor Options* dialog.

Filter A text based `TOS` or `TTP` program called by **Edith** activated by pressing Alternate plus a function key. See page 48.

Fixer See *dialog fixer*

Font A type face including size and text attributes (boldface).

Form feed A command that makes the printer start with a new sheet of paper.

Fragment A contiguous piece of text within a selection or a part of a text buffer. See the picture on page 23.

Insert mode When the editress is in *insert mode*, what you type in text windows will be inserted into the text.

Keyboard & Mouse dialog This dialog has been replaced by the *Interface Properties* dialog in **Edith** versions V1.210P and up.

Kurzel An abbreviation for a string of characters, expanded by pressing the **Escape** key. See page 47.

Macro A sequence of keystrokes, recorded as on a tape recorder, and activated by **Control** plus a function key. See page 47.

Note pad A text window where you can type small notes to yourself while you are working with the editress package. There is one note pad, and its contents are automatically saved to a file in **Edith**'s system directory when it is closed.

Position stack Each text window has its *position stack*. This is a special type of stack, for storing cursor positions. Works exactly like described under *stack*, but after you have taken an item off the stack, it is immediately inserted again at the bottom; **Edith** also allows only 4 items to be stored in the position stack, so after taking a cursor position off the stack 4 times, you will get the same set of positions again. The cursor positions on the position stack are marked by small black triangles on the scroll bar of a text window.

Project file A file containing a run-time command line. Projects are opened by *Open Project...* **Shift** **Control** **M** in the *File* menu, or by the command line sequence `-p` *project-file*. The `-p` option within a project file acts as a *goto*.

Replace mode In this mode, what you type in a text window will erase other text. When the editress is in replace mode, the cursor take the shape of a rounded rectangle.

Selection The part of the text in a text window which is selected. A selection can consist of an arbitrary number of *fragments*. Note that unlike in other editors, while there is a selection, there still is a cursor, and text can be typed both inside and outside the selection.

Shortcut A key combination that activates a button or a menu item. Short cuts for menu items are displayed in the menus; in a button which has a shortcut, one of the letters is underlined. The button is then activated by pressing **Alternate** plus that letter.

South-East Quadrant See the illustration on page 24.

Stack A computer-scientific term. Think of a pile of letters. When you receive a letter, you put it on top of the pile. When you take one off to read it, you will get the last letter you added to the pile. Also referred to as the LIFO "Last In, First Out" principle.

Standard font The font used in dialog boxes and menus. Normally the standard **ATARI** font.

Substitute That which is substituted for a target by the *Search/Replace* combination. After searching, the cut/paste buffer contains the *substitute*(s) for the marked target(s).

Tab An *editor tab* or a *true tab*. Both refers to the space generated by the tab key and the tabulator stops on the ruler.

Target An occurrence in a text window of the string or pattern you are *search*ing for.

Text block See *block*.

Text buffer See *buffer*.

Text display font The font for editable text fields in dialog windows.

Text window class **Edith**'s text windows are divided into 6 classes; every class has its own colour, size, font and ruler settings. Text window classes are: *default*, *alternative*, *binary*, *text buffer/trash can*, *note pad* and *manual page*.

Trash can The trash can contains data discarded from text buffers. It can be used to retrieve data which was accidentally lost. See page 35.

True tab ASCII character 9. Normally displayed as white space stretching out to the next *true tab stop*.

True tab stop A triangle in the upper row of triangles in a text ruler. Usually placed at 8 column intervals.

Window Options dialog This dialog has been replaced by the *GEM Options* dialog in **Edith** versions V1.210P and up.

D Trouble-shooting

When I type text, it doesn't go into the right window. Select *Active Window: Top* in the *GEM Options* dialog.

Edith is slow when I am using MultiTOS. This is true. It may help if you select *Options: GEM Options: Active Window: Top* and *Options: Interface Properties: Always as Arrow*.

Edith is slow when I am using the GEM clipboard as the main text buffer. Use *Delete Line* to delete lines, and *Delete* instead of *Cut*, for this does not use the cut/paste text buffer. (*advanced*) Use a hard disk cache with a write delay.

I can't drag and drop when using MultiTOS. You have an old version of **MultiTOS**. Perhaps **TOSWIN**, **MINIWIN** or other applications will allow drag and drop communication with **Edith**.

I am using GEMINI, and if I drop an icon onto an Edith text window, the text is sometimes inserted, and sometimes opened in a new window. When an icon is dropped onto the *text*, it is inserted; when it is dropped on the menu bar or mover of the window, it is opened into a new window.

The dotted pattern in Edith's scroll bars is unequal. Select *Snap all windows to 4x4* in the *GEM Options* dialog or use Martin Osieka's program **winx**.

When I try to format a paragraph, I get garbage. You may have noticed that part of the text is flashing when you select *format*. This part does not start at the left margin of your text window. This is because formatting normally applies to what is called a *south-east quadrant*. In this way, **Edith** is capable of formatting a column of text while ignoring everything that is left of that column. To get format to work, make sure that the cursor is at the beginning of the paragraph (i.e. column 0) when you select *Format*.

E Technical Notes

E.1 Limitations

In principle, files can be of arbitrary size, can have an arbitrary number of lines and a line can have arbitrary many columns. Problems in loading, processing or displaying texts may arise if

- ▷ A file has more than, approximately, two million lines, worst case (window height of 800 pixels).
- ▷ The size of a file exceeds, approximately, 16 megabytes.

In **Edith** versions V1.210 and up, there is no more restriction to the number of characters on a line, where *any* text operation is concerned. In particular, the *Format* **F4** operation can be used to flatten long single-line paragraphs when importing files from a word processor package. However, the right margin and all tab stops must be at column numbers less than or equal to 256.

The *Search* operation will truncate any substitutes or substitute lines (which it writes into the *Cut/Paste* buffer), to a length of at most 256 characters.

The following processing takes place when loading a text or text buffer:

- ▷ All ASCII 0 characters are replaced by blanks (ASCII 32).
- ▷ A line feed is added at the end of the text. The last line feed in the text is not part of the editable text.

The following processing takes place when saving a text, but not when saving a text buffer:

- ▷ Line feeds are replaced with the line feed type selected in the *File Options* dialog.

E.2 Bugs and Problems

Below we try to give a fair list of problems we and the first users encountered in **Edith**. Hopefully, we will be able to solve these problems in later versions. We apologise for any inconvenience caused by the shortcomings listed below. If you are severely suffering from any of these problems, or encounter new ones, please write to **ZFC** giving precise details of your exact problems, as well as details of your computer system.

- ▷ The colour selection pop-up in the *Text Appearance* dialog is very slow under **MultiTOS**.
- ▷ Error handling in *Binary* editing windows is bad.
- ▷ When **Edith** is running in the **MultiTOS** environment, filter output will be sent to the console. Therefore it is always recommended that you open a console window. This is done by adding a line to your `GEM.CNF` file like the following:

  ```
  run c:\multitos\miniwin.app c:\multitos\bin\console.ttp
  ```

  Running an interactive filter like a terminal-based editor (**VI Emacs**) is awkward under **MultiTOS**. This is because **Edith**, **MINIWIN** and the **TTP** program are all competing for the keyboard input. This is subject to improvement in later versions.
- ▷ When **Edith** is running as an accessory under a single tasking **TOS** with **GEMINI**, filters will be launched in the Mupfel console, even if the *over text terminal* option is off for a filter. However, filters now work with all versions of **GEMINI**.
- ▷ In the root directory of drive `U:` under **MiNT**, directories created by a symbolic link are not recognized. This is due to a bug in **MiNT**. The extended attribute enquiry function returns attribute value zero for files symbolically linked in the `U:\` directory.
- ▷ When `EDITRESS.ACC` is loaded from Karsten Isakovitsch's program **Chameleon**, the accessory cannot be un-loaded (especially under **Gemini**).
- ▷ When **Edith** is running out of memory, macros and the repeat operation do not automatically abort. This may result in repeated alert messages. Try to press the break combination (both **Shift** keys).

E.3 Miscellaneous Remarks

- The way files are matched in the file selector depends on whether you are using **MultiTOS** or not. When **MultiTOS** is used, the **Edith** wildcard system will be used. On a single tasking system, the standard **TOS** wildcards apply, *unless* you use the range-style wildcards. When a normal single tasking **TOS** is used, this results in the following rather strange situation: when the pattern is `*`, the file `EDITH.TXT` will *not* be found, while when the pattern is `[a-e]*`, it will.

- When the *auto-save* options for configuration files are on (*At Exit* dialog), the configuration files are saved under the last name you used. For example, when the default preferences are automatically saved, normally auto save will use the name `default.prf`, but if you have just saved the preferences under the name `mydeflts.prf`, then auto save will use that name instead.

E.4 The Edith Accessory Protocol

Command lines can be sent to the **Edith** accessory by means of an **AES** message. Use the following **C** routine to send a command line (using the **Gemini** header file `VAPROTO.H`):

```
#include "aes.h"

#define VA_START  0x4711

int main
(  ...
)  {  int ap_id

      ...

   ;  ap_id = appl_init()

      ...
;  }

...

void send_message_to_editress
(  const char*    command_line
)  {  int         buf[8]
   ;  int         edith_id = appl_find("EDITRESS")

   ;  if (edith_id >= 0)
      {  buf[0] = VA_START
      ;  buf[1] = ap_id
      ;  buf[2] = 0
      ;  *((const char*)(buf + 3)) = command_line
      ;  appl_write(edith_id, 16, buf)
;  }  }
```

Beware of the danger that the pointer `command_line` should still point to valid information when `EDITRESS.ACC` starts processing it. As an alternative, if you specify the value 0 for `command_line`, the editress will try to read the project file `EDITRESS.CMD` in the `EDITHSYSTEM` directory.