Professional Software and Internet Media
Keizersgracht 684/3
1017 ET Amsterdam
The Netherlands
E-mail: `zfc@zfc.nl`
WWW: `http://www.zfc.nl/`

Over the past five years, the development of the **Edith** project has had great 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 (CWI) and many users of the internet.

This LaTeX manual, the **HTML** on-line pages and the **nroff** system manual page were derived from a single source using **GNU m4**; prepared for double sided photocopying using **psbook**. Pictures: **XV** and **Calamus DTP**. All text editing: **Edith**.

Contents

1 Edith Pro Support and Licensing

Thank you for buying an EDITH Pro/X11 licence.

We hope you will enjoy using the program; a powerful but easy-to-use editor should increase your productivity in a pleasant way. Although we have done as much as we can to make **Edith** a comfortable and universal program which can be adapted to a spectrum of personal tastes, there will always be parts of the design that do not match *your* personal desires. Never hesitate to let us know!

For ordering information, questions and suggestions, feel free to contact the author (preferably by e-mail) at


```
ZFC
Keizersgracht 684/3
1017 ET Amsterdam
The Netherlands

E-Mail:    zfc@zfc.nl
WWW:       http://www.zfc.nl/
Phone:     +31 20 4 208 248
```


COPYING

Edith is available freely for personal use from the **ZFC** world wide web pages. It is *not* allowed to distribute **Edith** in any way, without prior written permission from **ZFC**. With a licence, **Edith** will run legitimately on a selected Internet hosts, and you will be guaranteed full email support. Without a licence, **Edith** will offer full functionality, but will continuously issue complaining messages urging you to buy a licence. However, unlimited personal use of unlicenced copies is not prohibited. *Organizations* are expected to purchase a licence after a two-month evaluation period. Educational organizations are not excluded, but will be offered a 50% reduced rate for **Edith** licences.

FEEDBACK

We look forward to hearing your experience with the **Edith** package, and are especially interested in your use of the **external commands** p.82 facility and any **shortcut sets** p.70 you might have written. We hope users will allow us to create a ‘forum’ on the **ZFC** web pages where ideas for special support for computer languages can be exchanged.

2 Installation

If you haven't installed **Edith Pro** already when evaluating the package, please follow the instructions below; if you have, you can proceed to **INSTALLING A LICENCE**.

If you are installing from the MS-DOS formatted floppy disk that came with the manual, create a new directory, and extract the **tar** file on the floppy disk in the new directory. A new set of **tar** files will be created.

INVENTORY

Make sure that you have at least the following files: the 'common' tar file

```
edith1.3xx.common.tar.gz
```

contains the on-line manual pages, and predefined configuration files.

Then, you will need at least one of the following tar files, containing the executable binary for your system(s):

```
edith1.3xx.linux-elf.tar.gz
edith1.3xx.sgi.tar.gz
edith1.3xx.sun.tar.gz
```

INSTALLING BINARY AND LIBRARY

Extract the tar files, preferably in one of the following directories:

```
/usr/lib
/usr/local
/usr/local/lib
/usr/local/contrib
~/lib
```

If you choose a different directory, you will need to set the environment variable `EDITHSYSTEM` to the full path name of the extracted directory `edith`, henceforth called the *edith library directory*.

Now copy the binary to a place where your shell can find it, or add an entry to your `PATH` variable.

Copy the manual page `edith.1` into `/usr/local/man/man1`, and use **make-whatis (1)** to update `/usr/local/man/whatis`.

INSTALLING A LICENCE

The file `licence` belongs to the purchaser only; it is sent to the purchaser by e-mail and included on the floppy disk with the manual. A system wide licence is installed by copying it into the **Edith** library directory. Licences for only a few hosts may alternatively be installed only by the users of these host machines; in this case they should be copied to a directory `.edith` in the user's home directory. The alternative approach allows several individual licences to license the same executable and library files.

For the licence to work, either the command `hostname` should show your *full* internet host name, or alternatively, `hostname` should give the simple name of your host, and `domainname` should give the full domain name.

▷ On some systems (**Linux**), this is not automatically the case; then, you will need to define your domain at boot time, for example by putting a line like

```
domainname yourcompany.com
```

in a startup file run by **init**, e.g. `/etc/rc.d/rc.local`.

You should now be able to run edith by typing `edith` at your shell prompt.

3 Getting started

The first thing you will need to know as an **Edith** user is how to start the program from a command line, or how to install it as a file viewer in your file manager, mailer program, etc.

In principle, **Edith** is started much like any other text editor, with a number of exceptions, explained in the next section.

3.1 Command lines and the Edith Pro server

Most frequently, you will probably be starting **Edith** Pro/X11 simply by issuing a command like `edith` *file* from an **xterm** window. Everything you type following `edith` we will call the *command line*.

An **Edith** command line typically consists of a number of file names, optionally followed by a specification of a position within the file. Binary files can be opened by prefixing `-b` to the file name. To open a file or **URL** into a browser window, prefix `-l` to the name of the file or location.

A number of alternative ways of starting **Edith** is provided through a set of small programs (shell scripts): `edivisual`, `edimore` and `ediman`, whose purpose is to emulate more closely the interfaces of a number of conventional **UNIX** tools.

SYNOPSIS

edith [**-name** *rsrc-name*] [**-display** *displayname*] [**-iconic**] [**-nofork**] [{ *file* | **-b***binary-file* | **-l***location* } ...]

edivisual *file* [**+***line-no*]

edimore [*file*...]

ediman [*section*] *name*

OPTIONS

The following options can be put anywhere between the list of files.

`-display, -iconic, -name`
: Equivalent to the standard X options

`-c`*file*
: Open *file* and do not return until its window is open; if used more than once on the same command line, only the first occurrence is interpreted.

`-w`*file*
: As `-c`, but wait for the user to finish editing the file and close its window. This corresponds to the way a standard **Unix** editor like **vi**, that doesn't run in the background, works.

`-nofork`
: Don't use the **client/server protocol** p.9; the use of this option is discouraged.

EXAMPLES

`edith /home/wanda/.Xdefaults +35,-12`
: will open `.Xdefaults` in Wanda's home directory, **mark** p.39 the 35th line from the top of the file, the 12th line from the bottom of the file, and move to line 35.

`edith .cshrc +/setenv/`
: will open the file `.cshrc`, and jump to the first occurrence of the word **setenv** in that file.

`edith main.c:330`
: A colon has the same effect as a plus sign (but is immediately appended to the file name).

`edith 'main.c:?draw_border?'`
: Opens file `main.c` and jumps to the *last* occurrence of `draw_border` in that file.

`edith -ba.out`
: Opens file `a.out` into a binary editing window.

`edith -lhttp://www.zfc.nl/`
: Opens a **URL**, in this case the **ZFC** home page.

Sequences like `-c-l`*file* and `-w-b`*file* are allowed.

These command lines can also be typed in **Edith**'s file selector.

CLIENTS, SERVERS AND FORKS

The first time the **Edith** program is run on your X workstation or terminal, it will respond with a message like

```
Starting Edith Pro server on display :0.
```

Even though the command will finish immediately, **Edith** will continue to run on your display. On any subsequent times you call the **Edith** program, instead of starting another copy of the program, the newly run copy will consider itself a *client*, send a message to the copy that is already running (the *server*), and then exit.

This ensures that all **Edith** windows on an X display behave coherently; e.g. the panels for changing colours and fonts will apply to all active editor windows. It also helps to avoid editing the same file twice in different windows.

If you want to use **Edith** as a standard editor, you will want other applications to run it, like a mailer program. These programs usually call an editor and assume it will not terminate before the user finished editing the file. Therefore **Edith** provides the command line option `-w`*file*, which will cause the program to wait until the window for the file following `-w` is closed. Meanwhile, other files can be edited as usual.

Use the special shell script `edivisual` for installing **Edith** as a default editor; it is equivalent in synopsis to the **vi** editor, and in fact, if the `DISPLAY` variable is not set, e.g. because you are working from a **telnet** or modem connection, it will run **vi** instead, or whatever editor is pointed to by the environment variable `EDITOR`. `edivisual` makes use of the `-w` option. Another shell script called `edimore` is the counterpart of **more**, or whatever pages is pointed to by the `PAGER` variable; it makes use of the command line option `-c` which before exiting merely checks that the **Edith** server has loaded the file. `ediman` corresponds to **man**; `edigrep` to **fgrep**.

If you are sharing an X display between more than one user, you may have problems with file permissions, as the **Edith** server is started by user 1, and subsequent `edith` calls by user 2 will be sent to the server started by user 1, which does not have full permissions for user 2's files. There are two solutions to this problem. The first is to give each user's copy of **Edith** a different *resource name*. To do this, put the following lines in the different user's `.cshrc` (or similar) files:

/home/user1/.cshrc:

```
alias edith 'edith -name edithUser1'
```

/home/user2/.cshrc:

```
alias edith 'edith -name edithUser2'
```

A more rigorous, discouraged approach is to disable the client-server protocol by specifying the command line option `-nofork`.

3.2 The graphical user interface

Operation of the **Edith** system, as well as managing user preferences, is completely windows and mouse-based. Some details of the user interface can be modified under *Interface options...* (**Options menu** [p.61], Shift Ctrl I).

Edith's windows can be divided into a number of different types.

THE EDITOR MASTER PANEL

The *editor master panel* is the small window with a menu bar and a number of icons you see when you start the **Edith** server by typing `edith` at the shell prompt. During your **Edith** session, this window will always be present, be it most likely as an icon.

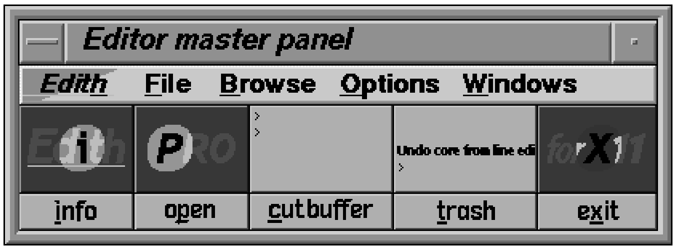

Figure 1: The editor master panel

TEXT WINDOWS

Text windows are discussed at length in the next section. The **menu bar** [p.46], **scroll bar** [p.20], **ruler** [p.66], **button bar** and lower **status/input line** in text windows can be en/disabled in the **size and gadgets panel** [p.74]. The text row and column are shown in the upper right corner of the window (menu bar must be switched on). When a text is modified, this is also indicated (status line must be on). When the status line is switched on, miscellaneous other information is shown there. Properties of text windows can be changed in the following panels: *Size and gadgets...* (**Options menu** [p.61], Shift Ctrl Z), *Text display/fonts...* (**Options menu** [p.61], Shift Ctrl D), *Colours...* (**Options menu** [p.61], Shift Ctrl C), *Text ruler...* (**Options menu** [p.60], Shift Ctrl R).

MENUS

Both the editor master panel and text windows have a menu bar. The menus can be used with the mouse, but also with the keyboard. Each menu title has an underlined

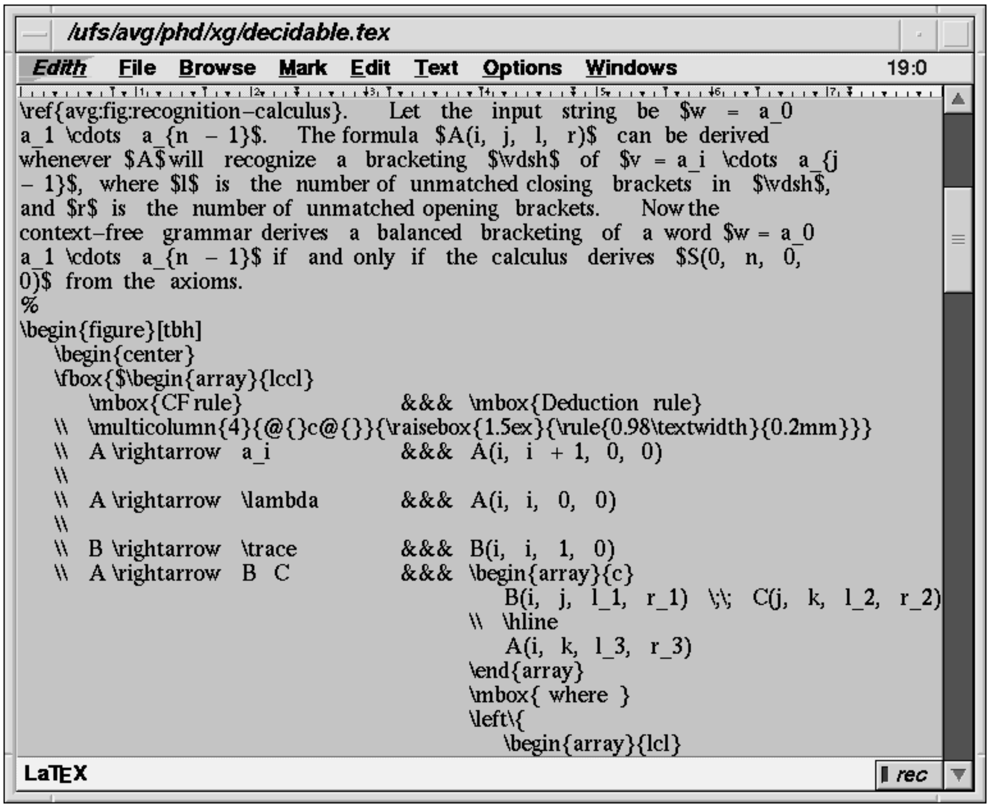

Figure 2: A text window

letter; typing that letter with the Alt key held down will open the menu. Use the cursor keys to walk up and down in the menu, or to move left and right to the other menus; use Return to select an item (or simply press the key code listed in the menu); press Escape to cancel. Most menu entries can be selected immediately with a keyboard combination which is shown in the menu.

BROWSERS AND BINARY WINDOWS

Two types of text windows are special. **Browser windows** [p.42] are read-only and have no text cursor. **Binary windows** [p.40] contain the binary representation of a file, and have a different behaviour for the operations *Save* (**File menu** [p.50], Alt S) and *Format paragraph* (**Text menu** [p.59], Alt Shift P).

PANELS

Panels come in two sorts; panels for configuring user options (opened from the **Options menu** [p.60] and **Windows menu** [p.62]), which can remain open while you are working in other windows; and panels that require you to give information immediately, before **Edith** can proceed; these are e.g. warning panels and the **file**

selector p.16.

In both types of panels, like in the menus, if a button has an underlined letter, typing that letter with Alt held down will do the same as clicking the button. The OK button, if marked with an 'arrow' sign, can be selected by typing Return. Use Ctrl w to close a window. If there is a button labelled cancel or similar, it is activated with the Escape key.

Frequently appearing in panels are *editable text fields*. In these white text fields, click to position the cursor. Use ⇐, ⇒ to walk left and right, use Begin and End to go to the beginning and end. Use Ctrl U to erase a field. Use Tab and Shift Tab to move forward/backward to other text fields. Move the cursor with Shift held down to select text in panel fields. Selection by moving over with the mouse while holding the left mouse button will also work.

Typing will delete the selection in panel text fields. The Delete key will also delete the selected text, but Backspace will not. Text is often *preselected* when you open a panel or a warning or question panel pops up, so when you start typing, what you type will replace the previous text.

Scrollable lists in panels can be operated with ⇑, ⇓, Page up, Page down, Ctrl Home, Ctrl End.

3.3 Using the file selector

Instead of typing `edith` in a shell in an **xterm** window, you can, when **Edith** is already running, also open text files from the *file selector*; choose *Open text...* (**File menu** [p.50], Ctrl O) from the master panel or from a text window. The file selector is also used to save files under a different name using *Save as...* (**File menu** [p.50], Alt Shift S), to import binary files using *Open binary...* (**File menu** [p.50], Shift Ctrl O), etc.

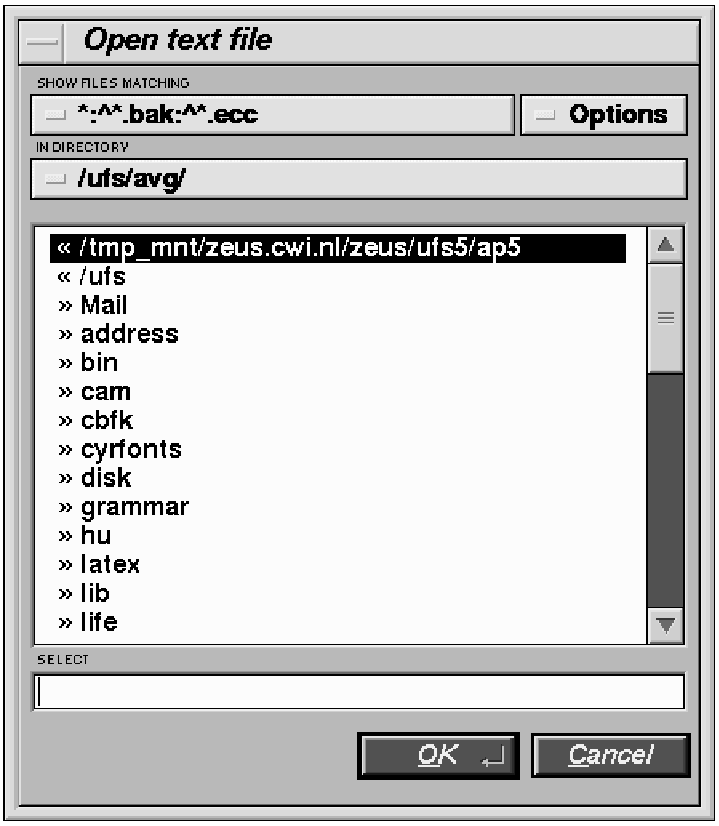

Figure 3: The file selector

SELECTING FILES USING THE MOUSE

Use the scroll bar to scroll through a directory; double click on a file to open it; click on a directory to go into it. Use the menu under *in directory* (Alt I) to find a number of recently accessed directories; use the menu under *show files matching* (Alt S) to change the displayed selection of files. There is a general-purpose menu under Options button.

SELECTING FILES USING THE KEYBOARD

Walk up and down through the list of files using the keys Up, Down, Page Up, Page Down, Home and Shift Home. Press Return to select a file or enter a directory. Press Alt R to re-read the directory. Clicking with the left mouse button while the Shift key is held down will cause the selected file to be copied to the text field at the bottom of the panel (so more than one file can be selected).

ENTERING FILES MANUALLY

Type one or more file names (i.e. an arbitrary **command line** p.8) and press return to open these. Non-existing names will be opened as new files, provided that their path exists. Opening a file which is already open will focus and top that window, as opposed to opening a second view on the same file. To open a second view on a file, use *New view on text...* (**Windows menu** p.62, Alt N). Wildcards (`*`, `?`, character ranges etc.) can also be typed. The Tab key will auto-complete a file name or wildcard pattern.

CONTROLLING THE DISPLAYED SELECTION OF FILES

The file mask shown at the top of the file selector, under *show files matching* has the same syntax as shell patterns, but can consist of a **sequence** of such patterns, separated by colons (`:`). If a pattern is preceded by a caret symbol (`^`), it is negative. If there are no positive patterns, `*` is assumed. The most recently used masks are available through the menu that pops up if you click on the mask. This menu also contains an option 'new pattern'; instead of using the mouse, the menu can be opened with Alt S.

EXAMPLES

`^*.bak:^*.ecc`
: will match any file, except Edith Pro backup files and carbon copies.

`m_*:n_*:^*.o`
: will match any file beginning with `m_` or `n_` but not those ending in `.o`

QUICK ACCESS TO FREQUENTLY VISITED DIRECTORIES

The last 6 directories visited are listed in the *in directory* menu at the top of the file selector (Alt I). This menu also allows you to store the currently displayed directory in a list of *bookmarks* (up to 8 bookmarks can be stored).

ACCEPTED FILE NAMES

The following are accepted file names. See also **extra URL types in browser windows** p.43.

`~` **or** `~username`
: Tildes and environment variables are recognized and expanded when the Tab key is used.

`a:/dir/file.txt`
: A file on a floppy disk. Interface uses **mtools**.

`ftp://host.domain/dir/file.txt`
: An anonymous **ftp** file. Interface uses **lynx**. Directories are considered regular files and are returned as an **HTML** file.

`http://host.domain/dir/file.html`
: A file on the web. Interface uses **lynx**.

`gopher://host.domain/dir/file`
: A gopher URL; output formatted in HTML. Interface uses **lynx**. Note that the **Edith** browser does not handle user input yet.

`user@host:~/dir/file.txt`
: A file on a remote system that has you listed in a `.rhosts` file. Interface uses **rcp** and **rsh**. Simply typing `user@host` will also work. See **rcp (1)** for details of the authentification system.

`mailto:user@host`
: File is mailed on close window. Interface makes use of **mail**.

`file.gz`
: File is uncompressed/compressed at load/save. Interfaced to **zcat**, **compress** and **gzip**. Does not work for remote files. (*But* because HTTP and FTP files are loaded using **lynx**, these types of file will also be unpacked.)

EXPERT FEATURES

Shell commands can be given by entering an exclamation mark (`!`) followed by the command.

If your environment contains a `$CDPATH` variable, whose syntax is identical to that of `$PATH` (a colon separated list of directory names), these will be taken into account by the file completion operation (Tab key).

4 The basics of text editing

This chapter completes the general introduction to using **Edith**. It explains the basic ways of moving around, editing, searching, etc. in text windows.

4.1 Moving around in editor windows

It is important to note that unlike in conventional window-based text editors, the cursor and the **selection** p.24 are independent. Also, the cursor is *always inside the viewable area* of the text, so when you slide through a document, the cursor may move afterward (it will only do so when necessary).

BROWSING WITH THE MOUSE

- ▷ Click the left mouse button anywhere in the text window to **move** the cursor (and **usually** p.24, to clear the selection).
- ▷ Use the arrow buttons to move up and down **line by line**.
- ▷ Use the slider to **slide** through the document.
- ▷ Click on the slider background with the left mouse button to **page up** and **page down**.
- ▷ Use the middle or right mouse button to **jump** anywhere immediately.
- ▷ On large documents, holding the Shift key down while sliding results in a more **high precision** movement (one text line per pixel in the scroll bar).

The size of the slider is an indication of how much of the document is displayed in the window; the text moves up and down immediately as the slider is moved.

There is no **horizontal** slider; to quickly scroll horizontally however, you can grab the **text ruler** with the left mouse button and move it around.

MOVING AROUND WITH THE KEYBOARD

The following key combinations move you up and down in a text window.

⇑, ⇓
: one row up, down

Page up, Page down
: page up, down, cursor row fixed in window

Ctrl ⇑, Ctrl ⇓
: paragraph up/down (the beginning of a paragraph is the first **letter** on a line following a line with no text).

Alt ⇑, Alt ⇓
: move displayed view up/down, cursor position fixed

Ctrl Home/Begin, Ctrl End
: move to the beginning/end of the text

Alt ⇐, Alt ⇒
: move to the previous/next **marker** p.55. On browser windows, these shortcuts are for *Back* (**Browse menu** p.52, Alt ⇐) and *Forward* (**Browse menu** p.52, Alt ⇒).

The operations in the next three lists, one for moving around, two for adding and deleting simple text, are called *line-based operations*. The notion of a line-based operation is important, because a sequence of such operations is thought of as one action; for example, after typing or deleting text, *Undo* (**Edit menu** p.56, Alt U) will make everything undone that has been done to the line you last changed. Moreover, these operations are the ones that are recorded by the **dribble recorder** p.36.

All keyboard shortcuts **not** *listed here, are found in the* **menus** p.46.

⇐, ⇒
: move one column (one letter) left/right.

Ctrl ⇐, Ctrl ⇒
: **word** p.68 left/right

Home, Begin, End
: move to the beginning/end of a line

BASIC EDITING WITHIN A LINE

a-z, A-Z, 0-9, !, @, /, **etc.**
: when typed are inserted directly into the text. When *Auto wrap* (**Options menu** p.60, Alt Shift W) is on, wrap words around whenever a **word** p.68 is started beyond the **right margin** p.66. The **beep** p.80 you will normally hear 3 columns before the right margin can be switched off.

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.

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.

Ctrl Backspace / Delete
: delete a **word** p.68 left/right of the cursor

Shift Backspace
: delete to beginning of line

Shift Delete
: delete to end of line

Erase line **(Edit menu** p.56**,** Ctrl U)
: empty the current line (but copy indentation from previous line)

LINES AND INDENTATION

Edith has a two sets of tab stops per window. The Tab key takes the tab stops in the lower row in the text ruler. By default, these tabs consist of three space characters. See **text ruler panel** p.66 for details. The tab and back-indent functions are line-based operations.

Tab
: to next editor **tab stop** p.66.

Shift Tab
: to the next true tab stop. *Note:* the behaviour of Tab and Shift Tab can be swapped in the **text ruler panel** p.66.

Escape
: back-indent (back to the previous level of indentation, looking upward in the text) or **shortcut** p.70 expansion.

By default, the *Auto indent* (**Options menu** p.60, Alt Shift I) function is on; this means that the Return key will copy the indentation from the previous line when creating a new one; so will *Erase line* (**Edit menu** p.56, Ctrl U).

Return, Enter
Create a new line (with indentation) and move the cursor to the new line.

Join lines (**Edit menu** p.56, Alt Shift J)
Join the current line to the next, replacing the indentation of the next line by at most one interconnecting space character.

Delete line (**Edit menu** p.56, Alt D)
Delete the current line.

Copy (**Edit menu** p.56, Alt C), *Cut* (**Edit menu** p.56, Alt X)
When there is no selection, *cut* and *copy* apply to the current line.

4.2 Selections and buffers

The selection and buffer system in **Edith** is slightly different than in most other editors, even those that are able to work with *rectangular* or *column shaped selections*. If parts of text are selected in an **Edith** text window, there is still also a cursor which can move around freely within the selected text. If the cursor moves out of the selection, it usually causes the text to be deselected.

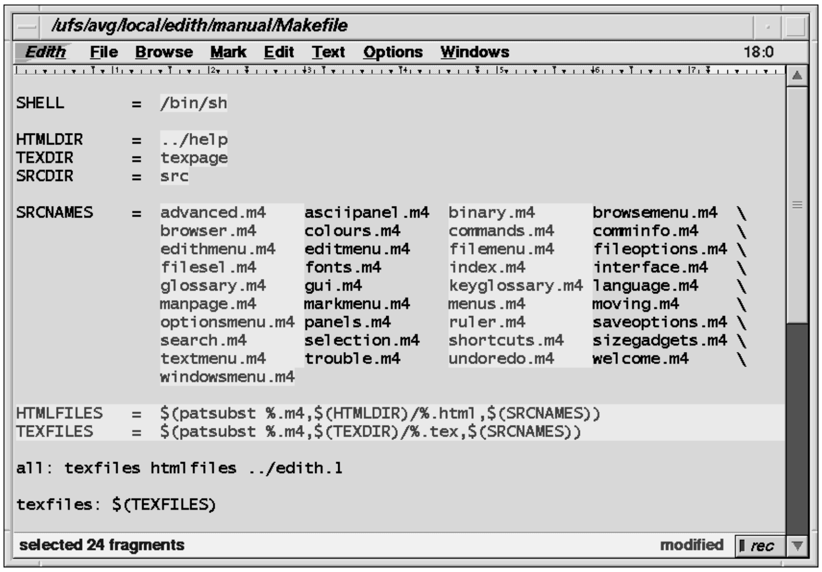

Figure 4: multi-fragment selections

The selection is *fragmented*, which means that the marked text is not necessarily a continuous part of the text. Any number of fragments can be marked simultaneously, and all text operations, like *Search engine...* (**Browse menu** [p.52], Ctrl F) and *Shift left* (**Text menu** [p.58], Ctrl J), have been designed to work on selections of any shape.

As usual in graphical text editors, there are *Cut* (**Edit menu** [p.56], Alt X), *Copy* (**Edit menu** [p.56], Alt C) and *Paste* (**Edit menu** [p.56], Alt V) functions. But as a consequence of the fragmentation of selections, the *cut buffer*, that is the place where a *copy*-ed selection is stored, is also divided into *fragments*. This hardly ever plays an important role; with one exception: the *Replace* (**Browse menu** [p.52], Ctrl H) operation, which exchanges one by one the fragments in the selection and the cut buffer.

The cut buffer can be inspected in the **editor master panel** [p.12]. An additional buffer visible in the master panel, the **trash buffer** [p.36], contains old text and old contents of buffers. Both can also be opened into a window from the **windows menu** [p.62]. A number of other buffers, not directly available to the user, plays a role in the *Undo* (**Edit menu** p.56, Alt U) system. Important to know is perhaps that *all* text windows share *one* buffer for the **undo** function, which explains why **undo** is usually only available for the window you worked in most recently.

SELECTION WITH THE MOUSE

- ▷ Move the mouse over the text while holding down the left mouse button to select a **continuous** text fragment. This will generally not move the cursor.
- ▷ Double click the left mouse button to select a **word** p.68.
- ▷ Move the mouse over the text while holding down the Ctrl key to select a **rectangular** area of text.
- ▷ Switch to *Additive selection* (**Mark menu** p.55, Caps Lock) to **add** new fragments to the selection.
- ▷ Unless in additive selection, click the left mouse button anywhere in the text window to **clear the selection** and move the cursor.
- ▷ Hold Shift and click/drag to adjust the end points of a selected fragment. If the selection was created with a double click, it will be aligned to word boundaries.
- ▷ Click and hold the left mouse button inside a selected fragment to **drag and drop** the selected text, either to another text window or the buffer displays in the **master panel**.
  - currently, no drag-and-drop protocols for communication with other applications have been implemented.
- ▷ Click the middle mouse button to **paste** the selection to or from another, or the same, window, including ones that do not belong to **Edith**.
- ▷ The right mouse button pops up a menu, whose contents vary depending on whether the menu bar is switched on or off.

SELECTION WITH THE KEYBOARD

Typing while there is a selection does not destroy or replace the selected text. Normally, walking away from the selection using the cursor keys will cause the selection to be cleared. This is *not* the case when *additive selection* is switched on. All these functions are found in the **mark menu** p.54.

Shift + cursor keys, Home, End, etc.
create a selection. When *Additive selection* (**Mark menu** p.55, Caps Lock) is selected, multiple fragments can be selected.

Alt Shift A
set beginning of (single fragment) selection. If there was no previous selection, the selected fragment will extend to the end of the text. Set the end point by (1) moving over with the cursor keys and *End selection* (**Mark menu** p.54, Alt Shift B), or (2) moving to the end point with the slider, and Shift click/drag.

Alt Shift B
set end of single fragment selection. If there was no previous selection, the selected fragment will start at the beginning of the text.

Alt Z
clear selection

Ctrl], Alt Q
select south-east quadrant.

Example: if the cursor in the picture is at the beginning of `texpage`, then the two fragments `texpage` and `src` form the quadrant south-east of the cursor.

Ctrl [, Alt Shift Q
select north-east quadrant.

Example: if the cursor in the picture is at the beginning of `src`, then the three fragments `../help`, `texpage` and `src` are a north-east quadrant.

4.3 Text search methods

Searching in **Edith** is divided into two facilities: the fast, incremental *quick* search-only method and the more elaborate *search engine*, which allows for replacement.

QUICK SEARCHING

To search for a single word once, literally and quickly, use *Quick search below...* (**Browse menu** [p.52], Ctrl Q) or *Quick search above...* (**Browse menu** [p.52], Ctrl P). By default, quick search is *incremental*, which is to say that as you enter the search target, **Edith** will start searching for the first occurrence immediately. Moreover, you can hit Tab to automatically 'complete' the search target string.

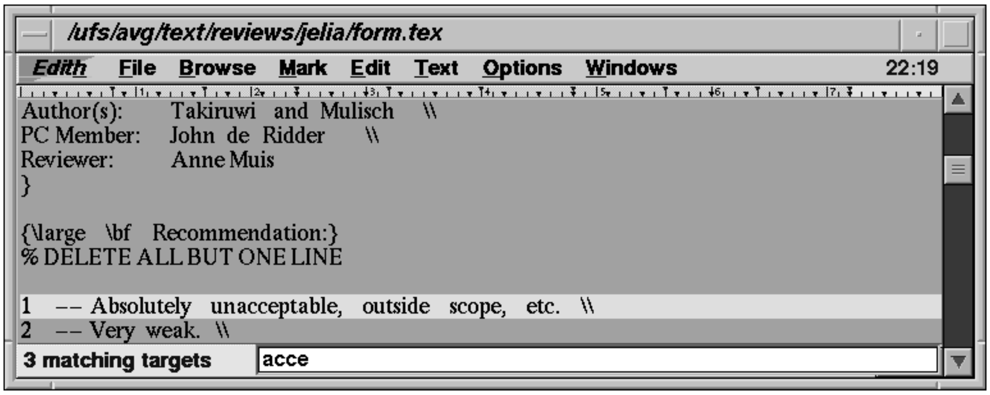

Figure 5: Quick search and target completion

Quick search marks the line containing the first target found in either direction, and leaves the cursor at the beginning of the matched occurrence in the text. To find further targets, use *Next target below* (**Browse menu** [p.52], Ctrl G) or *Next target above* (**Browse menu** [p.52], Ctrl D). To search for multiple targets, use the **search engine** [p.28].

- ▷ If the **status/input line** is switched off (in the **size and gadgets panel** [p.74], quick search will not be incremental; that is it will pop up a box where you can enter a target word, and searching will not begin until you click OK or Cancel.
- ▷ During incremental search, you can press Tab to 'complete' the entered target string. For example, if you typed `lawn` and there are 5 occurrences of the word *lawn* in the text, each followed by *mower*, your target will be completed to `lawnmower`, and **Edith** will tell you that there are 5 matching targets. Completion is done one **word** [p.68] at a time.

THE SEARCH ENGINE

To search for multiple targets, targets containing newlines, wildcard expressions, and for automatic substitution, use the *Search engine...* (**Browse menu** p.52, Ctrl F).

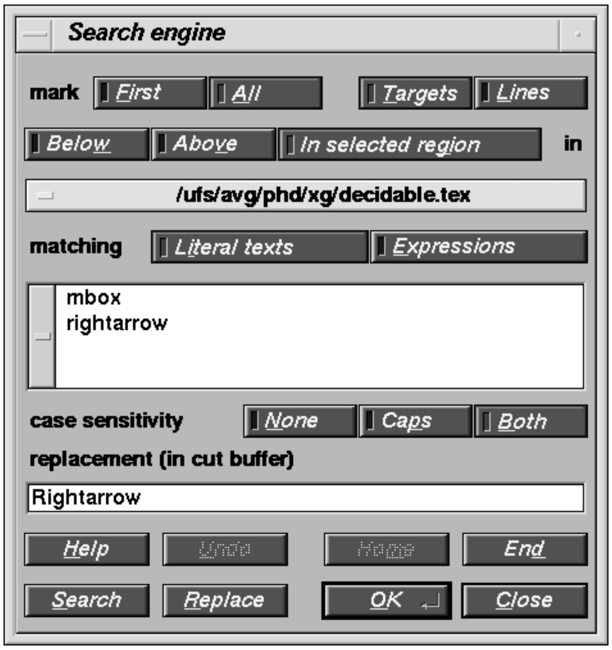

Figure 6: The search engine panel

SEARCH PARAMETERS

Select *first* if you want to highlight single targets, or *all* if all matching targets should be highlighted simultaneously. Select *mark lines* if instead of the targets, entire lines containing the targets should be selected. When lines are marked, the replacement function is disabled. Choose the direction in which you want to search by clicking *above* or *below*. If you preselected a number of lines in the text window, there will be a third option: *in selected region*, which performes a so-called *confined search*.

One or more targets can be entered into the upper text field. The grey button left of the text field will switch between single target search and simultaneous search for four targets.

Select *expression* if the targets contain wildcard patterns (see below for a list of wildcard patterns), or *literal text* otherwise.

Select *case sensitive: both* if upper and lower case should be distinguished, or *case sensitive: none* if they should be considered equal. You can also search 'caps-sensitive'; this means that an upper case character in the target expression will never match a lower case character, but a lower case character in the search expression is allowed to match an upper case letter in the text.

The *home* button moves the text cursor to the beginning of the file.

Each target has a substitute string or pattern associated to it. Switch between a target and its substitute using the Tab key.

TO SEARCH

After entering the parameters, click search or OK . Depending on the parameters, the first target, or all matching targets below the text cursor will be highlighted. If nothing is found, the search direction is reversed, and a subsequent Search or *Next target below* (**Browse menu** [p.52], Ctrl G) will perform an upward search.

TO REPLACE

There is no single 'search and replace' operation. Instead, the substitutes for the highlighted targets are stored in the cut buffer during the search operation. To replace the highlighted targets, press the *replace* button or select *replace* (Ctrl H).

The contents of the cut buffer can be inspected by opening the *Cut/replace buffer* (**Windows menu** [p.63], Shift Ctrl B).

NOTES

- ▷ the replacements are set only when a search action or sequence of search actions was started from the *Search engine*; not in *quick search* mode, and not when the *mark lines* option is selected.
- ▷ Sometimes it is more convenient to combine single target search with *Repeat dribble* (**Edit menu** [p.57], Ctrl R). Modify the first target by hand, and modify subsequent targets using *repeat*.

SPECIAL FACILITIES

Because there is a separate *search all* option, the search engine is good for more than simple search and replace actions. It can also be used to mark a number of targets and perform some other operation on the resulting selection, e.g. *Delete*

(**Edit menu** p.56, Alt Shift X), *Sort* (**Text menu** p.59, Alt Shift O), *Shift right* (**Text menu** p.58, Ctrl K) and *Repeat dribble* (**Edit menu** p.57, Ctrl R).

WILDCARD EXPRESSIONS

Use wildcard patterns (select the *expression* switch) to search for targets which are not literally the same, but match some general pattern with a number of 'holes'. You should also choose *expression* if you want to match newlines or true tabs. The following characters have a special meaning in the target and substitute expressions when wildcards are enabled:

Target expression

`?, !`
: Match any character, including line feeds and carriage returns

`*, &`
: Match any string or zero or more characters, not including line feeds or carriage returns

`[a-z]`
: Match any of a-z (always case sensitive)

`[^a-z]`
: Match any character not in a-z (always case sensitive)

`^`
: Match the beginning of a line

`$`
: Match the end of a line

`\n, \t`
: Match a newline or a tab.

`\c`
: Match the character c literally

`\( string \)`
: Marks part of the string for use in the replacement.

Substitute expression

`?`
: The character matched by `?`

`*`
: The string matched by `*`

`!`
: The character matched by `!` or by `[a-z]`, `[^a-z]`

`&`
: The string matched by `&` or the part enclosed by `\(` and `\)`.

`\c`
: The character `c` literally.

REPEATED SEARCH

To continue a single-target search, select *Next target below* (**Browse menu** p.52, Ctrl G) or *Next target above* (**Browse menu** p.52, Ctrl D). Multi-target and confined search can only be done from the *search engine* panel.

4.4 Special characters and the ASCII/ISO panel

Select *ASCII/ISO table* (**Windows menu** p.63, Shift Ctrl A) to open the *ASCII/ISO table panel*. It gives an overview of the character set associated to a window.

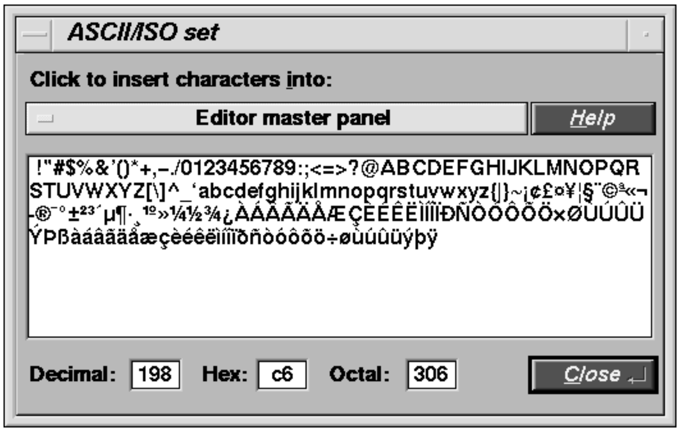

Figure 7: The ASCII/ISO table panel

▷ Click on a character to insert it into a text window or panel. In binary windows, the hexadecimal code will be inserted instead.

▷ Choose a different target window using the pull down menu.

ALTERNATIVE METHODS

Other methods to get characters not available on the keyboard are:

1. Hold down the Alt key, type a *decimal* ASCII code on the numerical keypad, and release alternate, or press the Enter key. The character corresponding to the ASCII code is inserted into the text.

   ▷ Note that the codes are *not* the same as those on an MS-DOS computer.

   ▷ To find the code for a character, open the ASCII/ISO table panel, and point at the character with the mouse. Its decimal, hexadecimal and octal representations can now be read at the bottom of the panel.

   ▷ For those who know what it means, any *keysym* can be typed.

2. Press the multi-key or compose key, then enter an accent followed by a letter. Accent keys are quotes, dashes, commas.

- ▷ The compose key must be defined for this to work. On some keyboards, this is the scroll lock key.
- ▷ For this and the next option, only the **ISO** Latin 1 character set is currently supported.

3. Press a *dead key* followed by a letter.

   - ▷ these dead keys, which correspond to accents, must be defined on your keyboard (see **xmodmap (1)**) for this facility to work.

4. There are preset **shortcut sets** p.70 that offer shortcuts for commonly typed special characters. For example, in the set **latin1.preset**, an **a** with an *accent grave* is obtained by entering an **a** followed by a backquote and Escape. There are shortcut sets for 6 **ISO** Latin character sets.

4.5 To undo and to do again

As usual, the operation *Undo* (**Edit menu** p.56, Alt U) is there to recover after making a mistake. It will undo the last operation performed, or in case of ordinary typing on one line, whatever has changed on that line since you started typing there.

Selecting *Undo* twice is equivalent to doing nothing. **Edith** does not know a *redo* command as found in many other text editing software. To recover from a more complex mistake, use the trash buffer.

Undo is often only available in the window you last worked in. This is because all text windows share a bit of memory (a **buffer** p.24) that stores the information needed for recovery.

USING THE TRASH BUFFER

For the sake of simplicity, **Edith** has the concept 'undo plus undo is equivalent to doing nothing'. This has the disadvantage that as soon as you make two subsequent mistakes, there is no easy way to recover the old text. However, in most of the cases, this data *will* be available under *Trash buffer* (**Windows menu** p.63, Shift Ctrl T). The trash buffer contains all material that the *Undo* (**Edit menu** p.56, Alt U) function has had to remove earlier. The old text can be selected, and dragged-and-dropped back to the original window easily.

In many cases, the trash buffer system turns out to be more versatile than a so-called *multi-stage undo* system. The most common example is: you type some good text (A), remove it by accident, then type a lot of good text (B). To get text A back in a multi-stage undo system, all of B would be lost.

THE DRIBBLE RECORDER

When you are typing, your actions are continuously being recorded. Recording starts when you type or delete a character, and then records subsequent typing, cursor motion and deletion on the same line. Recording stops automatically when (1) you move to a different line, (2) choose some non-line based operation or (3) walk left-then-right, or right-then-left. When recording stops automatically, cursor motion at the end of the recorded sequence is discarded. Recording can be switched on/off manually with the Rec button in the **status/input line** p.74;

this may be necessary if you wish a recording to begin or end in cursor motion operations.

The recording can be played back using *Repeat dribble* (**Edit menu** p.57, Ctrl R). Three recordings can be saved under keyboard combinations Alt 1, 2, 3 using *Store dribble* (**Edit menu** p.57, Ctrl T).

It is possible to play back a dribble over *several lines simultaneously*; just select the lines and *Repeat dribble* (**Edit menu** p.57, Ctrl R).

4.6 Advanced editing techniques

We now discuss a number of special features of **Edith** Pro that contribute essentially to the added editing value of the package, and shouldn't go unnoticed.

CREATIVE USE OF ARBITRARILY SHAPED SELECTIONS

The fragmented **selection** [p.24] system allows you to mark *any* selection of the text in a window, and all operations (except external commands) will operate sensibly on such selections. For example, you can enter a table of numbers, select a column, and use *Shift right* (**Text menu** [p.58], `Ctrl` `K`) to just that column to the right. You can cut away column 3 using *Cut* (**Edit menu** [p.56], `Alt` `X`), and use *Overlay* (**Edit menu** [p.56], `Alt` `Shift` `V`) to paste it back as column 5. You can align text with *Indent to column...* (**Text menu** [p.58], `Alt` `Shift` `D`). You can format non-continuous text as a paragraph using *Format paragraph* (**Text menu** [p.59], `Alt` `Shift` `P`). After switching on *Replace mode* (**Mark menu** [p.55], `Insert`), you can erase columns using *Delete* (**Edit menu** [p.56], `Alt` `Shift` `X`), and overwrite columns using *Overlay* (**Edit menu** [p.56], `Alt` `Shift` `V`) (instead of inserting them line by line).

REPEATING DRIBBLES ON MULTIPLE LINES

Repeat dribble (**Edit menu** [p.57], `Ctrl` `R`) (see **Undo & Repeat** [p.36]), when performed on a selection consisting of two or more lines, will repeat the recorded dribble to each line in the selection. It takes some time to understand the details of how the dribble recorder works, but especially when editing computer code or tables of numbers, multiple repeat is a valuable tool.

USING SHORTCUTS

For computer languages with rather long and tedious command sequences, such as LaTeX and HTML, *text shortcuts* can improve your editing convenience and speed considerably. A shortcut is an abbreviation (typically less than 6 letters). You type the abbreviation, then hit `Escape`, and it is replaced with a longer text. These shortcuts can freely position the cursor inside the replaced text and can ask for parameters—strings to be substituted in the replacement. Shortcuts can be defined in the **shortcut panel** [p.70].

TEXT MARKERS

Each editor window has a set of four *text markers* available. A position in the text is marked with *Put marker* (**Mark menu** p.55, Alt P). New markers appear on the scroll bar as small black triangles. The markers are traversed using *Back* (**Browse menu** p.52, Alt ⇐) and *Forward* (**Browse menu** p.52, Alt ⇒). Some functions (text home, text end, *Goto line...* (**Browse menu** p.53, Alt G)) create markers automatically. When all four markes are set and an additional one is requested, the one set longest ago is deleted first. The text marker system is also used when line numbers are specified for a file on the **command line** p.8.

SOURCE TAGS

Tags are known from the **vi** editor. A file `tags` in the directory of the file you are editing contains a number of keywords, each with a way for the editor to find information linked to that keyword.

In a sense, tags are to collections of plain text files what *links* are in sets of HTML documents on the web (or the **Edith** on-line manual). They are most frequently used by programmers who mark the definition of a function, so they can go back to its definition later just by selecting the function name where it's used in other files.

There are programs that create `tags` files for you automatically. **Edith** also offers *Create source tag* (**Mark menu** p.55, Alt Shift H) to allow you to enter arbitrary tags. Once a tag has been created for a keyword, you can jump back to it by selecting the keyword (or usually just putting the cursor inside it) and choosing *Edit source...* (**Browse menu** p.53, Ctrl E).

▷ In contrast to standard **tags** implementations, the `tags` file is allowed to be higher up in the path of the text window from which *Create source tag* (**Mark menu** p.55, Alt Shift H) or *Edit source...* (**Browse menu** p.53, Ctrl E) is chosen.

4.7 Viewing and editing binary files

Select *Open binary...* (**File menu** p.50, Shift Ctrl O) to open binary files for editing in hexadecimal **C** format.

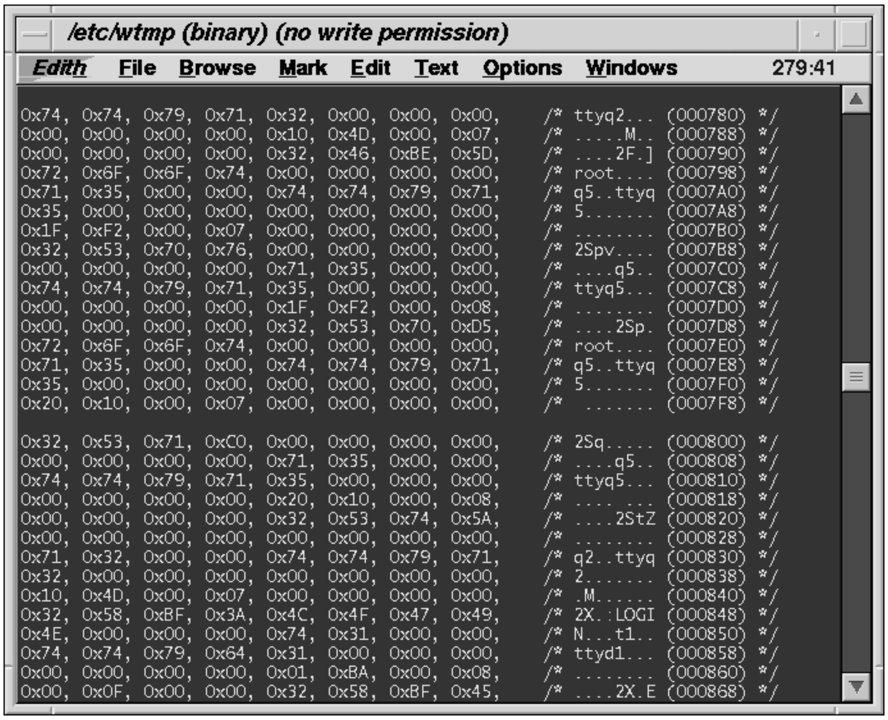

Figure 8: An editor window in binary mode

Binary mode is not essentially different in editing behaviour than a standard text window. The **C** representation of the file can be changed as you would edit a text file.

When *Save* is selected from a binary editing window, you are offered the choice to save the **C** representation or to compile it back into the (modified) binary file itself.

SYNTAX

The text in a binary editing window is interpreted as a sequence of hexadecimal bytes. **C** sugaring such as commas, curly brackets and `0x` prefixes is optional. Text between **C** comments (`/*`, `*/`) is ignored.

To check the syntactic correctness of your changes, select the changed lines and with the mouse and choose *Format paragraph* (**Text menu** p.59, Alt Shift P).

5 Using the Edith info browser

The *info browser* adds a flexible information system to the **Edith** system. Browsers pop up when functions such as *Edith info index* (**Edith menu** [p.48], Ctrl I), *Private info index* (**Edith menu** [p.48], Ctrl Y), *System manual...* (**Browse menu** [p.53], Ctrl M), *Open location...* (**File menu** [p.50], Ctrl L), *New view on text...* (**Windows menu** [p.62], Alt N), etc. are used.

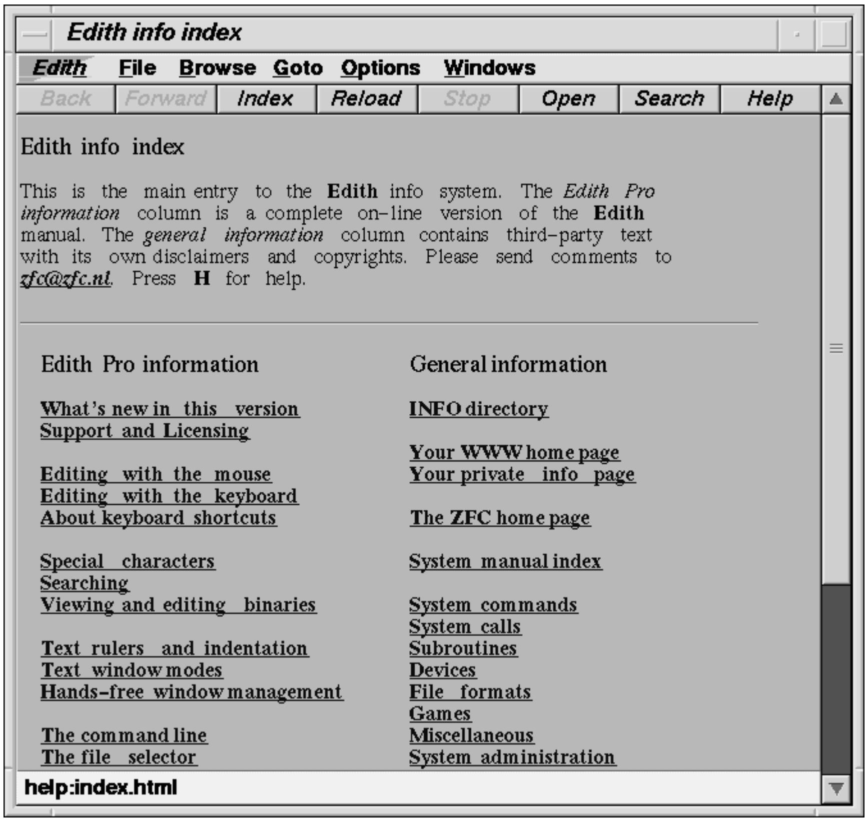

Figure 9: A browser window

BROWSING WITH THE MOUSE

The **Edith** browser windows work very much like a normal WWW browser.

- ▷ Click on a bold underlined phrase (a *link*) to visit a page with information described by that phrase
- ▷ Click with the right mouse button to open a menu with functions such as *back* and *forward*
- ▷ The *Mark* menu contains previously visited pages and bookmarks.

BROWSING WITH THE KEYBOARD

Most standard key combinations will work.

H, ?
: Open this help page.

B **or** *Back* (**Browse menu** p.52, Alt ⇐)
: Back to the previous page.

F *Forward* (**Browse menu** p.52, Alt ⇒)
: Return to the page you were watching before you selected *Back*.

D
: Back to the info index, or when in an **info** page, to the **info** top directory.

Q **or** *Close window...* (**File menu** p.51, Ctrl W)
: Close browser window.

Space
: Page down.

Tab, Shift Tab
: Go to the previous/next link in **info**, **help** and **HTML** pages.

Enter
: Open the selected link.

MAKE A PRIVATE INFO HOME PAGE!

If you know a little about HTML, you can create a personal **“hotlist”** in HTML with references to the information you need most frequently while editing text. You can put this page full of clickable items that open sets of files, open a file selector, go to a frequently visited web page, open a system manual, etc.

To add such items to your personal info home page, just go to your private page with *Private info index* (**Edith menu** p.48, Ctrl Y), and then select *Edit source...* (**Browse menu** p.53, Ctrl E).

A preset frame for writing your personal home page has been installed for you in the file `.edith/private_index.html` in your home directory. You can change this default location by setting the environment variable `$PRIVATE_HOME`.

USING HTML TO CREATE NEW PAGES

The **Edith** browser recognizes a subset of HTML 2.0 similar to what is supported by text-only web browsers such as **lynx**. The following extra types of URL are supported (see also the **file types** p.18 that are recognized throughout **Edith**).

`help:file.html`
: An **Edith** help file.

`edit:/dir/file.txt /otherdir/otherfile.txt...`
: Open an editor on one or more text files. The file name can be any file recognized by the **file selector** p.18.

`info:file` **or** `info:file#node`
: An **info** page.

`apropos:keyword`
: Open an index on the **system manual**. Leave out keyword for the complete index.

`man:name`
: Open a system manual by name.

`selectfile:/path/dir/`
: Open the file selector on a local or remote directory.

`www.foo.com` **or** `ftp.foo.com`
: You are allowed to leave out `http://` or `ftp://`.

6 Menus and shortcuts

The **Edith** menus reflect the logical order of using a text editor. Almost all entries in the menus have a *keyboard shortcut* that is a combination of Alt, Ctrl, Shift and a letter or special key. You will probably want to learn the most frequent key combinations; combinations for less frequently needed functions you will probably never learn by heart and always select from the menus.

Edith menu p.48 (Alt H)
Read the manuals

File menu p.50 (Alt F)
Open a file

Browse menu p.52 (Alt B)
Move to where you were editing

Mark menu p.54 (Alt M)
Select or point to bits of text. (**Goto** on browser windows).

Edit menu p.56 (Alt E)
Copy or delete text that is selected/pointed to

Text menu p.58 (Alt T)
. . . or modify them

Options menu p.60 (Alt O)
Adapt the behaviour of the editor to your personal taste

Windows menu p.62 (Alt W)
Arrange windows, etc.

6.1 The Edith menu

The *Edith* menu contains help functions and access to external shell commands.

ENTRIES IN THE EDITH MENU

About Edith Pro...
Pops up a box with the address and version information for your current copy of **Edith**.

*

Help..., Alt H
Opens an info page with immediate help varying on the type of window you are working in.

Edith info index..., Ctrl I
Opens the on line index on this documentation, including references to (1) emacs info, (2) all system man pages listed in 'whatis' files, (3) your WWW home page and (4) your 'private home page'.

Private info index..., Ctrl Y
Open your **private info index** p.43, where you can put links to whatever information you like to have quickly available, or want to edit regularly.

*

Execute command..., Alt Shift M
Execute an arbitrary command (such as `make`), whose output (if any) will be redirected to a browser window.

Filter..., Alt Shift F, Ctrl !
Pops up a panel in which you can enter a shell command line. The command will be executed with the selected text as its standard input; the standard output of the command will be substituted for the selection.

A running filter can be killed using the Escape key.

If the selection consists of multiple fragments, these will be glued together (as in *Toggle single/column* (**Mark menu** p.54, Alt Y)).

*

Configure commands..., Shift Ctrl X
Opens the **external commands** p.82 panel, where you can specify external commands with an entry in the *Edith* menu.

*

The next two entries in the **Edith** *menu contain a customizable list of external commands defined in* **external commands panel** p.83:

Default setup..., Shift Ctrl Y
Generic external commands available at all times.

User modes..., Shift Ctrl U
Mode-specific external commands; available only on text windows of a mode other than *Default setup*.

6.2 The File menu

The *File* menu contains facilities for reading and writing texts or binaries from/to files or networks, and for opening and closing windows.

ENTRIES IN THE FILE MENU

Open text..., Ctrl O
: Open the **file selector** p.16 to select one or more text files for editing.

Open next, Ctrl N
: Open the next file from the **input queue** p.78.

New untitled text, Shift Ctrl N
: Open a new, untitled text window in 'default' editing mode.

*

Open binary, Shift Ctrl O
: Open the file selector to select a file opened in **binary editing mode** p.40.

Open location, Ctrl L
: Open a local or remote file into a browser window. Instead of the file selector, a box pops up in which you type a *full* file name or WWW-style resource locator (URL). All file types accepted by the **file selector** p.18 and **browser** p.43 are accepted. Remote files will be loaded using auxiliary programs such as **lynx** and **rcp**.

*

Reload..., Alt R
: Reload a text or a browser page. When reloading a text, a panel will pop up giving you the choice between the original file, its last written backup or the last written carbon copy.

Insert..., Ctrl Insert
: Select a file to be inserted (as in *Paste* (**Edit menu** p.56, Alt V)) into the text window.

*

Save, Alt S
Save the contents of an editor window back to the original file. This menu entry will be enabled regardless of whether the text has been modified, but the entry adds ‘needed’ or ‘not needed’ depending on whether modifications where made to the file since it was last read from or written to a file.

Save as..., Alt Shift S
Select a new file to save contents of current editor window to. (Changing the file name associated to the window, i.e. in the title bar, is optional).

Save all texts, Alt Shift Y
Save all open (editable) text windows that have a valid file name, *i.e.*, are not ‘untitled’.

Save selection, Alt Shift Z
Save the selected text.

*

Close window, Ctrl W
Close a window. Not to be confused with, and often to be preferred to, the *Quit* option below.

Quit, Shift Ctrl Q
Leave **Edith Pro**. All **Edith** windows on the current display will be closed, and unless you answer any resulting pop-up panel question with Cancel, the **Edith Pro server** p.9 will exit. Note that once you tell **Edith** to stop asking for a confirmation at exit, the confirmation box will not appear again, unless you switch it on again in the **interface options** p.80 panel.

It is tempting, but not rewarding, to use *Quit* too often. Usually, it is sufficient to just *Close* a text window.

6.3 The Browse menu

The *Browse* menu is for access to operations that move around in and between text files. Groups found in the menu include **info and web navigation**, **Searching** p.28, and an interface to the **man**, **apropos**, **ctags** commands.

ENTRIES IN THE BROWSE MENU

Back, Alt ⇐
: Go back to the previous page in the info browser, or go back to the previous marker in an editor window.

Forward, Alt ⇒
: Go back to the next page (*i.e.* whereever you were before you went *back*) in the info browser, or go back to the next marker in an editor window.

Stop, Escape
: Stop a running filter or external command (by sending SIGTERM).

*

Search engine..., Ctrl F
: Open the **search engine** p.28 panel that allows for wildcard search, multiple target search, confined search, and the selection of text for automatic substitution.

Next target below, Ctrl G
: Find the next target. Applies both to quick search and engine search, whichever was used last.

Next target above, Ctrl D
: Find the next target, but in upward direction.

Replace, Ctrl H
: Swap the contents of the selection and the *cut buffer*. Used most often after a **search** p.28 operation, but also useful for manual replacement. If the number of fragments in the text is not equal to that in the cut buffer, the entire buffer will be substituted for each of the selected fragments.

Quick search below..., Ctrl Q, Ctrl /
: Search quickly for a literal string, without wildcards or substitution. In most cases, quick search will be **incremental** p.28. For fast manual substitution in combination with quick searching, the *Repeat dribble* (**Edit menu** p.57, Ctrl R) operation can be useful.

Quick search above..., Ctrl P, Ctrl ?
    Search p.28 for a literal string backward in the text.

*

System manual..., Ctrl M
    Find a system manual page. When a word is selected, the manual for that word will be opened. When there is no selection, a box will pop up asking you to specify the title of a manual page.

Apropos..., Ctrl A
    List the system manual pages whose description contains the selected keyword. Otherwise similar to *system manual*.

Edit source..., Ctrl E
    On HTML windows (including **Edith** info windows), this will open the HTML source in an editor window. On editor windows, the source definition of the keyword the cursor is on will be searched using the **tags** p.39 method. If there is no such keyword, an input box will pop up. Source tags can be created automatically using programs such as `ctags`, or by using *Create source tag* (**Mark menu** p.55, Alt Shift H).

*

Goto line, Alt G
    Enter a line number and jump to that line. A *sequence* of line numbers is also allowed. The previous cursor position, and the any line numbers following the first one entered are marked as in *Put marker* (**Mark menu** p.55, Alt P).

Find bracket, Ctrl B, Ctrl %
    When the cursor is before or after a bracket (preference to before), moves the cursor to the matching bracket in the text. Brackets matched currently are `(`, `)`, `[`, `]`, `{` and `}`. Quoting is not taken into consideration.

6.4 The Mark menu

The *Mark* menu provides facilities for marking text and text positions. They all apply to text windows only.

ENTRIES IN THE MARK MENU

Select all, Alt A
: Select the entire text.

Invert selection, Alt I
: Turns the selection inside out (whatever was selected becomes unselected and the other way around)

Clear selection, Alt Z
: Remove the selection.

*

Begin selection, Alt Shift A
: Set or change the starting point of a single-fragment selection. If there is no previous selection, the end point is the end of the text.

End selection, Alt Shift B
: Set or change the end point of a single-fragment selection. If there was no selection, the starting point is text home.

*

Select SE quadrant, Alt Q, Ctrl], Ctrl }
: Select a *south-east quadrant*, i.e. everything right of and below the cursor until there is a line which doesn't stretch out beyond the current cursor column (except for white space or indentation characters). This is *a bit* in the spirit of the South-East quadrant rule found in some programming languages, but not quite. It is especially useful for aligning arrays of variable definitions in programming languages, or indented/juxtaposed paragraphs of text. The resulting selection will be discontinuous, i.e. it consists of a fragment per selected line.

Select NE quadrant, Alt Shift Q, Ctrl [, Ctrl {
: Select a *north-east quadrant*: as in select SE quadrant, but upward.

Toggle single/column, Alt Y
: Transforms a continuous selection to a column, or a multi-fragment selection back to a continuous one.

*

Additive selection, Caps Lock
: Toggle *additive selection* on/off. When this function is on, selecting fragments with the mouse or with the keyboard will *not* destroy any fragments of text selected before, creating a so-called *multi-fragment selection*. Also, when additive selection is switched on, the text selection is *persistent*, i.e. text will not be unhighlighted when you walk out of the selected area using the arrow keys.

Replace mode, Insert
: Switch between insert and replace mode. In replace mode, text that is typed will overwrite other text; moreover the *Delete* (**Edit menu** p.56, Alt Shift X) and *Overlay* (**Edit menu** p.56, Alt Shift V) functions will behave differently.

*

Put marker, Alt P
: Mark the current cursor position. Up to four positions can be marked, and will appear as black triangles on the scroll bar of a text window. At most one position per line is marked. The marked positions can be traversed using *Back* (**Browse menu** p.52, Alt ⇐) and *Forward* (Alt ⇒).

Create source tag, Alt Shift H
: Enters the selected keyword into the closest `tags` file, so that it can be found by *Edit source...* (**Browse menu** p.53, Ctrl E). If the cursor is not on a keyword, or something larger than a keyword is selected, you will be asked to enter the keyword.

6.5 The Edit menu

The *Edit* menu contains those operations that involve *manipulation* or *transfer* of text in the sense of copying and deletion, *undo* and repeat.

ENTRIES IN THE EDIT MENU

Undo, Alt U
: Undo the last operation performed. This is a **single-stage** p.36 undo function.

*

Cut, Alt X
: Copy the selected text into the *cut buffer* and delete it.

Copy, Alt C
: Copy the selected text into the cut buffer. If multiple fragments are selected, the resulting cut buffer will be accordingly 'fragmented'.

Paste, Alt V
: Paste the contents of the cut buffer into the text as a single continuous fragment; the inserted text is selected.

*

Delete, Alt Shift X
: Delete the selected text. In **replace mode** p.55, the text is not deleted, but replaced by white space.

Overlay, Alt Shift V
: Paste the cut buffer into the text, left-aligning the inserted text fragment-by-fragment, line-by-line, at the column position of the cursor. This can be used, e.g., to put two narrowly formatted paragraphs, or columns of numerical data next to eachother. In **replace mode** p.55, the text is not inserted, but overwritten line by line (e.g. to replace a column of numbers).

*

Erase line, Ctrl U
: Erase the current line, or erase an input field in a panel. Generates indentation like the Return key does; see *Auto indent* (**Options menu** p.60, Alt Shift I).

Delete line, Alt D
    Delete the current line.

Join lines, Alt Shift J
    Join the current line to the next one, putting one blank character in between, after removing the indentation of the next line, including non-blank indentation characers as defined in the **language options** p.68 panel.

*

Repeat dribble, Ctrl R, Ctrl .
    Repeat the last sequence of cursor/typing operations applied to a single line.

- ▷ If there is a multi-fragment selection or a single selection of at least 2 full lines, the dribble is applied to each line of each fragment in the selection.

Store dribble, Ctrl T
    Store the current dribble (see above). A panel pops up where you can choose to store the last sequence of cursor/typing operations as dribble 1, 2 or 3.

Do dribble 1, 2, 3, Alt 1, 2, 3
    Repeat stored dribble 1, 2, 3, as in *repeat dribble*.

6.6 The Text menu

The *Text* menu contains those operations on a file that *alter* selected fragments of text. All operations in the text menu operate on the selection, on a fragment-by-fragment, then line-by-line basis. If there is no selection, a default scope will be selected.

ENTRIES IN THE TEXT MENU

The following operations have a **default scope** *of one word.*

To lower case, Alt Shift L
: Convert all letters, including ones in the upper ISO-Latin-1 set, to lower case.

▷ Functions involving the Upper/lower case distinction may not work correctly for all 8 bit character sets (reversed for Cyrillic koi set, not at all for ISO-Latin-5).

To upper case, Alt Shift U
: Convert to upper case.

Capitalize, Alt Shift C
: Capitalize all letters that are not preceded by a letter.

*

The following operations have a **default scope** *of one line.*

Shift left, Ctrl J
: Shift the selected lines left over one **tab stop** p.66.

Shift right, Ctrl K
: Shift right over one tab stop.

Shift left one, Alt J
: Shift the selected lines left by one space character.

Shift right one, Alt K
: Shift right by one space character.

Indent to..., Alt Shift D
: Pops up a panel in which you enter a column number. All selected lines will be indented to exactly that column (a ragged left margin will be destroyed).

*

Format Paragraph, Alt Shift P

Format a paragraph, which is from the cursor until the first empty line, where the 'indent characters' specified in the **language options panel** p.68 are considered 'blank'. If so selected in the **editor options** panel, the paragraph will be aligned to the right margin as set on the **text ruler** p.66.

The default scope of *format paragraph* is one paragraph, starting at the cursor position. Multiple paragraphs may also be formatted at once (just select the paragraphs to be formatted), but this feature is a bit problematic with different levels of indentation.

The following operations have **default scope***: south-east quadrant.*

Sort lines, Alt Shift O

Sorts the selection fragment-by-fragment, line-by-line.

Reverse lines, Alt Shift R

Reverses the lines in the selected fragments.

*

The following operations have **default scope***: whole text.*

Compress tabs, Alt Shift T

Convert as many sequences of space characters as possible to true tabs.

Expand tabs, Alt Shift E

Expand true tabs to sequences of spaces.

Kill trailing blanks, Alt Shift K

Remove any space or tab characters at the end of selected lines.

Convert newlines..., Alt Shift N

Pops up a box that offers a choice between MS-DOS, Macintosh and UNIX style newlines. The selected text is then converted to that type.

6.7 The Options menu

The *Options* menu provides access to dialog panels for a large number of configurable options. Everything that is configurable in the **Edith** system can be set from within the program, and manual editing of configuration files is not recommended.

ENTRIES IN THE OPTIONS MENU

Auto indent, Alt Shift I
: Switch on/off the auto-indent function. When the **auto indent** function is switched on, leading **white space** p.68 is copied to the new line when Return is pressed, and on *Erase line* (**Edit menu** p.56, Ctrl U).

Auto wrap, Alt Shift W
: Switch on/off the auto-wrap function. When **auto wrap** is switched on, words typed will be wrapped whenever they go beyond the *right margin* (set in the **text rulers panel** p.66 or by clicking on the text ruler with the right mouse button).

Show matching brackets, Alt Shift M
: Switch on/off the matching brackets function, that highlights an opening bracket if you type a closing one: whenever a closing bracket `)`, `]` or `}` is typed, the editor will search for the corresponding opening bracket. If it is found, the cursor will flash briefly (or longer, when it is of a different type) at the opening bracket. If the bracket structure is wrong, you will hear a **beep** p.80 or see a message in the **status line** p.74.

Format right-aligns, Shift Ctrl P
: When **format right-aligns** is selected, *Format paragraph* (**Text menu** p.59, Alt Shift P) will fill paragraphs such that their right margin is aligned. Note that this does not affect the behaviour of the **auto wrap** function.

*

Text ruler..., Shift Ctrl R
: Open the **text rulers** p.66 panel, that specifies the positions of tab stops, the right margin, and the way the Tab key works.

Language options..., Shift Ctrl L
: Open the **language options** p.68 panel, which defines the appearance of a 'word', which characters are to be regarded as 'blank', and how *word-left* and *word-right* should behave.

Shortcuts..., Shift Ctrl S
Open the **shortcuts** p.70 panel, that defines language-dependent abbreviations.

*

Text display..., Shift Ctrl D
Open the **text display/fonts** p.72 panel that defines fonts and some special options for text display in editor windows.

Size and gadgets..., Shift Ctrl G
Open the **text window size and gadgets** p.74 panel, that determines the default size of text windows, and switches on/off the *gadgets* of a text window, such as the scroll bar and the ruler.

Colours..., Shift Ctrl C
Opens the panel that allows you to distinguish different sorts of text windows by using **colours** p.76.

*

File options..., Shift Ctrl F
Open the **file options** p.78 panel (maximum number of files to open, carbon copies, backups).

Interface options..., Shift Ctrl I
Opens the **interface options** p.80 panel (window placement, window management, cursor blink speed, caps lock).

*

Save options...
Save the full, current **Edith configuration** p.88. There is no keyboard shortcut for safety reasons, but the command can be selected from the keyboard by opening the *Options* menu with Alt O and walking down with the cursor keys.

6.8 The Windows menu

The *Windows* menu contains operations for managing the size and position of windows, switching and defining editor modes, and gives access to an ASCII character set panel and two temporary editor buffers.

ENTRIES IN THE WINDOWS MENU

New view on text..., Alt N
: Opens a new, read only, view on the text in the current edit window.

*

Smaller font, **Num pad** -
: Shrink the font of an editor window.

Larger font, **Num pad** +
: Enlarge the font of an editor window.

Resize..., Ctrl Z
: After selecting this entry, you can type a new size for a text window. The size should be entered as **cols**×**rows** where **cols** and **rows** are the new dimensions of the window. Both **cols** and **rows** may be omitted. Press return to revert to the original size as set in the **size and gadgets panel** p.74.

*

Text window modes..., Shift Ctrl M
: Open the **text window modes** p.64 panel that manages the individually defined parameter sets and triggering file masks for different editing modes.

Switch mode..., Ctrl S
: Pops up a menu in which you can choose a new editing mode for the current text window.

*

Cycle windows, Ctrl Tab
: Cycle through the open (mapped) **Edith** windows.

Tile vertically, Shift Ctrl V
Arrange the open (mapped) edit windows vertically stacked, covering the full screen.

Tile horizontally, Shift Ctrl H
Arrange the open (mapped) edit windows horizontally stacked, covering the full screen.

*

Cut/replace buffer, Shift Ctrl B
Open the *cut buffer* into a read-only browser window. The cut buffer and trash buffer can also be inspected by double clicking on the fragments displayed in the *Editor master panel* that is always open (usually as an icon) when the Edith server is running on your display.

Trash buffer, Shift Ctrl T
Open the **trash buffer** p.36 which stores previously deleted material, into a browser window.

ASCII/ISO table, Shift Ctrl A
Open the **character set table** p.34 panel.

*

Close all windows, Shift Ctrl W
Close all **Edith** windows (except the master panel, which is at most iconified).

7 Customizing the editor in the option panels

This section gives a detailed description of the customization panels found in the **Options**[p.60] menu. These are separated into three groups—the *generic panels* (**files**[p.78] and **interface**[p.80] *mode-specific panels* (**ruler**[p.66], **language**[p.68], **shortcuts**[p.70], **fonts**[p.72], **size/gadgets**[p.74], **colours**[p.76]) and **commands**[p.82]) (this one is in the **Edith**[p.48] menu), and *mode-specific switches* (**indent, wrap, brackets, right-align**[p.60]).

7.1 Text window modes

A *text window mode* is a class of text windows that has its own behaviour and its own set of configurable options.

The three *preset* text window modes have a fixed purpose; these are **default** (Default setup), used for editing 'generic' files. **binary** (Binary editors) for *Open binary...* (**File menu**[p.50], Shift Ctrl O) and **browser** (Info browsers) for the info browser (*Edith info index* (**Edith menu**[p.48], Ctrl I), *System manual...* (**Browse menu**[p.53], Ctrl M), *New view on text...* (**Windows menu**[p.62], Alt N) etc.)

For the other 8 modes, you can choose whether these modes should have individual parameters for e.g. colour settings and dimensions. When a text mode does *not* have individual parameters, the ones for the *Default setup* apply.

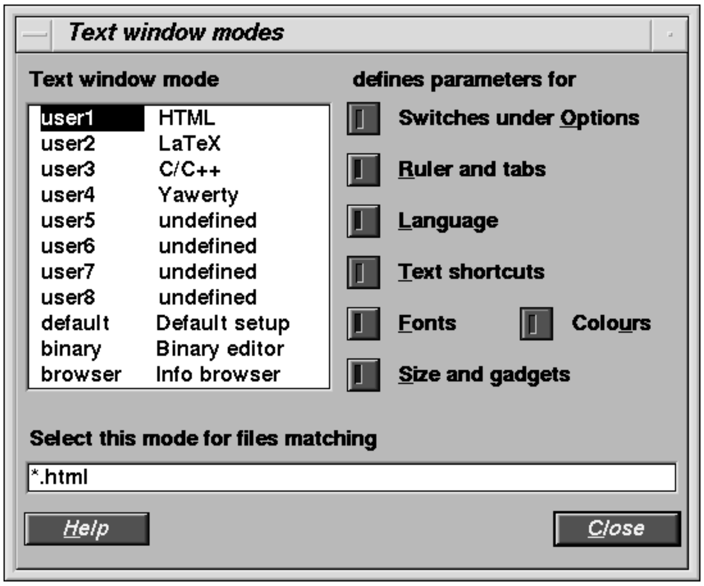

Figure 10: The text window modes panel

Select *Text window modes...* (**Windows menu**[p.62], Shift Ctrl M) to open the *text window modes* panel. This panel lets you manage the eight user defined text window modes, and three predefined types of text windows.

In addition to switching on or off parameter sets for text window modes, you can specify, for each non-preset mode, a *name* and a set of *file masks*. The latter are supposed to match the types of file you most frequently edit; modes shipped with the initial **Edith** configuration are user1: LaTeX, user2: **HTML** and user3: C/C++, which provide **text shortcuts** p.70 for three popular computer languages. Files matching the specified masks will automatically be opened in the corresponding user-defined text mode, and hence get their own colouring, font, editor parameters and shortcut sets. User defined modes are automatically selected on loading a text, based on the name of a file (e.g. in a mode for HTML editing the triggering file mask could be `*.htm*`). If no name is matched, the **default** mode will be selected.

The number of language-specific settings is still limited, but will be extended later. Long term plans are plug-ins for indentation rules, bracketing and quotes, and text macros.

7.2 The text ruler panel

Select *Text ruler...* (**Options menu** p.60, Shift Ctrl R) to open the *text rulers* panel (values related to the text ruler): the positions of the *editor* tab stops and *true* tab stops, the *right margin*, and the way the Tab key works.

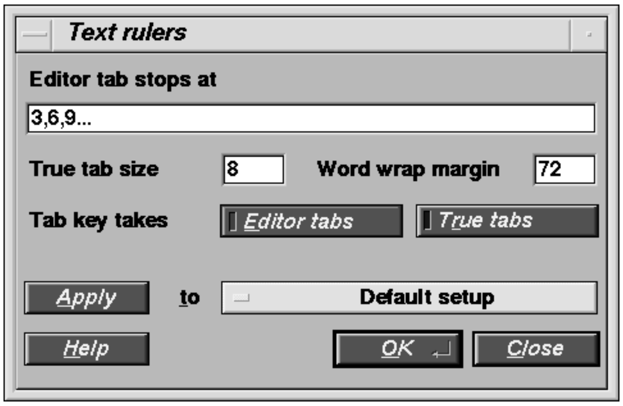

Figure 11: The text ruler panel

INDENTATION AND THE TAB KEY

A ruler has two sets of tab stops. The *editor tabs*, typically at equal distances of 3 or 4 columns, and the *true tabs*, typically at equal distances of 8 columns. There are two editing modes.

Tab key takes editor tabs (default)
When the Tab key is pressed, blanks are inserted up to the next editor tab stop.

Tab key takes true tabs
When the Tab key is pressed, a true tab (ASCII 9) is inserted into the text, resulting in a single blank space stretching to the next true tab stop.

In both of the modes, Shift Tab performs what Tab would do in the other mode. The tab key mode is set in the text rulers panel.

The problem with true tabs is obvious: text writers have different indentation preferences, so looking at one text, a different true tab distance may be necessary than when looking at another. The recommended way to edit text is therefore to use editor tabs, while keeping a standard true tab distance of 8 columns for backward compatibility with text writers who still use true tabs. Using an occasional true tab may be unavoidable (e.g. in `make` files); it can be entered with Shift Tab, and will, in the standard **text display** p.72 settings, appear as a dotted stretched space.

THE RULER ON TEXT WINDOWS

The text ruler is present on text windows when selected in the **size and gadgets** p.74 panel. The ruler shows the positions of the editor tabs (black) and true tabs (blue), and indicates the columns between column 0 and the **right margin** for text formatting in yellow. The tab stops in the lower row are taken by the Tab key, the stops in the upper row by Shift Tab.

Grabbing the ruler with the left mouse button will perform horizontal scrolling. Use the middle mouse button to set, delete or move an editor tab. Use the right button to change the right margin for text formatting. Double click on the ruler to open the **text rulers** panel.

AN ADDITIONAL NOTE

See also the **language options** p.68 panel, where certain non-blank characters can be specified to be counted as indentation. The Escape key triggers a **back-indent** p.22 function which can be thought of as the inverse of Tab.

7.3 The language options panel

Select *Language options...* (**Options menu** [p.60], Shift Ctrl L) to open the *language options* panel. in which the behaviour of the basic text editing functions can be set individually for a selected number of **text window modes** [p.64].

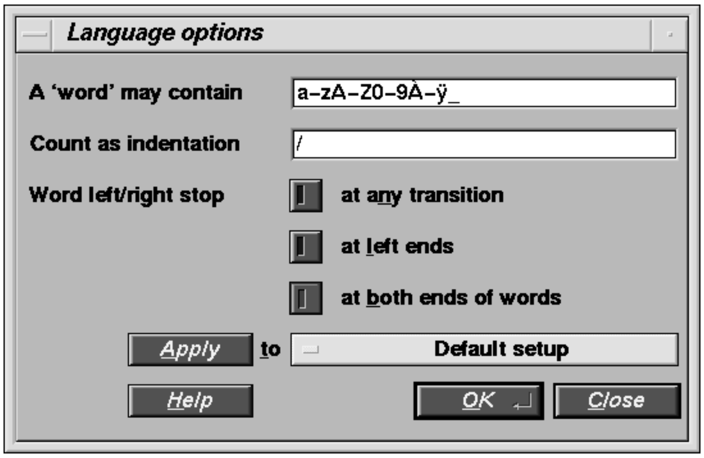

Figure 12: The language options panel

The panel lets you specify two character sets; one defines which letters, numbers etc. make up a **word**. The other, *count as indentation*, which nonblank characters are to be considered equivalent to **white space** (the space character, tab, carriage return and line feed are always in this set, so they don't need to be specified).

- ▷ The definition of a word in the language options panel applies throughout the editor, except for the purely text-oriented *Format paragraph* (**Text menu** [p.59], Alt Shift P) and *Auto wrap* (**Options menu** [p.60], Alt Shift W), which distinguish only between space and non-space. *Format paragraph* does respect a double space after a sentence-ending period.

  - To include the `-` sign as an in-word character, make sure to have it preceded by a back-slash: `\-`. The un-escaped dash character is used to include *range expressions* like `a-z`.

- ▷ Comment prefixes in computer languages, such as `**` in **C**, `//` in **C++**, # in shell scripts, `%` in LaTeX should be set to 'count as indent'. End commented paragraphs with an empty line, so that *Format paragraph* (**Text menu** [p.59], Alt Shift P) will format them correctly. In LaTeX, make inventive use of lines containing just a `%` symbol, so that *Format paragraph* will not scramble e.g. mathematical formulae that immediately follow plain text.

The characters of the ASCII/ISO set are now divided into *three sets*: the in-word characters, the white space characters and *interpunction* (everything that is not in the other sets).

You can now select how *word left*, *word right*, *delete word* and *backspace word* should behave in terms of these three character classes (See also **Moving around** p.20).

7.4 The text shortcuts panel

Select *Shortcuts...* (**Options menu** p.60, Shift Ctrl S) to open the *text shortcuts* panel. Text shortcuts can improve your editing speed considerably for certain computer languages, but only for certain ones; especially languages such as **HTML** and LaTeX.

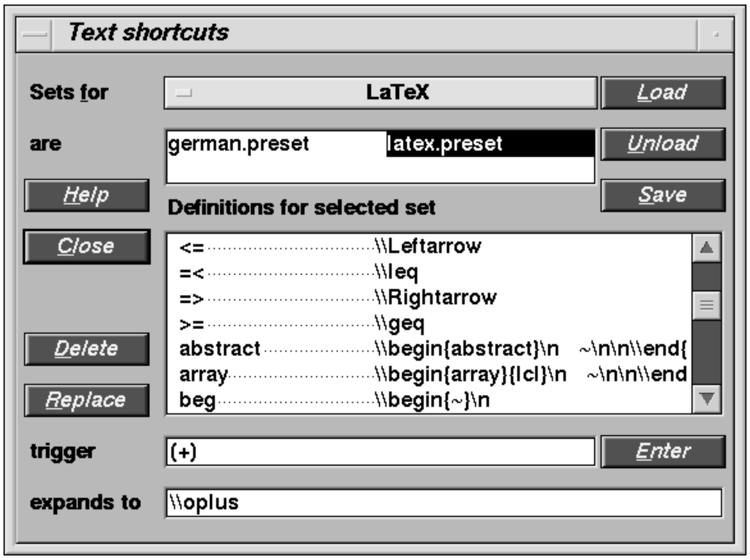

Figure 13: The text shortcuts panel

A shortcut consists of a ‘trigger’ and an ‘expansion’. Type the trigger in a text window, and hit Escape with the cursor exactly after the trigger. The trigger is removed and replaced with its expansion.

The expansion may contain plain text, escape sequences (`\120` or `\xa6`) for special characters, newlines (`\n`), a tilde (`~`), and a question mark (`?`). The tilde is not part of the expansion, but indicates the point where the cursor should be put after expansion. A question mark will caus a box to pop up on shortcut expansion. What is typed in the box gets substituted for every occurrence of `?` in the expansion.

To get a normal tilde, question mark or backslash, use `\~` or `\\`.

The best way to learn how to use shortcuts is to look at the example sets (browse through the predefined sets for the different modes as defined in the shortcut panel). The supplied shortcut sets only serve as examples.

Every text mode has up to four shortcut sets, which are saved in separate configuration files, so that they can be shared between different text modes. Shortcut sets are saved automatically on *Save options* (**Options menu** p.61). When a predefined shortcut set is changed, the suffix `.preset` is removed from its name.

7.5 The text window display/fonts panel

Select *Text display/fonts...* (**Options menu** p.61, Shift Ctrl D) to open the *text display/fonts* panel. The panel is primarily a font selector for text windows, but it also lets you set the fonts for the info browser and dialog panels, and a number of details of how white space is displayed in text windows.

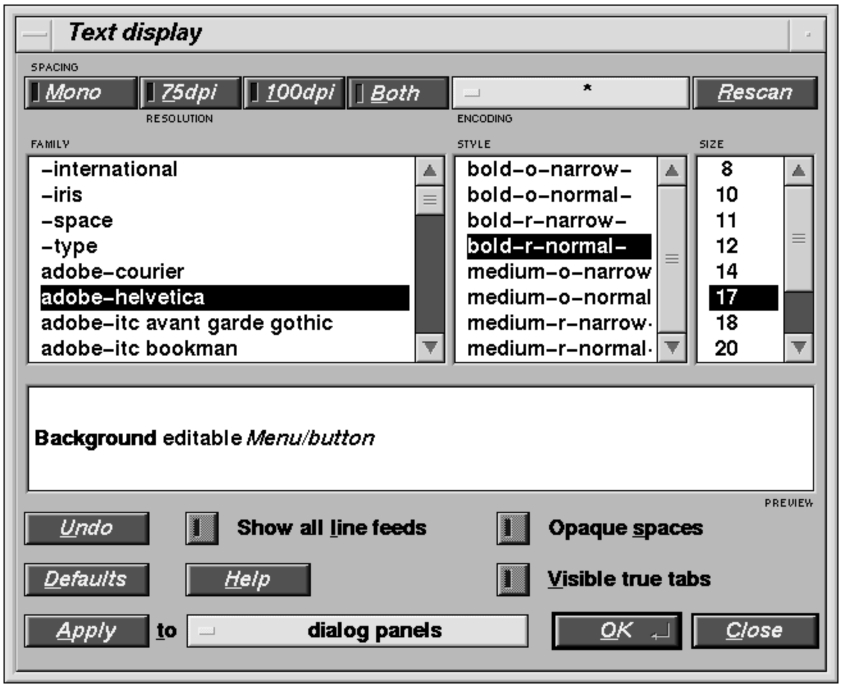

Figure 14: The text window display/fonts panel

The information displayed in the text display panel can be set individually for each **text window mode** p.64 and for **dialog panels**. Switch modes using the menu at the bottom of the panel.

The top row of the panel contains **global** parameters for the selected font set. After changing these values, *Rescan* needs to be clicked. The displayed font family names are restricted to the values for **spacing**, **resolution** and **encoding**; the displayed styles are those available for the selected family; the sizes shown are those for the selected family in the selected style.

The **Undo** button will undo the changes you made to the font set since the last time **Apply** or **OK** was selected; **Defaults** will restore the built-in defaults (Adobe Helvetica/Lucida Typewriter).

To transfer a font set from one mode to another, select the first mode, switch to the other, select **Undo**, then **Apply**.

TEXT WINDOW SPECIFIC SETTINGS

Selecting *display true tabs* will cause true tabs (ASCII 9) in the text to be displayed as thin dotted lines, so as to distinguish them visibly from sequences of space characters (ASCII 32). This option is switched on in the default configuration.

Selecting *display spaces* will cause space characters to be displayed as ‘cups’, so they can be counted and distinguished from true tabs; also useful if space characters at the end of a line should be distinguishable.

When *display all line feeds* is selected, all line endings will be displayed symbolically (as LF, CR/LF or CR). When it is not selected, this is only done for lines ending in a non-unix style line feed (CR/LF for an **MS-DOS** file, CR for one created on a **Macintosh**).

Display line numbers shows line numbers at the end of lines (except for lines which are longer than the window can display).

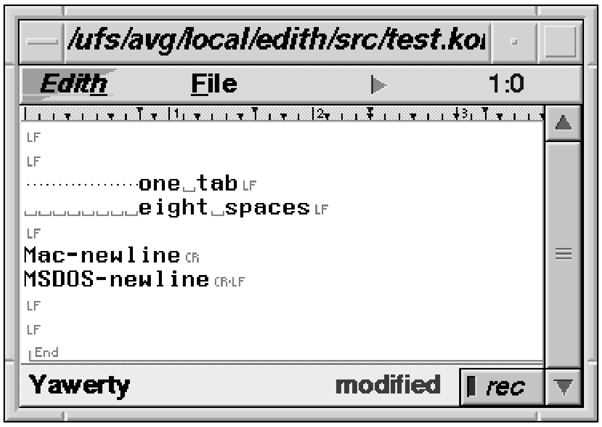

Figure 15: A window with all except show line numbers switched on

7.6 The text window size and gadgets panel

Select *Size and gadgets...* (**Options menu** [p.61], Shift Ctrl Z) to open the *text window size and gadgets* panel. It contains the parameters that determine the size and level of decoration of the windows for a selection of **text window modes** [p.64].

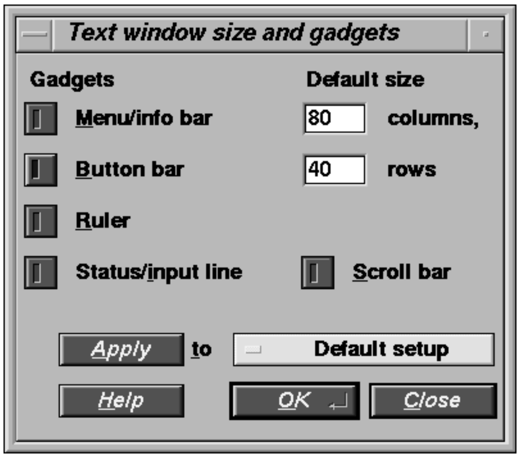

Figure 16: The size and gadgets panel

The *default size* specified in the dialog is set when a text is opened, and is the default value for *Resize...* (**Windows menu** [p.62], Ctrl Z).

The *gadgets* include a **text ruler** [p.66], the **menu bar**, a **button bar**, the **scroll bar** and the **status/input line**.

Switching off these gadgets will give you more space in text windows. However, it also means less flexibility, and less information. The **menu bar** [p.46] shows the text position. The **scroll bar** [p.20] indicates the stored **text markers** [p.55]. The **status line** gives regular informational messages, and speeds up entry of search targets, line numbers, and permanently indicates the editing mode and the state (unchanged/modified) of the text.

7.7 The text window colours panel

Select *Colours...* (**Options menu** [p.61], Shift Ctrl C) to open the *text colours* panel.

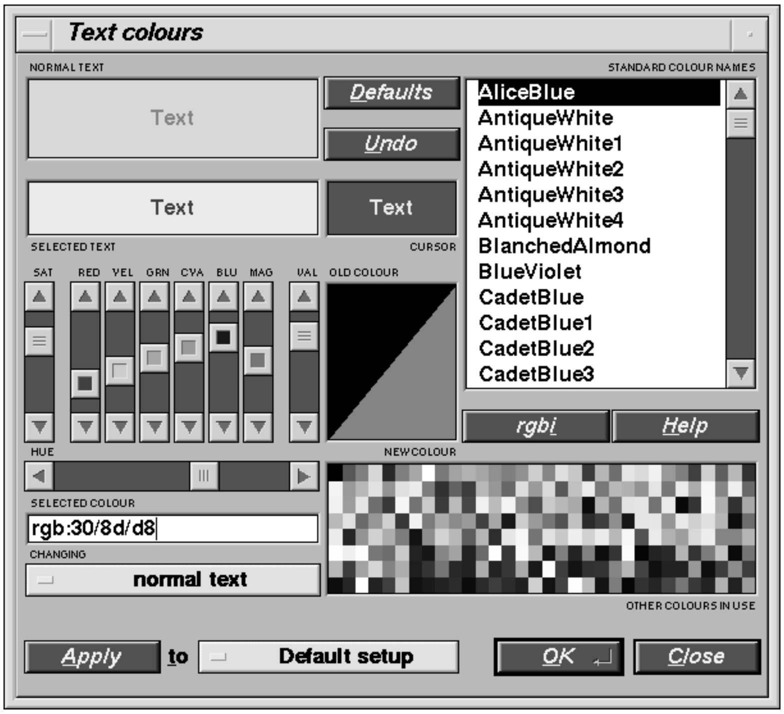

Figure 17: The text colours panel

Colours can be specified for selected text window **modes** [p.64]. Switch through the modes using the popup menu at the bottom of the text colours panel.

After selecting a text window mode, select the particular colour you want to edit through the changing... menu or by clicking the mouse in one of the preview fields at the top of the panel.

There are then several ways of changing the selected colour.

- ▷ select a colour name in the list at the top right of the panel.
- ▷ type a (partial) colour name and hit the Tab key.
- ▷ move the red/yellow/green/cyan/blue/magenta sliders. Moving **red**, **green** or **blue** will have an immediate effect on *three* of the sliders; moving **yellow**, **cyan** or **magenta** will cause all but one, the ‘opposite colour’, of the six sliders to move.
- ▷ move the hue/saturation/value sliders. The *hue* is the pure colour nuance, the *saturation* controls the ‘greyness’ of the colour, and the *value* is its brightness.

▷ point and click on a colour in the field at the bottom right of the panel to select a colour in the ‘default colourmap’ for your display (these include colours that are not used by Edith). This selection is not useful on true colour displays.

After editing all the components you wanted to change, click Apply, or OK (OK will close the panel).

The Undo button abandons the changes to the colours made since you last clicked Apply or OK or switched modes. It can be used to copy a set of colours from one editing mode to another. Defaults will revert to the built-in default colours for the selected text window mode (antique white for text modes, bordaux red for binaries, and grey for the info browser).

TECHNICAL DETAILS

Please note that for every additional colour you choose, several new colours may be allocated, because a colour may need to be displayed in a number of levels of intensity for e.g. shadow effects. The number of colours your screen can display is probably limited, therefore other applications may suffer when **Edith** reserves too many colours. **Edith** is fairly greedy as far as colours are concerned; up to 40 colour shades are allocated with the supplied default configuration. It may help to start other colour-hungry applications before starting the Edith pro server.

Edith will hardly ever suffer from the behaviour of other applications, because on 8-bit (PseudoColor) screens, **Edith**’s colours are allocated *both* in the default colour map and in a private colour map. This way, **Edith** gets all the colours it needs, with a minimum of colourmap flashing as you change from **Edith** windows to those of other applications.

On displays with multiple colour maps (currently only Silicon Graphics), **Edith** will make use of a private colourmap only. When on a black and white display or in extreme situations, **Edith** will approximate unavailable colours by stippling with black and white.

If you are experiencing colour differences because you are using **Edith** on X displays from different manufacturers (e.g. a Sun and a Silicon Graphics), then click the rgbi button to convert the rgb style colours generated by the sliders to the more or less device-independent *red-green-blue-intensity* notation. This button only affects one colour at a time.

7.8 The file options panel

Select *File options...* (**Options menu** [p.61], Shift Ctrl F) to open the *file options* panel, which specifies the behaviour of **Edith** when loading and saving text files.

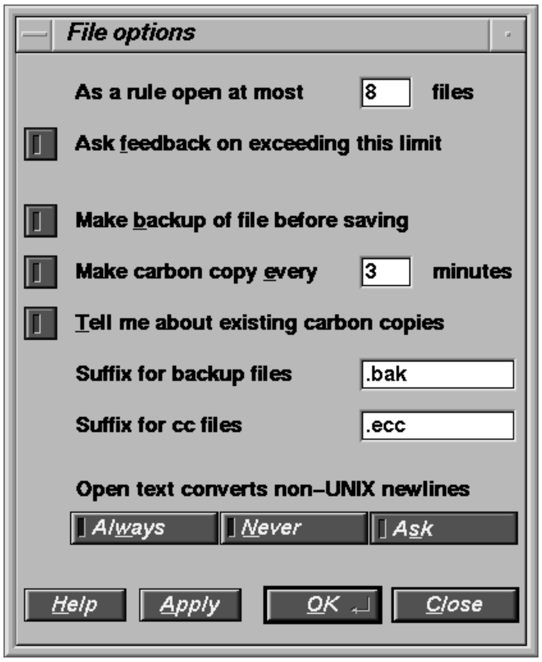

Figure 18: The file options panel

The number of files opened in one go (usually from the command line) can be limited. When you attempt to open more files than the specified limit, you can choose for remaining files to be *queued* immediately (see below), or alternatively for the editor to ask for feedback.

The automatic **backup** and interval *carbon copy* (see below) functions can be switched on and off in this panel. Furthermore, you can specify how backups and carbon copies should be named, and at which interval carbon copies should be written.

Normally, you are informed when there is a carbon copy of a file when you attempt to open it. This feature can be switched off in the file options panel.

Finally, you can choose whether non-UNIX newlines (as used on Macintosh and MS-DOS systems) should be automatically converted when loading a text.

THE INPUT QUEUE

In the file options panel, you can specify a maximum to the number of text files that can be opened for editing in a single command, a single command usually being a call to **Edith** from a terminal window. When a command exceeds this number, an alert panel will pop up asking whether some of the requested files should be put 'on hold' in the *input queue*. Files that are on hold can be opened with *Open next* (**File menu** p.50, Ctrl N). **Edith** will, if so chosen in the options panel, also notify you of files in the input queue as soon as all other editor windows have been closed.

CARBON COPIES

Apart from the usual *backups*, **Edith** has an additional safety measure: *carbon copies*. These are copies of files, like backups, but they are made either on your request (when closing a window whose contents have changed), or when the **Edith** system unexpectedly terminates, e.g. by a `kill` command or broken display connection. Another option is to have carbon copies made at set intervals. This can all be configured in the file options panel.

You will be informed of the presence of a carbon copy when opening the original file. On saving the file, carbon copies will be deleted.

7.9 The interface options panel

Select *Interface options...* (**Options menu** [p.61], Shift Ctrl I) to open the *interface options* panel and customize the general looks and sounds of the editor.

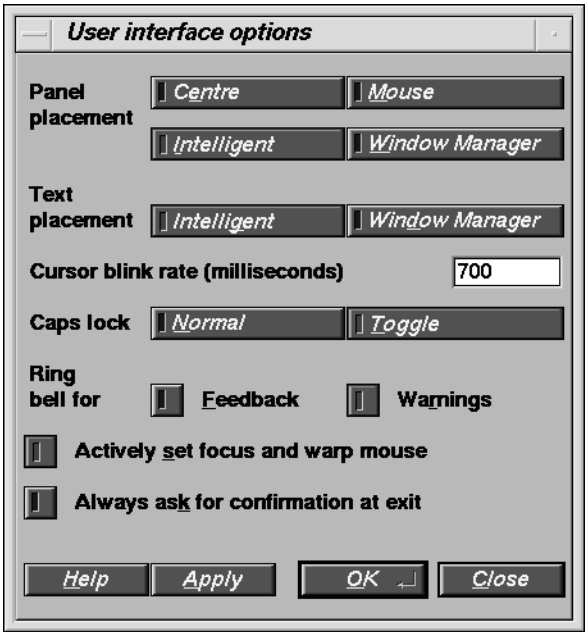

Figure 19: The interface options panel

A number of *Panel placement* and *text placement* options can be selected; the safest setting is *window manager*, and is necessary for some window managers. It means that placement of all windows is identical to what you would expect for simple **X** applications. **Centre** will centre panels on the screen. **Mouse** will attempt to place panels near the mouse pointer. **Intelligent** placement puts windows next to other windows, in those places where they cover the least amount of space in other windows.

The *cursor blink rate* for both texts and dialog panels can be set in milliseconds. The value zero means that the cursor will not blink.

Furthermore, you can choose how often you want to hear a *bell sound*. This is especially useful when the **status line** [p.74] is switched off. Select

feedback
: when you want to hear sounds to confirm the operations *Copy* (**Edit menu** [p.56], Alt C) and *Put marker* (**Mark menu** [p.55], Alt P).

warnings
: when you want to hear warning sounds in the following cases: search failure,

bracket mismatch, warning bell near end of line (not in **auto wrap** p.60 mode), and a few more rare ones.

When *actively set focus and warp mouse* is selected, **Edith** will attempt to keep the 'input focus' where you are most likely to want to type, by both requesting the focus and warping the mouse around. When this feature is switched on, **Edith** can be operated practically without using the mouse, even when many different windows are involved; however, it may not work properly with all window managers.

Finally, you can choose here whether **Edith** should *always* ask for a confirmation when you select *Quit* (**File menu** p.51, Shift Ctrl Q). When the configuration (everything under the **options** p.60 menu) has changed however, you will always get a warning box before **Edith** really exits.

HANDS-FREE WINDOW MANAGEMENT

If your window manager acts favourably, you should in principle select *intelligent placement* and *actively set focus and warp mouse* in the *interface options* panel. This reduces the number of cases you will need to move the mouse to the windows you want to work in (this is often annoying in graphical-user-interface applications that pop up lots of informational message boxes).

You can cycle between different windows using *Cycle texts* (**Windows menu** p.62, Ctrl Tab); text windows can be arranged in a tiled pattern using *Tile vertically* (**Windows menu** p.62, Shift Ctrl V) and *Tile horizontally* (**Windows menu** p.62, Shift Ctrl H). Windows can be moved around the screen by holding down Ctrl and pressing one of the arrow keys on the numerical pad. On some systems, *num lock* has to be selected for this function to work (experts may work around this using `xmodmap`, which can be recommended especially for **Linux** users).

SUPPORTED WINDOW MANAGERS

Edith Pro has been tested under various methods of window placement with **4Dwm**, **fvwm 1.2**, **mwm**, **twm** and **tvtwm**. Intelligent placement is known not to work with **vtwm**. For these functions to work properly with all window managers, a certain amount of knowledge about the behaviour of various window managers is necessary. Therefore if this behaviour changes in future versions, this might have consequences for some of **Edith**'s window placement modes. Please report any problems with window managers immediately so a fix can be considered.

7.10 The external commands panel

Select *External commands...* (**Edith menu** [p.48], Shift Ctrl X) to open the *external commands* panel, where you can enter shell commands that get an entry in the **Edith menu** [p.48], and can be assigned a function key.

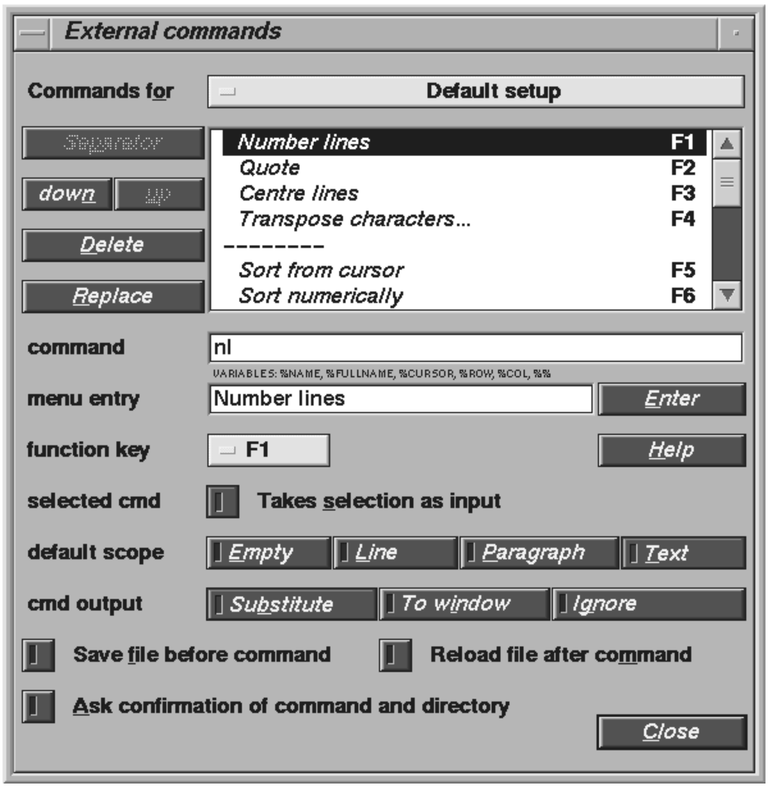

Figure 20: The external commands panel

Commands can be defined for each **text window mode** [p.64]. The commands for *Default setup* are always available, and are assigned function keys F1-F20. Commands for the other modes, such as LaTeX, are available only in windows for that mode, and have function key combinations Ctrl F1 - Ctrl F20. In the commands panel, switch between these modes through the pop-up menu at the top of the panel.

The scrolling **list** in the top right corner of the panel shows the names of the commands in the order in which they will appear in the *Edith* menu. The order can be changed using the down and up buttons; use the separator button to put sets of external commands of a similar nature into groups.

The lower half of the panel shows the parameters for the command selected in the list. *To replace a selected command entirely* by a new definition, change the values and click replace. If either the name or the command line have been changed, but one of them remains the same, use Enter. Use Enter also to enter a new command that is not yet in the list.

The meaning of the parameters in the panel is best explained by looking at the commands that are automatically installed when you first use **Edith**.

Commands for Default setup

Number lines (F1)
: On text windows only; numbers selected lines, or the entire text if there is no selection.

  Calls the **nl** program, that takes a number of lines as its standard input, and produces on its standard output the same lines prefixed with numbers. The switch *takes selection as input* is selected, so when calling this command from the *Edith* menu or by hitting F1, the selected text is sent to the standard input of `nl`. *Default scope* is set to *text*, so when there is no selection, the entire text will be selected and sent to `nl`. *Cmd output: Substitute* is selected so that the output of `nl` will be substituted for the selection.

Quote (F2)
: Quote selected text (as in an email message), or a paragraph if there is no selection.

  Definition is nearly identical to *number lines*, except that default scope is now *paragraph*. The command itself is not a simple UNIX command like **nl**, but an **awk (1)** script, that prefixes its input lines with >. On **Linux** systems, check **gawk (1)**.

Centre lines (F3)
: Centre selected lines; default scope: paragraph.

  Analogous to *quote*.

Transpose characters... (F4)
: Calls the **tr** program on selected lines of text; defaults to a paragraph. A box pops up for the user to confirm the transposition patterns. Default action is to swap case.

  By default, this command swaps upper and lower case. However, because *ask confirmation of command and directory* is selected, a **confirmation box** will pop up when *Transpose characters* is selected or F4 is pressed. In this box, the *transpose pattern* `'a-zA-Z' 'A-Za-z'` passed to the `tr` program is selected. This preselection is specified in the *external commands panel* by the use of the special variable `%cursor` in the command's definition. `%cursor` marks the cursor position, or when used twice, the left and right end of the selection in the confirmation box.

The transpose pattern consists of two sets of characters; characters in the first set will be replaced by those in the second. See **tr (1)** for details. Because the sets are preselected, you can change them immediately by typing two new sets. What you have typed to replace the preselected patterns will automatically pop up again the next time you call *transpose characters*, provided that you made changes *only* to the preselected text.

Following a good custom, the *menu entry* for this command ends in an ellipsis (**...**), to indicate that a panel will be opened when the command is selected, requiring further information from the user.

Sort from cursor (F5)

Sort selected lines from the current cursor column. Using this function requires a bit of practice. Select 10 lines with the mouse, move inside the lines *with the cursor keys*, move to the column which contains the data to be used as the sorting key, and hit F5. Default scope is entire text.

Calls **sort +***number*. New is the use of the variable `%col`, which is replaced with the current cursor column. A variable `%row` is also available.

Sort numerically (F6)

Sort selected lines numerically (from beginning of line). Default scope is entire text.

Remove duplicate lines (F7)

Removes consecutive identical lines, by calling the program **uniq**.

Evaluate expression... (F8)

Evaluate the selected numerical expression (e.g. `(3 + 4) * 5`), and display it in a browser window. If there is no selection, looks at one line.

Calls the program **bc**. New is the setting *cmd output: to window*, that instead of replacing the selection, sends the output of the command to a new (browser) window.

Following a less obviously good custom of the developer of **Edith**, the menu entry of this command *also* ends in an ellipsis, even though the window that pops up when the command is executed does not require user interaction. But the command definitely causes a new window to be opened, and that too deserves some form of indication.

Count words... (F9)
Count the number of selected bytes, words and lines, or when there is no selection, the number of bytes, words and lines in the whole text. Calls the **wc** command.

Spell... (F10)
A simple spell checker. Produces a list, in a new window, of words which are in the selected text, but do not appear in the standard system dictionary.

Calls the **ispell (1)** command with the option `-l`, sorts the result and removes duplicate lines. This is equivalent to `ispell`'s predecessor **spell (1)**; however, `spell` is no longer present on all UNIX systems (Linux slackware).

Fgrep and open... (F11)
Pops up a box where a search string and a set of files is preselected. Type the search string and the file names as you would do for **fgrep (1)** , and hit Enter. Edith will then open all of the specified files that contain the search string, with the cursor at the first occurrence of the string. You can proceed searching for the string with *Next target below* (**Browse menu** p.52, Ctrl G).

Print... (F12)
Calls **lpr**, with a confirmation. New is *cmd output: ignore*.

Commands for HTML

Preview... (Ctrl F11)
Opens an **Edith** browser window on the HTML file you are editing.

Commands for LaTeX

Comment out (Ctrl F1)
Puts comment signs (%) before the selected lines.

Comment in (Ctrl F2)
Removes comment signs (%) before the selected lines.

LaTeX... (Ctrl F5)
Runs LaTeX on the current file (but pops up a box where you can change the name of the source file and the directory).

New is the use of *Save file before command* that causes the text in the window from which this command is selected to be saved before `make` is executed.

BIB*TEX*... (Ctrl F6)
Runs BIBTEX on the current file.

Open log file... (Ctrl F9)
Opens the log file for the TEX file you are editing.

Preview... (Ctrl F11)
Opens **xdvi (1)** on the current TEX file.

Print... (Ctrl F12)
Prints the current TEX file using **dvips (1)** and **lpr (1)**.

Commands for C/C++

Comment out (Ctrl F1)
Puts comment signs around the selected lines.

Comment in (Ctrl F2)
Removes comment signs around the selected lines.

Make... (Ctrl F5)
Calls **make (1)**, with a confirmation box; the current file is saved before the command is executed.

Make clean... (Ctrl F6)
Runs `make clean`.

Open header file... (Ctrl F9)
Opens the header file for the the **C** file you are editing (*i.e.*, the suffix `.c` is replaced with `.h`).

Open C file... (Ctrl F10)
Opposite of *open header file*.

CONFIRMED COMMANDS

Some of the commands do not execute immediately, but pop up a box asking you to confirm the command and the directory where it is executed. The first time you run the command, its directory is set to the directory belonging to the window you call it from. On subsequent times you run it, the previous directory will be preselected in the confirmation box. If you make changes to the command itself however, these are *not* remembered, unless part of the command line was preselected using `%cursor`, in which case modifications to the preselected part of the command are stored.

VARIABLES

The variables allowed in the *command* field have been discussed at length in the examples. A summary:

`%name`
: The short name of the text file; left unexpanded if the command is not called from a text window. In the case of a command with *ask confirmation of command and directory*, the name is relative to the last given directory.

`%fullname`
: The full path name of the text file

`%cursor`
: Useful only if *ask confirmation of command and directory* is switched on. Specifies the position of the text cursor within the confirmation box. If used twice, specifies the beginning and the end of a preselected part of the command (usually a parameter).

`%row`
: Expands to the current cursor row.

`%col`
: Expands to the current cursor column.

`%%`
: Expands to a single `%`.

7.11 Saving options; configuration files

Select *Save options* (**Options menu** p.61) to save the entire **Edith** configuration, including the selected shortcut sets, and the editor/appearance settings for each of the defined **text window modes** p.64. Note that the configuration may be lost without notification when you *Quit* (**File menu** p.51, Shift Ctrl Q) the application.

ORGANIZATION OF FILES WRITTEN BY SAVE OPTIONS

Save options (**Options menu** p.61) will write the files listed below. Please do not edit these files; everything can be customized in the application panels. **Edith** cannot be guaranteed to work properly after 'manual tampering' with the files.

However, knowing what files are written and what for may help you clean up old, unwanted configuration files.

`~/.edith/options.ad`
: General editor setup (options menu), names and selection of defined parameters for editor modes.

`~/.edith/`*mode*`.ad`
: Sub-setup for *mode*, where *mode* is the fixed, short mode *identifier* (e.g. `user2`) of a **text window mode** p.64.

`~/.edith/`*setname*`.esc`
: Shortcut definitions for set *setname*.

`~/.zfc-defaults`
: Dialog panel setup (shared with other applications built with the ZFC user interface toolkit).

The following file is not saved by *Save options*, but will be written automatically when you exit the **Edith** program.

`~/.edith/setup.ad`
: Info and file selector bookmarks, search parameters, last visited directories.

The private info index is also stored here.

`~/.edith/private_index.html`
: Your **private index page** p.43.

A Troubleshooting, Questions and Answers

I clicked "no further confirmations" in a warning panel, and indeed I never get a warning again, but how do I get the warnings back if I want to see them again?
Try *Interface options...* (**Options menu** [p.61], Shift Ctrl I) or *File options...* (**Options menu** [p.61], Shift Ctrl F).

I open a C file, but I get a window in 'default setup' mode
Is your C file a backup? If it doesn't end in `.c` or `.h`, it is not recognized as a C file. Switch to C editing mode using *Switch mode...* (**Windows menu** [p.62], Ctrl S).

I can't edit the size and gadgets for mode LaTeX. I get to see the values for "default setup"
Don't forget to tell **Edith** that your LaTeX mode has its own values for *size and gadgets*, under *Text window modes...* (**Windows menu** [p.62], Shift Ctrl M).

I am using *auto wrap*, but sometimes it doesn't seem to do it's work
Auto wrap only wraps when the cursor is beyond the right margin; it is not a complete text formatting function. To reformat a paragraph after inserting or deleting words in the middle of a line, use *Format paragraph* (**Text menu** [p.59], Alt Shift P). You may need to switch off *Format right-aligns* (**Options menu** [p.60], Shift Ctrl P).

What is the purpose of the rec button in the status line?
It indicates when the *dribble recorder* is on; see *Repeat dribble* (**Edit menu** [p.57], Ctrl R).

What is the purpose of the black arrows in the scroll bar?
They are *position markers*, see *Put marker* (**Mark menu** [p.55], Alt P), *Back* (**Browse menu** [p.52], Alt ⇐), *Forward* (**Browse menu** [p.52], Alt ⇒).

Edith crashed while I was editing an important file
If **Edith** hangs, kill the program by sending it a `TERM` signal, e.g. by typing `killall -15 edith`, or by destroying a window. **Carbon copies** [p.79] will then be written for any files that were modified when the **Edith** server was killed. Then re-open the important files; **Edith** will offer you the option to load the carbon copies instead of the old originals.

Edith refuses to save a file, but it has the required permissions
Sometimes **Edith**'s underlying file system still thinks a file is unreachable,

even though the problems it has noticed earlier have been fixed. Clicking *retry* in the box claiming that the file is unwritable should help. Note that once **Edith** notices that a file is not writable, the normal *Save* (**File menu** p.50, Alt S) function will behave like *Save as...* (**File menu** p.50, Alt Shift S).

I opened a large number of files, and said yes when Edith offered to put some of them in the *queue*. I finished editing the first set of files; how do I open the others?
Open single files from the queue using *Open next* (**File menu** p.50, Ctrl N). When *all* open text files (excluding browsers) are closed or iconified, **Edith** will pop up a message panel asking you if the next few files should be opened.

I installed `edivisual`/`edimore` as an editor/viewer and it complains that some files dont exist'
Make sure that the **Edith** server is running on the same host as the program that called the editor. Files like `/tmp/edimore.12647` or `/tmp/snd.23847` are not always readable from remote machines, even if these machines are on the same network as your workstation.

I installed `edivisual` as my mail editor; when I work on a modem/remote shell connection, I get (a) cannot connect to display ... or (b) I get `vi`, and I would prefer `Emacs`
When working on a slow connection, make sure that you do **NOT** have the environment variable `DISPLAY` set. The environment variable `EDITOR` can be used to select an editor different than **vi** for when there is no X display.

A window I just opened has the wrong size
This is a (known) bug, that occurs only rarely; the windows seem to always have the smallest possible size. Fix it with Ctrl Z then Return.

B Keyboard Glossary

Other sources of information: See **moving around** p.20 and **selection** p.24 for lists of key combinations on editor windows; see **the graphical user interface** p.12 for a general explanation of operating **Edith**'s user interface.

RATIONALE

A way of getting to appreciate the choice of the menu shortcuts is to remember that

- ▷ combinations with Alt only will generally apply to the *current* (text or panel) window only. The Ctrl key on the other hand is usually assigned to an operation that can be selected *anywhere* in the application.
- ▷ only less frequently used combinations will have Shift
- ▷ some keycodes in the **Edit menu** p.56 and **Browse menu** p.52 have alternatives that resemble those of the **VI** text editor, but have Ctrl added. These are e.g. Ctrl ., Ctrl %, Ctrl /, Ctrl ?.
- ▷ conformance to **Emacs** key codes is severly limited. There is *Erase line* (**Edit menu** p.56, Ctrl U), and a number of single key strokes on browser windows. People have asked for more, but as with the **VI** codes, most of these are clumsy and old-fashioned, because they date back to the time that computer keyboards didn't even have such things as arrow keys.

OTHER KEYS

As far as possible, other keys labelled with special functions, such as the **Sun** L keys, should work. If they don't, do not hesitate to **write** to the author so that they can be included.

Linux users who haven't done so already, may need to put a command like

```
xmodmap -e 'keycode 22 = BackSpace BackSpace'
```

in their `.xsession` or `.xinitrc` file. A more complete fix to the rather limited Linux keyboard setup is available from **ZFC** on request. On (some?) PC keyboards, a number of key combinations with both Shift and Ctrl generate a response only when the *right* Shift key is used.

ALPHABETICAL LIST OF KEYBOARD COMBINATIONS

The following is a list, ordered by their position on the keyboard, of key combinations in the menus, and those that apply to text windows only; combinations with Alt may not apply in dialog panels, because they are assigned to buttons.

F1 - F20 External commands (**Edith menu** p.48) for *Default setup*.

Ctrl F1 - F20 External commands (**Edith menu** p.48) for text window modes other than *Default setup*.

Escape back-indent or **shortcut** p.70 expansion. On browsers: *Stop* (**Browse menu** p.52). On panels: abort through a button like Cancel .

Tab To next edit/true **tab stop** p.66. Go to next text input field on panels. Select next link on browser windows.

Shift Tab To next true/edit (alternative) tab stop. Go to previous text input field on panels. Select previous link on browser windows.

Left tab Same as Shift Tab

Ctrl Tab *Cycle texts* (**Windows menu** p.62)

Caps Lock *Additive selection* (**Mark menu** p.55)

Alt 1 *Do dribble 1* (**Edit menu** p.56), also Ctrl 1

Alt 2 *Do dribble 2* (**Edit menu** p.56), also Ctrl 2

Alt 3 *Do dribble 3* (**Edit menu** p.56), also Ctrl 3

Ctrl % *Find bracket* (**Browse menu** p.53), also Ctrl B

Ctrl A *Apropos...* (**Browse menu** p.53)

Shift Ctrl A *ASCII/ISO table* (**Windows menu** p.63)

Alt A *Select all* (**Mark menu** p.54)

Alt Shift A *Begin selection* (**Mark menu** p.54)

B *Back* (**Browse menu** p.52) on browsers

Ctrl B *Find bracket* (**Browse menu** p.53), also Ctrl %

Shift Ctrl B *Cut/replace buffer* (**Windows menu** p.63)

Alt B Open browse menu

Alt Shift B *End selection* (**Mark menu** p.54)

Shift Ctrl C *Colours...* (**Options menu** p.61)

Alt C *Copy* (**Edit menu** p.56)

Alt Shift C *Capitalize* (**Text menu** p.58)

D *Go to info index* (on browsers)

Ctrl D *Next target above* (**Browse menu** p.52)

Shift Ctrl D *Text display/fonts...* (**Options menu** p.61)

Alt D *Delete line* (**Edit menu** p.56)

Alt Shift D *Indent to column...* (**Text menu** p.58)

Ctrl E *Edit source...* (**Browse menu** p.53)

Shift Ctrl E *Execute command...* (**Edith menu** p.48)

Alt E Open edit menu

Alt Shift E *Expand tabs* (**Text menu** p.59)

F *Forward* (**Browse menu** p.52) on browsers

Ctrl F *Search engine...* (**Browse menu** p.52)

Shift Ctrl F *File options...* (**Options menu** p.61)

Alt F Open file menu

Alt Shift F *Filter...* (**Edith menu** p.48), also Control !

Ctrl G *Next target below* (**Browse menu** p.52)

Alt G *Goto line...* (**Browse menu** p.53)

H *Help* (**Browse menu** p.52) on browsers

Ctrl H *Replace* (**Browse menu** p.52)

Shift Ctrl H *Tile horizontally* (**Windows menu** p.62)

Alt H Open **Edith** menu; often: help.

Alt Shift H *Create source tag* (**Mark menu** p.55)

Ctrl I *Edith info index* (**Edith menu** p.48)

Shift Ctrl I *Interface options...* (**Options menu** p.61)

Alt I *Invert selection* (**Mark menu** p.54)

Alt Shift I *Auto indent* (**Options menu** p.60)

Ctrl J *Shift left* (**Text menu** p.58)

Alt J *Shift left one* (**Text menu** p.58)

Alt Shift J *Join lines* (**Edit menu** p.56)

Ctrl K *Shift right* (**Text menu** p.58)

Alt K *Shift right one* (**Text menu** p.58)

Alt Shift K *Kill trailing blanks* (**Text menu** p.59)

Ctrl L *Open location...* (**File menu** p.50)

Shift Ctrl L *Language options...* (**Options menu** p.60)

Alt Shift L *To lower case* (**Text menu** p.58)

Ctrl M *System manual...* (**Browse menu** p.53)

Shift Ctrl M *Text window modes...* (**Windows menu** p.62)

Alt M Open mark menu

Alt Shift M *Show matching brackets* (**Options menu** p.60)

Ctrl N *Open next* (**File menu** p.50)

Shift Ctrl N *New untitled text* (**File menu** p.50)

Alt N *New view on text...* (**Windows menu** p.62)

Alt Shift N *Convert newlines...* (**Text menu** p.59)

Ctrl O *Open text...* (**File menu** p.50)

Shift Ctrl O *Open binary...* (**File menu** p.50)

Alt O Open options menu

Alt Shift O *Sort* (**Text menu** p.59)

Ctrl P *Quick search above...* (**Browse menu** p.52), also Ctrl ?

Shift Ctrl P *Format right-aligns* (**Options menu** p.60)

Alt P *Put marker* (**Mark menu** p.55)

Alt Shift P *Format paragraph* (**Text menu** p.59)

Q *Close window* (**File menu** p.51) on browsers

Ctrl Q *Quick search below...* (**Browse menu** p.52), also Ctrl /

Shift Ctrl Q *Quit* (**File menu** p.51)

Alt Q *Select south-east quadrant* (**Mark menu** p.54), also Ctrl [

Alt Shift Q *Select north-east quadrant* (**Mark menu** p.54), also Ctrl]

Ctrl R *Repeat dribble* (**Edit menu** p.56), also Ctrl .

Shift Ctrl R *Text ruler...* (**Options menu** p.60)

Alt R *Reload...* (**File menu** p.50)

Alt Shift R *Reverse lines* (**Text menu** p.59)

Shift Ctrl S *Shortcuts...* (**Options menu** p.60)

Ctrl S *Switch mode...* (**Windows menu** p.62)

Alt S *Save* (**File menu** p.50)

Alt Shift S *Save as...* (**File menu** p.50)

Ctrl T *Store dribble* (**Edit menu** p.56)

Shift Ctrl T *Trash buffer* (**Windows menu** p.63)

Alt T Open text menu

Alt Shift T *Compress tabs* (**Text menu** p.59)

Ctrl U *Erase line* (**Edit menu** p.56)

Alt U *Undo* (**Edit menu** p.56)

Alt Shift U *To upper case* (**Text menu** p.58)

Shift Ctrl V *Tile vertically* (**Windows menu** p.62)

Alt V *Paste* (**Edit menu** p.56)

Alt Shift V *Overlay* (**Edit menu** p.56)

Ctrl W *Close window...* (**File menu** p.51)

Shift Ctrl W *Close all windows* (**Windows menu** p.63)

Alt W Open windows menu

Alt Shift W *Auto wrap* (**Options menu** p.60)

Shift Ctrl X *External commands...* (**Options menu** p.61)

Alt X *Cut* (**Edit menu** p.56)

Alt Shift X *Delete* (**Edit menu** p.56)

Ctrl Y *Private info index* (**Edith menu** p.48)

Alt Y *Toggle single/column* (**Mark menu** p.54)

Alt Shift Y *Save all texts* (**File menu** p.50)

Ctrl Z *Resize...* (**Windows menu** p.62)

Shift Ctrl Z *Size and gadgets...* (**Options menu** p.61)

Alt Z *Clear selection* (**Mark menu** p.54)

Alt Shift Z *Save selection...* (**File menu** p.50)

Backspace Delete character left of cursor (**Moving around** p.20); *note* on some **Linux** systems the backspace key is, incorrectly, indistinguishable from Delete. Contrary to Delete, doesn't delete the selection in panels.

Shift Backspace Delete to beginning of line (**Moving around** p.20, **User interface** p.12)

Ctrl Backspace Delete word left of cursor (**Moving around** p.20, **User interface** p.12)

Return Create a new line. Exit a panel through an affirmative button (usually OK). In browsers: open selected link.

Ctrl] *Select SE quadrant...* (**Mark menu** p.54), also Alt Q

Ctrl [*Select NE quadrant...* (**Mark menu** p.54), also Alt Shift Q

Ctrl . *Repeat...* (**Edit menu** p.57), also Ctrl R

Ctrl / *Quick search below...* (**Browse menu** p.52), also Ctrl Q

`?` *Help* (**Browse menu** [p.52]) on browsers

`Ctrl` `?` *Quick search above...* (**Browse menu** [p.52]), also `Ctrl` `P`

`Space` *Page down* on browsers

`Insert` *Replace mode* (**Mark menu** [p.55])

`Ctrl` `Insert` *Insert...* (**File menu** [p.50])

`Delete` Delete character right of cursor (**Moving around** [p.20]). In panels, deletes selected text (**User interface** [p.12]).

`Shift` `Delete` delete to end of line (**Moving around** [p.20])

`Ctrl` `Delete` Delete word right of cursor (**Moving around** [p.20])

`Home` Line home (**Moving around** [p.20]), also `Begin`

`Shift` `Home` Extend fragment to line home (**Selection** [p.24])

`End` Line end (**Moving around** [p.20])

`Shift` `End` Extend fragment to line end (**Selection** [p.24])

`Ctrl` `Home` Text home (**Moving around** [p.20]), also `Ctrl` `Begin`

`Shift` `Ctrl` `Home` Extend fragment to text home (**Selection** [p.24])

`Shift` `Ctrl` `End` Extend fragment to text end (**Selection** [p.24])

`Page up` Page up (**Moving around** [p.20])

`Shift` `Page up` Extend fragment one page up (**Selection** [p.24])

`Page down` Page down (**Moving around** [p.20])

`Shift` `Page down` Extend fragment one page down (**Selection** [p.24])

`⇑` Cursor up (**Moving around** [p.20])

`Shift` `⇑` Extend fragment one line up (**Selection** [p.24])

`Ctrl` `⇑` Paragraph up (**Moving around** [p.20])

`Shift` `Ctrl` `⇑` Extend fragment one paragraph up (**Selection** [p.24])

`Alt` `⇑` Display up (**Moving around** [p.20])

⇓ Cursor down (**Moving around** p.20)

Shift ⇓ Extend fragment one line down (**Selection** p.24)

Ctrl ⇓ Paragraph down (**Moving around** p.20)

Shift Ctrl ⇓ Extend fragment one paragraph down (**Selection** p.24)

Alt ⇓ Display down (**Moving around** p.20)

⇐ Cursor left (**Moving around** p.20, **User interface** p.12)

Shift ⇐ Extend fragment one character to the left (**Selection** p.24, **User interface** p.12)

Alt ⇐ *Back* (**Browse menu** p.52)

⇒ Cursor right (**Moving around** p.20, **User interface** p.12)

Shift ⇒ Extend fragment one character to the right (**Selection** p.24, **User interface** p.12)

Alt ⇒ *Forward* (**Browse menu** p.52)

Ctrl ⇐ Word left (**Moving around** p.20)

Shift Ctrl ⇐ Extend fragment leftward by one word (**Selection** p.24)

Ctrl ⇒ Word right (**Moving around** p.20)

Shift Ctrl ⇒ Extend fragment rightward by one word (**Selection** p.24)

Ctrl **Num pad** 7 Move window to top left corner of screen

Ctrl **Num pad** 8 Move window to top of screen

Ctrl **Num pad** 9 Move window to top right corner of screen

Ctrl **Num pad** 4 Move window to left of screen

Ctrl **Num pad** 5 Move window to middle of screen

Ctrl **Num pad** 6 Move window to right of screen

Ctrl **Num pad** 1 Move window to bottom left corner of screen

Ctrl **Num pad** 2 Move window to bottom of screen

Ctrl **Num pad** 3 Move window to bottom right corner of screen

Num pad $\boxed{\text{-}}$ *Smaller font* (**Windows menu** p.62)

Num pad $\boxed{+}$ *Larger font* (**Windows menu** p.62)

$\boxed{\text{Enter}}$ Same as $\boxed{\text{Return}}$, always.

C Glossary of terminology

ASCII
: *American Standard Code for Information Interchange* refers to the standard first 127 characters of the 256 character **ISO** character set, which is compatible with **MS-DOS** and **Macintosh**. A file in **ASCII** is a text which contains plain text without any additional information such as fonts, text rulers or pictures.

Auto indent
: When this function is switched on through *Auto indent* (**Options menu** p.60, Alt Shift I), **Edith** will automatically indent lines created when you press the Return key.

Auto wrap
: When this function is switched on through *Auto wrap* (**Options menu** p.60, Alt Shift W), 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. Switched on off under *File options...* (**Options menu** p.61, Shift Ctrl F). See also *carbon copy*. By default, the name of a backup file is obtained from the original name by adding the suffix `.bak`. Backups are not made for files whose name contains `/tmp/`.

Binary
: A file that contains arbitrary data (not text). See **editing binary files** p.40.

Buffer
: A piece of text, divided into fragments, stored in computer memory, but not part of a text window. **Edith** has two text buffers available to the user, the *cut buffer* and the *trash buffer*. A number of additional buffers exist in the underlying system (for *undo* and selection).

Carbon copy
: A copy of a file saved automatically at regular intervals, when abandoning the contents of a text window, or when the editor exits unexpectedly and files were modified. See **file options panel** p.78. By default, the name of a backup file is obtained from the original name by adding the suffix `.ecc`.

Cut buffer
: The buffer containing data from *Copy* (**Edit menu** p.56, Alt C) and *Cut*

(**Edit menu** p.56, Alt X).

Default text window
A normal window for editing text, whose file name does not match the patterns set for any **user defined text modes** p.64. Text window appearance and ruler settings for *Default setup* apply.

Dialog (window)
A non-text window (panel).

Dribble
A **line macro** p.36.

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

Editor tab stop
A triangle, normally in the lower row of triangles, in the text ruler. These tab stops are used when *tab key generates edit tabs* is selected in the **text ruler** p.66 panel.

Filter
An external, text based UNIX program called by **Edith**, that is invoked on a selected fragment of text in a window, and whose output is substituted for the selected text.

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

Fragment
A continuous piece of text within a selection or a part of a text buffer.

Input queue
When the number of files to open in one go exceeds a number set in the **file options panel** p.78, **Edith** will ask whether some of the files should be put in the *input queue*.

Insert mode
When **Edith** is switched to insert mode using *Replace mode* (**Mark menu** p.55, Insert), what you type in text windows will be inserted into the text. This is the default behaviour.

Macro
Edith has *line*-macros only; they are called *dribbles*.

Marker
Each text window (except browsers) has four *text position markers*. These are set by *Put marker* (**Mark menu** p.55, Alt P), and traversed using *Back* (**Browse menu** p.52, Alt ⇐) and *Forward* (**Browse menu** p.52, Alt ⇒).

Mode
See *text window mode*.

North-East Quadrant
The vertical mirror image of a *south-east quadrant*.

Panel font
The font used in dialog panels. Set under *Text display/fonts...* (**Options menu** p.61, Shift Ctrl D).

Record, rec
Record a **dribble** p.36.

Queue
See *input queue*.

Replace mode
When **Edith** is switched to replace mode using *Replace mode* (**Mark menu** p.55, Insert), what you type in a text window will erase other text. Default is *insert mode*. Replace mode is indicated with a block-style text cursor. In replace mode, the behaviour of *Delete* (**Edit menu** p.56, Alt Shift X) and *Overlay* (**Edit menu** p.56, Alt Shift V) is different.

Selection
The part of the text in a text window which is selected (highlighted). 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. See **selection** p.24.

Shortcut (1)
An abbreviation for a string of characters, expanded by pressing the Escape key. See **shortcut panel** p.70.

Shortcut (2)
A key combination that activates a button or a menu item. Shortcuts 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 Alt plus that letter.

South-East Quadrant
The part of the text to the right and below the cursor, until there is a line which has at most white space beyond the current cursor position. Created by *Select south-east quadrant* (**Mark menu** p.54, Alt Q).

Substitute
That which is substituted for a target by the *search/replace* function. After searching, the cut buffer contains the substitute(s) for the marked target(s).

Tab
An *editor tab* or a *true tab*. Both refer to the space generated by the tab key and the tab stops on the ruler, but an editor tab consists of space characters and a true tab is a single ASCII code (9). See **text rulers panel** p.66.

Target
An occurrence in a text window of the string or pattern you are **searching** p.28 for.

Text buffer
See *buffer*.

Text window mode
Edith's windows are divided into 11 **modes** p.64, 8 of which are user-defined, which have different settings for appearance and editing behaviour.

Trash buffer
The trash buffer contains data discarded from text windows and other text buffers. It can be used to retrieve data which was accidentally lost, and can no longer be recovered by *Undo* (**Edit menu** p.56, Alt U). See *Trash buffer* (**Windows menu** p.63, Shift Ctrl T).

True tab
ASCII character 9, displayed as space stretching out to the next *true tab stop*; appearance set under *Text display/fonts...* (**Options menu** p.61, Shift Ctrl D).

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

D Technical notes

Edith was originally developed for the **Atari ST/TT** series, and **Edith Pro/X11** is today still based on a widget set that is largely based on **GEM**, whose grid-based `.rsc` file system still delivers interface panel layouts that are superior in organization to those compiled with standard X widget sets that use certain 'logical' ways of defining interface layout.

The size of various resources is in principle unbounded—files are allowed to be of any number of lines, lines are allowed to be any length, etc. However, some limitations apply to the values in the files saved by *save options*, and it is among other things therefore that it is not recommended to edit the configuration files manually.

When storing a text in memory, **Edith** stores a pointer to every line in the text. This means that in addition to the size of the text, 4 or 8 bytes are reserved for each line in the text. So a text consisting of line feeds only, on a 64-bit system, occupies nearly ten times its size of memory.

The text buffer system underlying the *undo* system is not really satisfactory.

Edith has been checked for run-time errors and memory leaks using **Purify**. A small number of less significant leaks is still present on some systems, due to problems in third party libraries.

Null characters (**ASCII** code 0) are removed from text files when they are loaded.

Nonstandard escape codes may end up in a file when an **HTML** document is saved. This is circumvented by using *select all* and *save selection* instead.