

**Megaed**

**COLLABORATORS**

	<i>TITLE :</i> Megaed		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 6, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Megaed</b>	<b>1</b>
1.1	Inhaltsverzeichnis der MegaEd-Beschreibung	1
1.2	readmefirst	2
1.3	panic	2
1.4	history	3
1.5	übersicht	3
1.6	fähigkeiten	4
1.7	programmstart	5
1.8	benötigtedateien	5
1.9	konfiguration	6
1.10	konfiguration2	6
1.11	standart-konfiguration	8
1.12	fehler-umsetzer	10
1.13	textfenster	11
1.14	menuhelp	12
1.15	befehlsübersicht	12
1.16	alert	17
1.17	arrange_horizontal	17
1.18	arrange_vertical	17
1.19	backspace	18
1.20	beep	18
1.21	check_brackets	18
1.22	clear	18
1.23	clear_make	19
1.24	close_all_folds	19
1.25	close_server	19
1.26	close_window	20
1.27	cmove	20
1.28	cmove_down	20
1.29	cmove_left	21

---

---

1.30	<code>cmove_right</code>	21
1.31	<code>cmove_up</code>	21
1.32	<code>compile</code>	22
1.33	<code>copy_block</code>	22
1.34	<code>copy_line</code>	22
1.35	<code>count_str</code>	23
1.36	<code>cut_block</code>	23
1.37	<code>debug_make</code>	23
1.38	<code>del</code>	24
1.39	<code>del_to_line_end</code>	24
1.40	<code>del_to_line_start</code>	24
1.41	<code>delay</code>	24
1.42	<code>delete</code>	24
1.43	<code>display_prefs</code>	25
1.44	<code>duplicate_line</code>	25
1.45	<code>edit_a_module</code>	25
1.46	<code>end</code>	26
1.47	<code>end_now</code>	26
1.48	<code>execute</code>	26
1.49	<code>hot_help</code>	27
1.50	<code>iconify</code>	27
1.51	<code>indent_left</code>	27
1.52	<code>indent_right</code>	28
1.53	<code>info</code>	28
1.54	<code>input_asc</code>	28
1.55	<code>insert_block</code>	28
1.56	<code>insert_bufferline</code>	29
1.57	<code>insert_file</code>	29
1.58	<code>insert_string</code>	29
1.59	<code>jump</code>	29
1.60	<code>jump_bottom_view</code>	29
1.61	<code>jump_end</code>	30
1.62	<code>jump_error</code>	30
1.63	<code>jump_last_change</code>	30
1.64	<code>jump_last_position</code>	30
1.65	<code>jump_left_view</code>	31
1.66	<code>jump_line</code>	31
1.67	<code>jump_line_end</code>	31
1.68	<code>jump_line_start</code>	31

---

---

1.69	jump_mark	32
1.70	jump_next_fold	32
1.71	jump_next_tab	32
1.72	jump_next_word	32
1.73	jump_next_word_end	32
1.74	jump_prev_fold	33
1.75	jump_prev_tab	33
1.76	jump_prev_word	33
1.77	jump_prev_word_end	34
1.78	jump_right_view	34
1.79	jump_start	34
1.80	jump_top_view	34
1.81	kill_all_folds	35
1.82	kill_fold	35
1.83	kill_line	35
1.84	kill_mark	35
1.85	last_hot_help	36
1.86	load	36
1.87	load_default_prefs	36
1.88	load_makefile	36
1.89	load_original	37
1.90	load_prefs	37
1.91	make	37
1.92	make_all	38
1.93	make_fold	38
1.94	mark_block	39
1.95	matching_bracket	39
1.96	module_editor	39
1.97	msg	41
1.98	new_window	41
1.99	next_window	41
1.100	open	42
1.101	open_all_folds	42
1.102	open_server	42
1.103	prefs	43
1.104	prev_window	44
1.105	print	44
1.106	print_block	44
1.107	replace	45

---

---

1.108	reset_prefs . . . . .	45
1.109	restore_line . . . . .	46
1.110	return . . . . .	46
1.111	return_nosplit . . . . .	46
1.112	return_revert_insertmode . . . . .	46
1.113	return_split . . . . .	47
1.114	run . . . . .	47
1.115	run_make . . . . .	47
1.116	save . . . . .	48
1.117	save_all . . . . .	48
1.118	save_as . . . . .	48
1.119	save_default_prefs . . . . .	49
1.120	save_icon . . . . .	49
1.121	save_makefile . . . . .	49
1.122	save_makefile_as . . . . .	50
1.123	save_prefs . . . . .	50
1.124	save_program . . . . .	50
1.125	scroll . . . . .	50
1.126	scroll_down_page . . . . .	51
1.127	scroll_up_page . . . . .	51
1.128	search . . . . .	51
1.129	search_next . . . . .	52
1.130	search_prev . . . . .	52
1.131	set_clip . . . . .	53
1.132	set_mark . . . . .	53
1.133	set_right_border . . . . .	53
1.134	sort_lines . . . . .	54
1.135	tab . . . . .	54
1.136	toggle_fold . . . . .	54
1.137	write_mode . . . . .	54
1.138	zoom_all_windows . . . . .	55
1.139	zoom_window . . . . .	55

---

# Chapter 1

## Megaed

### 1.1 Inhaltsverzeichnis der MegaEd-Beschreibung

MegaEd-Inhaltsverzeichnis

Zuerst lesen

Das ist neu (History)

Übersicht und Einleitung

Übersicht der Fähigkeiten

Programmstart

Benötigte Dateien

Konfiguration

Standart-Konfiguration

Fehler-Umsetzer

Textfenster

Text Manager

Blockmarkierungen

MenuHelp

Modul Editor

Befehlsübersicht

Hilf mir !!!

## 1.2 readmefirst

Aktuelle Anmerkungen zu MegaEd 1.0

Die letzte Version 0.9, welche auch Programm des Monats im Amiga Magazin 1/95 wurde, war aus Zeitgründen recht unvollständig und fehlerhaft. Ich hoffe Sie haben sich nicht allzusehr davon abschrecken lassen, denn nun kann ich Ihnen endlich Version 1.0 präsentieren. Zwar kann man bei einem so großen Programm keine Fehlerfreiheit garantieren, aber zumindest traten bei mir weder Enforcer-Hits, Bugs oder Abstürze auf (ich programmiere und schreibe Anleitungen natürlich nur mit MegaEd!).

Die Makeverwaltung kennt bis jetzt folgende Übersetzer:

```
-Maxon C++
-Manx Aztec C 5.2
-Hisoft Devpac
-PCQ
-A68k
-BLink
```

Alle weiteren Compiler können zwar benützt werden, jedoch funktioniert die Fehlererkennung nur, wenn Sie selber einen Fehler-Umsetzer schreiben. Ich würde mich sehr freuen wenn Sie diese Fehlerumsetzer mit Quellcode an mich senden würden. Ich werde Sie dem MegaEd-Paket hinzufügen, damit alle Anwender davon profitieren können. Besonders würden ich mich SAS C Umsetzer freuen. Als keine Gegenleistung bekommen Sie die neuste Version von MegaEd kostenlos zugeschickt.

Für das Update schicken Sie einen, an sich selbst adressierten und mit 3DM frankierten Umschlag, eine Diskette und eine Tafel Schokolade (bin Schokoholiker!) an die folgende Adresse (langweilige Kapitalisten können auch 7,- DM in Briefmarken oder als V-Scheck beifügen):

Markus Junginger, Hainenbachstr. 6, 89522 Heidenheim

## 1.3 panic

```
###          ##### # # #####
# ##        # # ## # #
# #         # # # # #
# #         # # # # #
# #         # # # # #
# #         # # # # #
# ##        # # # ##
###          ##### # #
```

```
#####      ### # # # ##### ##
# #         # # ## # # # # ##
# #         # # # # # # # ##
#####      # # # # # # # ##
#           ##### # # # # # ##
#           # # # # ## # # #
#           # # # # # # ##### ##
```

Keine Panik

## 1.4 history

Die Neuerungen innerhalb der Versionen:

V0.9

Diese Version wurde Programm des Monats im Amiga-Magazin 1/95, war aber aus Zeitmangel teilweise unvollständig und fehlerhaft!

V1.00

Fehler verbessert:

- fataler Fehler bei "Make laden" führte später zum Absturz
- fataler Fehler bei "Speichern" führte später zum Absturz
- Enforcer Hits beseitigt

Neue Funktionen:

- Block drucken
- Formatierungen

V1.01

Fehler verbessert:

- Blockoperationen korrigiert
- Mini-Bug bei Laden

V1.10

Fehler verbessert:

- Bei "Löschen" im Modul-Editor ohne Angewähltes Modul gab es einen Guru

Neue Funktionen:

- Neue Einstellungsmöglichkeiten
- Kleineres Make-Fenster
- Make-Module können versteckt werden
- Zeileumbruch kann abgestellt werden (für DME-Umsteiger)

## 1.5 übersicht

Übersicht

MegaEd ist wie Sie schon sicher erahnt haben, ein Texteditor. Mit MegaEd können Sie also Texte, Programmquellcode und Batchdateien erzeugen und wenn es unbedingt sein muß, auch Briefe schreiben.

Natürlich weiß ich, das es von dieser Sorte Programm schon etliche Vertreter gibt, doch da ich eigensinnig (wahrscheinlich eher dumm) bin und die Programmierung eine Herausforderung darstellte, habe ich noch einen geschrieben.

Bei der Programmierung von MegaEd wurde besonders auf folgende Punkte geachtet: kinderleichte Bedienung, viele mächtige Funktionen, freie Konfigurierbarkeit und Schnelligkeit.

---

## Anforderungen

Sie brauchen einen Amiga mit min. OS2.0 und min. 512KB RAM. Ein paar Funktionen sind erst ab 2.1 bzw. 3.0 vollständig lauffähig. Mit Festplatte, Turbo und Fastram geht natürlich alles viel besser...

## Autor & Copyright

MegaEd wurde 1994 von Markus Junginger programmiert (Copyright © 1994 Markus Junginger). Das Programm ist PD und darf deshalb ohne weiteres weitergegeben werden. Änderungen am Programm und den dazugehörigen Dateien sind untersagt. MegaEd wird laufend weiterentwickelt und deshalb wäre ich Ihnen dankbar wenn Sie mich unterstützen würden. Wenn Sie einen Programmfehler aufgespürt haben oder weitere Anregungen haben, schreiben Sie bitte an folgende Adresse:

Markus Junginger  
Hainenbachstr. 6  
89522 Heidenheim

Wenn ich Ihnen antworten soll, legen Sie bitte einen adressierten und frankierten Rückumschlag bei, sonst werde ich schweigen wie ein Grab.

## Updates

Da das Programm ständig weiterentwickelt wird, können Sie von mir Updates erhalten: für das Update schicken Sie ein an sich selbst adressierten und mit 3 DM frankierten Umschlag, eine Diskette und eine Tafel Schokolade (bin Schokolohiker!) an die obige Adresse (langweilige Kapitalisten können auch 7,- DM in Briefmarken oder als V-Scheck beifügen).

## 1.6 Fähigkeiten

Hier die bedeutendsten Fähigkeiten aufgelistet:

- beliebig viele Texte und Fenster können geöffnet werden
  - Text Manager
  - Eine umfangreiche, intuitiv bedienter Make-Umgebung speziell für Programmierer. Der Compiler wird perfekt in MegaEd eingebettet!
  - Menus, Tastenbelegungen und QuickGadgets sind konfigurierbar
  - Jeder Text ist ein eigener Task und arbeitet somit unabhängig
  - Die Druckroutine ist ebenfalls ein selbständiger Task
  - Unterstützung des Clipboard-Devices
  - 256 verschiedene Textblöcke definierbar
  - schnelles Scrolling und schnelle Bearbeitung
  - Infowindow gibt Zusatzinformationen
  - QuickGadgets erlauben schnelle Ausführung von oft benutzten Funktionen
  - Onlinehilfe und MenuHelp-Fähigkeit
  - HotHelp Unterstützung
  - Undo/Redo
  - Klammerprüfung
  - Viele Einstellungsmöglichkeiten
  - Unterstützung von Public-Screens
  - AppWindows und AppIcons
-

-Automatische Backups  
-und vieles mehr...

## 1.7 programmstart

MegaEd-Programmstart

MegaEd wird entweder aus der Workbench oder Shell durch Angabe des Hauptprogramms gestartet. Dabei haben Sie die Möglichkeit noch Argumente anzugeben, nämlich beliebig viele Texte und ein Makefile. Makefiles werden automatisch erkannt und müssen somit nicht besonders gekennzeichnet werden. Auf der Workbench klicken Sie zuerst MegaEd, dann in Verbindung mit der Shift-Taste alle weiteren Dateien, die eingelesen werden sollen. In der Shell geben Sie einfach nach MegaEd alle weiteren Dateien, die eingelesen werden sollen.

Beim Programmstart werden hauptsächlich folgende Aktionen ausgeführt:

- Prüfen ob OS2.0 oder höher vorhanden
- "Warte-Fenster" öffnen
- Einige Libs öffnen
- Default-Einstellungen laden
- Icons einlesen
- MsgPort's erstellen
- Clipboard öffnen
- Fonts öffnen
- Screen öffnen
- Argumente auswerten

## 1.8 benötigtedateien

Benötigte Dateien

Zwar würde es reichen MegaEd nur mit dem Hauptprogramm zu starten, jedoch würden hiermit einige schöne Funktionen lahmgelegt. Also sollten Sie darauf achten, daß die zu MegaEd gehörigen Dateien (im richtigen Verzeichnis) vorhanden sind.

MegaEd

Das Hauptprogramm

MegaEd.guide

Die Hilfedatei im AmigaGuide® Format

Configs/MegaEd.config

Die Belegungsdatei für Menü, Tastatur und Gadgets

Configs/TextIcon

Das Icon für Texte

Configs/MakeIcon

Das Icon für Makefiles

Configs/MegaEdIcon

Das Icon für das AppIcon

ENV:MegaEd/MegaEd.prefs

Die Einstellungsdatei

ENVARC:MegaEd/MegaEd.prefs

Die dauerhaft gespeicherte Einstellungsdatei

LIBS:amigaguide.library  
Für spezielle Hilfefunktionen (z. B. MenuHelp)  
LIBS:hothelp.library  
Für das Tool "HotHelp" von Maxon

## 1.9 konfiguration

### Konfiguration

Das Erscheinungsbild von MegaEd können Sie fast nach Belieben ändern. Das heißt, Sie haben die Möglichkeit die Menu-, Gadget- und Tastaturbelegung ganz nach Ihren Wünschen zu gestalten. Sie können aber auch schon vordefinierte Konfigurationen benutzen.

Wenn Sie die Belegungen selber entwerfen wollen, müssen Sie sich an ganz bestimmte Regeln halten, wie bei Programmiersprachen. Aber keine Angst, das Ganze ist natürlich viel einfacher.

Wenn Sie MegaEd starten, so wird versucht die Konfigurationsdatei "configs/MegaEd.config" zu lesen. Wird diese nicht gefunden oder enthält sie Fehler, wird eine interne Konfiguration benutzt.

Im Verzeichnis "Configs" befinden sich einige Konfigurationsdateien und außerdem noch das Standarticon. Alle Dateien sollten unverändert bleiben, solange Sie nicht ganz genau wissen was sie tun. So könnten Sie eine Datei laden und als "MegaEd.config" speichern um diese Datei zur Standardkonfiguration zu machen.

Siehe auch:

### Konfigurations-Syntax

## 1.10 konfiguration2

### Konfigurations-Syntax

Die Config-Dateien sind alle nach einem bestimmten Schema aufgebaut. Die Groß-/Kleinschreibung muß nicht wie folgt verwendet werden, sorgt aber für mehr Übersicht.

So wird mit dem #-Zeichen immer bestimmt, was mit den folgenden Zeilen belegt werden soll. Am Dateiende muß immer #END stehen um das Ende anzuzeigen.

Sie haben folgende Möglichkeiten:

#MENU <Anzahl>

Damit werden die nächsten Zeilen als Menu-Belegung aufgefaßt. Hinter "#MENU" müssen Sie eine Zahl angeben, die angibt wieviele Menu-Punkte maximal existieren. Ist Sie zu klein, wird ein Fehler ausgegeben, und Sie müssen die Zahl heraufsetzen. Natürlich müssen Sie die Anzahl nicht genau abzählen, Sie können z.B. ruhig 50 mehr angeben.

#QUICK\_GADGETS

MegaEd stellt maximal 40 Gadgets zur Verfügung, die in jedem Fenster erscheinen. mit diesem Kommando können Sie diese nach Wunsch belegen.

#KEYS\_NONE

---

Damit können Sie die Tasten belegen. Die Bezeichnung nach dem Unterstrich gibt an, welche Tasten (Shift, Alt, Control) zusätzlich gedrückt werden müssen. Diese Tasten können auch kombiniert werden:

```
#KEYS_NONE           Ohne Zusatz
#KEYS_SHIFT          Mit Shift
#KEYS_CONTROL        Mit Ctrl
#KEYS_ALT             Mit Alt
#KEYS_SHIFT_CONTROL  Mit Shift und Ctrl
#KEYS_SHIFT_SHIFT_CONTROL_ALT Mit Shift, Ctrl und Alt
#KEYS_SHIFT_CONTROL_ALT Mit Ctrl und Alt
```

#### Befehle

Die Aktionen (Befehlsfolge) die bei Anwahl eines/einer Gadgets/Menupunktes/Taste ausgeführt werden, werden immer gleich beschrieben. Die Befehlsfolgen können beliebig viele Befehle enthalten.

Die Befehle stehen immer zwischen geschweiften Klammer {}, auch wenn nur ein Befehl aufgeführt ist.

Nach einigen Befehlen können oder müssen noch Parameter angegeben werden.

Ist ein Befehl vollständig muß ein Semikolon(;) stehen.

Beispiele:

```
{Save_As;}
{Save; Clear; Alert "Text gespeichert und gelöscht";}
{Save "S:Startup-Sequence";}
```

#### Menubeschreibung (#MENU)

Es gibt drei Arten von Menupunkten: Title(Titelzeilen), Item(Menupunkt) und Sub\_Item(Unterverzeichnis eines Menupunkt).

Die Menubeschreibung wird Menu für Menu erstellt. Nach den Schlüsselwörtern (Title, Item und Sub\_Item) muß ein Text folgen, welcher zur Darstellung verwendet wird. Danach die Befehlsfolge. Wird dieser Menupunkt oft benötigt, kann ein Shortcut in Anführungszeichen nach der Befehlsfolge angegeben werden.

Zur besseren Gliederung können die Kommandos "Item\_Bar" und "Sub\_Item\_Bar" benutzt werden, welche einen Trennstrich ziehen.

Beispiel:

```
#Menu
Title "Projekt"
  Item "Info..." {Info;} "I"
  Item "Speichern"
    Sub_Item "Text 1" {Save "Text1";}
    Sub_Item_Bar
    Sub_Item "Text 2" {Save "Text2";}
Title "Menu 2"
  Item "Ende" {End;} "Q"
```

#### Tastaturbeschreibung

Zuerst wird die zu belegende Taste(ohne Anführungszeichen) angegeben, dann dazugehörige Befehlsfolge. Existiert zu einer Taste keine solche Belegung, so wird sie(falls möglich) in das Textfenster eingefügt.

Für einige Tasten müssen Sie folgende Schlüsselwörter schreiben:

```
Enter/Return
F1 bis F10
Tab
Esc
Space
Del/Delete
```

Help

Up, Down, Left, Right (Cursortasten: Hoch, Runter, Links, Rechts)

N0 bis N9 (Zahlen am Nummerblock rechts)

N. (Punkt am Nummerblock rechts)

Beispiele:

```
#KEYS_NONE
  Return {Return;}
#KEYS_CTRL
  Y {Kill_Line;}
```

Gadgetbeschreibungen

Zuerst wird ein Text mit max. 7 Buchstaben in Anführungszeichen angegeben, welcher im Gadget erscheint. Danach kommt die Befehlsfolge.

Beispiele:

```
#QUICK_GADGETS
  "Laden" {Load;}
  "Ende" {End;}
```

## 1.11 standart-konfiguration

Standart-Tastatur

Die Konfiguration kann zwar den eigenen Wünschen angepaßt und komplett verändert werden, aber ich will an dieser Stelle trotzdem ein paar Worte über die Standart-Tastaturbelegung verlieren.

Tasten die alleine betätigt werden:

Return	Fügt neue Zeile ein. Gegebenfalls wird der Text ab dem Cursor in die neue Zeile übernommen.
Help	Für das Wort unter dem Cursor wird ein HotHelp-Hilfefenster geöffnet
F1	Text laden
F2	Text speichern
F3	Neues Textfenster
F4	Suchen
F5	Block kopieren
F6	Ein Modul zum editieren auswählen
F7	-
F8	Modul-Editor
F9	Make-Programm erstellen
F10	Make-Programm ausführen

Tasten die mit Shift betätigt werden:

(Pfeil)Links	An den Anfang der Linie springen
Rechts	An das Ende der Linie springen
Hoch	Eine Seite nach oben scrollen
Runter	Eine Seite nach unten scrollen

Backspace	Buchstaben links vom Cursor bis zum Zeilenanfang löschen
Del	Buchstaben ab dem Cursor bis zum Zeilenende löschen

---

Help	Für das Wort unter dem Cursor wird ein HotHelp-Hilfefenster geöffnet
Return	Springt zum Anfang der nächsten Zeile. Der Text bleibt unverändert.
F1	Text öffnen
F2	Text speichern als...
F3	Textfenster schließen
F4	Ersetzen
F5	Block einfügen
F6	Modul zum editieren auswählen
F7	-
F8	Modul-Editor
F9	Make-Programm Module compilieren, aber nicht linken
F10	Make-Programm debuggen

Tasten die mit Shift betätigt werden:

Help	Letztes HotHelp-Fenster zeigen
------	--------------------------------

Tasten des Ziffernblocks:

8	Cursor rauf
2	Cursor runter
4	Cursor links
6	Cursor rechts
9/PgUp	Eine Seite nach oben scrollen
3/PgDn	Eine Seite nach unten scrollen
7/Home	Zum Textanfang springen
1/End	Zum Textende springen
0/Ins	Zeile einfügen ohne das die aktuelle Zeile geteilt wird
./Del	Zeile löschen

1 bis 0	Zur Markierung 1 bis 10 springen
---------	----------------------------------

Links	Zur ersten sichtbaren Spalte springen
Rechts	Zur letzten sichtbaren Spalte springen
Rauf	Zur ersten sichtbaren Zeile springen
Runter	Zur letzten sichtbaren Zeile springen

R	Alle Veränderung an der Zeile werden rückgängig gemacht
Q	Buchstaben ab dem Cursor bis zum Zeilenende löschen
Y	Die Zeile wird gelöscht, kann aber wieder eingefügt werden
U	Eine gelöschte Zeile wird wieder eingefügt
I	Eine gelöschte Zeile wird wieder eingefügt
D	Die aktuelle Zeile wird verdoppelt
C	Die Zeile wird zwischengespeichert und kann mit Ctr+U oder Ctrl+I eingefügt werden

Tasten die mit Shift und Control betätigt werden:

1 bis 0	Die Markierung 1 bis 10 wird gesetzt
---------	--------------------------------------

Links	Scrollt eine Spalte nach links
Rechts	Scrollt eine Spalte nach rechts
Rauf	Scrollt eine Zeile nach links
Runter	Scrollt eine Zeile nach rechts

---

Tasten die mit Alt betätigt werden:

Links	Springe zum nächsten Wort
Rechts	Springe zum vorherigen Wort
Rauf	Springe zum Textanfang
Runter	Springe zum Textende

Tab	Springe zum nächsten Tabulator
-----	--------------------------------

Return	Wie normales Return, allerdings wird die neue Zeile entgegengesetzt der Einstellung eingerückt oder nicht
--------	---

Tasten die mit Alt und Shift betätigt werden:

Return	Eine neue Zeile wird eingefügt ohne die bestehende Zeile aufzutrennen
--------	---

Esc	MegaEd wird sofort beendet ohne Speichern oder Sicherheitsabfragen
-----	--

Links	Springe zum nächsten Wortende
Rechts	Springe zum vorherigen Wortende
Rauf	Springe zur vorherigen Falte
Runter	Springe zur nächsten Falte

Tab	Springe zum vorherigen Tabulator
-----	----------------------------------

Tasten die mit Shift, Control und Alt betätigt werden:

1 bis 0	Die Markierung 1 bis 10 wird gelöscht
---------	---------------------------------------

## 1.12 fehler-umsetzer

### Fehler-Umsetzer

Wenn Sie programmieren oder z. B. mit TeX arbeitet müssen Sie Ihre Quellcodes einem Compiler übergeben. Um dies komfortabel zu gestalten wurde in MegaEd eine Make-Verwaltung eingebaut, welche verschiedene Quelltexte und Übersetzer erlaubt (siehe

Module\_Editor  
)

Schön und gut, aber Sie haben zusätzlich mit den "Fehler-Umsetzern" die Möglichkeit, die vom Compiler abhängige (Fehler-)Ausgabe in ein Format umzuwandeln, welches von MegaEd "verstanden" wird. Somit kann sich MegaEd Fehler und Fehlerposition im Quelltext merken, damit Sie Ihre Fehler komfortabel auf Knopfdruck anspringen und verbessern können.

Diese Umsetzer sind prinzipiell nur Miniprogramme, die MegaEd nach jeder Übersetzungsaktion aufruft.

Momentan sind Umsetzer für folgende Programme vorhanden:

Maxon C, Atzec C, Devpac2

Sollte Ihr Programm nicht dabei sein, können Sie diesen Zustand selbst beheben und einen eigenen Umsetzer schreiben. Dazu sind keine großen Programmierkenntnisse erforderlich, zumal Sie auch die Quelltexte zu den bereits existierenden Umsetzern als Ausgangsbasis verwenden können.

Es muß beachtet werden:

- Die Ausgabe des Programms (Kompiler, Linker, TeX, ...) wird in die Datei "T:MegaEdMake-ErrFile" umgeleitet. Diese Datei muß eingelesen werden.
- Der Umsetzter muß daraus eine neue Datei "T:MegaEdMake-Errors" erstellen, welche ein MegaEd verständliches Format hat. Wurden keine Fehler gefunden, sollte die Datei gelöscht werden.

Das Format bei Linker-Umsetzung:

Wurde kein Fehler gefunden, so kann (muß aber nicht) die Codegröße des erstellten Programms übergeben werden. Dazu stellen Sie ein "S" (Size) vor die Zahl. Bsp.: S10531

Wurden dagegen Fehler gefunden, so schreiben Sie einfach die Fehlertexte, je durch ein Return (Asc-Code:10) abgeschlossen, in die Datei.

Das Format bei Kompiler-Umsetzung:

Wurde kein Fehler gefunden, so löschen sie "T:MegaEdMake-Errors".

Wurden dagegen Fehler gefunden, erstellen Sie eine aus zwei Zeilen bestehende Fehlerbeschreibung: Wenn es sich lediglich um eine Warnung handelt, schreiben Sie ein "W", bei einem echten Fehler ein "E" in die Datei. Danach folgt die Zeile und durch ein "/" getrennt die Spalte. Durch ein Return (Asc:10) getrennt, folgt nun der eigentliche Fehlertext, welcher ebenfalls mit einem Return abgeschlossen sein muß.

Bsp.:

W35/2

No prototype for this function defined

E42/10

Illegal Argument

## 1.13 textfenster

### Das Textfenster

Jede geöffnete Datei wird in einem sogenannten Textfenster editiert. Da MegaEd beliebig viele Dateien editieren kann, gibt es auch beliebig viele Textfenster, nur durch den Speicherplatz beschränkt.

Im Wesentlichen sind die Textfenster in vier Teile unterteilt:

System-Rahmen: System-Gadgets zum Schließen, Zoomen, Vergrößern und Gadgets um innerhalb des Textes zu scrollen (Balken und Pfeilgadgets).

Info-Anzeige: Links oben befinden sich 3 Zeilen, die einige wichtige Informationen anzeigen. "Zeile x/x" zeigt die aktuelle Zeile und die Anzahl aller Zeile, "Spalte:x" die aktuelle Spalte und "Änderungen:x" die Anzahl aller Veränderungen, die seit dem letzten Speichern gemacht wurden. In der gleichen Spalte unter dem Wort "Spalte:" befindet sich die Textstatus-Anzeige. Sie besteht aus einzelnen invertierten Buchstaben:

F: Der Cursor steht auf einer geschlossenen Falte

f: Der Cursor steht auf einer geöffneten Falte

B: Die Blockmarkierung ist aktiv

Ü: Der Überschreiben-Modus ist aktiv

Die unterste Zeile (Mitteilungszeile) gibt bei vielen Funktionen Auskunft, damit Sie immer wissen was los ist. Auch einige Fehlermeldungen erscheinen hier.

Quick-Gadgets: Rechts neben der Info-Anzeige befinden sich einige Gadgets, die durch die

Konfiguration

belegt sind.

Editierbereich: Dieser Hauptteil erlaubt die eigentliche Texteingabe.

Alle Eingaben werden hier an der aktuellen Cursor-Position eingefügt. Der Cursor kann auch mit der Maus bewegt werden, indem Sie auch die neue Position klicken. Bei einem Doppelklick wird die Blockmarkierung aktiviert.

Alle Textfenster sind sogenannte AppWindows, d. h. Sie können einen beliebigen Text laden, indem Sie einfach sein Icon von der Workbench in das Textfenster ziehen und loslassen. Wenn Sie mehrere Icons auf einmal in das Fenster ziehen (Shift-Taste), so werden entsprechend viele neue Fenster geöffnet ohne das den anderen Text zu verändern.

## 1.14 menuhelp

MenuHelp

Mit der "Amigaguide.library" ist es auf einfache Weise möglich eine Beschreibung über einen beliebigen Menüpunkt zu bekommen. Drücken Sie dazu wie gewohnt die rechte Maustaste und zeigen Sie auf den gewünschten Menüpunkt. Anstatt nun die Taste loszulassen, drücken Sie die Help-Taste, worauf die "MegaEd.guide" Datei eingeladen und die dem Menüpunkt zugehörige Seite aufgeschlagen wird.

Menüpunkte welche durch die Konfigurationsdatei mit mehr als einem Befehl belegt sind, können (noch) nicht komplett erläutert werden. Hier wird nur der erste Befehl der Befehlsfolge erläutert.

Der zu erläuternde Befehl wird auch im Infofenster ausgegeben.

Das Textfenster ist erst wieder benützbar, wenn das Hilfefenster geschlossen wird.

## 1.15 befehlsübersicht

MegaEd Befehle

Bei der

Konfiguration

können folgende Befehle verwendet werden.

Bei einigen Befehlen können oder müssen Argumente abgegeben werden, je nach Angabe der Argumente in der Erklärung: befindet sich das Argument in eckigen Klammern "[ ]", so kann es angegeben werden, muß aber nicht. Ohne Klammern muß es angegeben werden.

alert

arrange\_horizontal

arrange\_vertical

backspace

beep

check\_brackets

---

```
clear
clear_make
close_all_folds
close_server
close_window
cmove
cmove_down
cmove_left
cmove_right
cmove_up
compile
copy_block
copy_line
count_str
cut_block
debug_make
del
del_to_line_end
del_to_line_start
delay
delete
display_prefs
duplicate_line
edit_a_module
end
end_now
execute
  format_block
format_center
format_left
format_right
```

---

---

hot\_help  
iconify  
indent\_left  
indent\_right  
info  
input\_asc  
insert\_block  
insert\_bufferline  
insert\_file  
insert\_string  
jump  
jump\_bottom\_view  
jump\_end  
jump\_error  
jump\_last\_change  
jump\_last\_position  
jump\_left\_view  
jump\_line  
jump\_line\_end  
jump\_line\_start  
jump\_mark  
jump\_next\_fold  
jump\_next\_tab  
jump\_next\_word  
jump\_next\_word\_end  
jump\_prev\_fold  
jump\_prev\_tab  
jump\_prev\_word

---

---

jump\_prev\_word\_end  
jump\_right\_view  
jump\_start  
jump\_top\_view  
kill\_all\_folds  
kill\_fold  
kill\_line  
kill\_mark  
last\_hot\_help  
load  
load\_default\_prefs  
load\_makefile  
load\_original  
load\_prefs  
make  
make\_all  
make\_fold  
mark\_block  
matching\_bracket  
module\_editor  
msg  
new\_window  
next\_window  
open  
open\_all\_folds  
open\_server  
  prefs  
prev\_window  
print

---

---

```
print_block
replace
reset_prefs
restore_line
return
return_nosplit
return_revert_insertmode
return_split
run
run_make
save
save_all
save_as
save_default_prefs
save_icon
save_makefile
save_makefile_as
save_prefs
save_program
scroll
  scroll_page_down
scroll_page_up
search
search_next
search_prev
  select_modules
set_clip
set_mark
set_right_border
sort_lines
```

---

```
tab
toggle_fold
write_mode
zoom_all_windows
zoom_window
```

## 1.16 alert

```
Alert Text
```

Öffnet Requester

Es wird ein Requester, der den angegebenen Text darstellt, geöffnet. Der Anwender muß diesen bestätigen, indem er das Gadget anwählt. Dieser Befehl sollte nur benutzt werden, wenn der Anwender auf etwas wirklich wichtiges aufmerksam gemacht werden muß.

Beispiele:

```
Alert "Hallo!"
```

Siehe auch:

```
Msg
```

## 1.17 arrange\_horizontal

```
Arrange_Horizontal
```

Ordnet Fenster horizontal

Die Position und Größe aller geöffneten Fenster werden so verändert, daß sie den Screen zu gleichen Teilen einnehmen. Dabei bleibt die Fensterbreite größtmöglich, die Höhe teilt sich aber auf.

Siehe auch:

```
Arrange_Vertical
```

## 1.18 arrange\_vertical

```
Arrange_Vertical
```

Ordnet Fenster vertikal

Die Position und Größe aller geöffneten Fenster werden so verändert, daß

---

sie den Screen zu gleichen Teilen einnehmen. Dabei bleibt die Fensterhöhe größtmöglich, die Breite teilt sich aber auf.

Siehe auch:

Arrange\_Horizontal

## 1.19 backspace

Backspace

Löscht Zeichen

Das Zeichen links vom Cursor wird gelöscht. Ist der Cursor am Zeilenanfang, so wird diese Zeile mit der vorhergehenden verbunden.

Siehe auch:

Delete

## 1.20 beep

Beep

Bewirkt Aufblitzen des Screens

## 1.21 check\_brackets

Check\_Brackets

Klammern überprüfen

Es wird überprüft, ob alle Klammern ("(", "[", "{") in der richtigen Reihenfolge geschlossen werden. Diese Funktion ist speziell für C Programme nützlich, da Klammerfehler hiermit viel schneller gefunden werden. Allerdings gibt es hier Einschränkungen, da auch Klammern in Strings und in Kommentaren als Strukturklammern gesehen werden.

Siehe auch:

Matching\_Bracket

## 1.22 clear

Clear

Text wird gelöscht

---

Der Text im aktuellen Textfenster wird gelöscht, damit Sie eine neue Datei erstellen können.

Siehe auch:

Close\_Window

## 1.23 clear\_make

Clear\_Make

Make wird gelöscht

Es wird alles zurückgesetzt (gelöscht), was durch den Moduleditor eingegeben wurde.

Siehe auch:

Module\_Editor

## 1.24 close\_all\_folds

Close\_All\_Folds

Schließt alle Falten

Siehe auch:

Open\_All\_Folds

'  
Toggle\_Fold

## 1.25 close\_server

Close\_Server

Schließt Textmanager

Das Fenster des Textmanagers wird geschlossen. Dasselbe können Sie durch das Closegadget oder durch einen Druck auf Esc erreichen. Ist sonst kein Textfenster mehr geöffnet, so erscheint ein Requester, mit dem Sie entweder MegaEd beenden können, oder ein neues Fenster öffnen können.

Siehe auch:

Open\_Server

---

## 1.26 close\_window

Close\_Window

Schließt Textfenster

Das Textfenster wird geschlossen. War dies das letzte, so erscheint ein Requester, mit dem Sie entweder MegaEd beenden können, oder ein neues Textfenster öffnen können.

Siehe auch:

New\_Window  
,  
Clear

## 1.27 cmove

CMove Spalten,Zeilen

Bewegt den Cursor relativ

Der Cursor wird um die angegebene Anzahl an Spalten und Zeilen bewegt. Eine positive Anzahl bewegt den Cursor nach recht bzw. unten, eine negative nach links bzw. oben.

Beispiele:

CMove -1,-1;

CMove 1,0;

Siehe auch:

CMove\_Left  
,  
CMove\_Right  
,  
CMove\_Up  
,  
CMove\_Down

## 1.28 cmove\_down

CMove\_Down

Bewegt den Cursor runter

Der Cursor wird eine Zeile nach unten bewegt.

Siehe auch:

CMove  
,

```
CMove_Left  
,  
CMove_Right  
,  
CMove_Up
```

## 1.29 cmove\_left

```
CMove_Left
```

Bewegt den Cursor links

Der Cursor wird eine Spalte nach links bewegt.

Siehe auch:

```
CMove  
,  
CMove_Down  
,  
CMove_Right  
,  
CMove_Up
```

## 1.30 cmove\_right

```
CMove_Right
```

Bewegt den Cursor rechts

Der Cursor wird eine Spalte nach rechts bewegt.

Siehe auch:

```
CMove  
,  
CMove_Left  
,  
CMove_Down  
,  
CMove_Up
```

## 1.31 cmove\_up

```
CMove_Up
```

Bewegt den Cursor hinauf

Der Cursor wird eine Zeile nach oben bewegt.

---

Siehe auch:

```
CMove
,
CMove_Left
,
CMove_Right
,
CMove_Down
```

## 1.32 compile

Compile

Übersetzten aller Module

Gemäß der Daten, die mit dem  
Modul-Editor

angegeben wurden, werden nun alle

Module ggf. übersetzt aber nicht gelinkt.

Ist nützlich, wenn man ein Modul fertig editiert hat und sich nun einem  
anderen zuwendet, da im Hintergrund das fertige Modul Übersetzt wird.

Weiter Infos siehe

```
Make
.
```

Siehe auch:

```
Modul-Editor
,
Make
```

## 1.33 copy\_block

Copy\_Block

Kopiert den markierten Bereich ins Clipboard

Siehe auch:

```
Mark_Block
,
Insert_Block
,
Cut_Block
```

## 1.34 copy\_line

Copy\_Line

Kopiert die aktuelle Linie in einen Zwischenspeicher

Siehe auch:

Insert\_Bufferline

### 1.35 count\_str

Count\_Str

Das Vorkommen einer Zeichenkette wird gezählt

Es öffnet sich der Suchrequester (siehe

Search

). Der angegebene Suchtext wird

im Text gesucht und die vorkommende Anzahl in einem Requester ausgegeben.

Siehe auch:

Search

### 1.36 cut\_block

Cut\_Block

Verlagert den markierten Bereich ins Clipboard

Der markierte Bereich wird im Text gelöscht, aber zuvor ins Clipboard kopiert, damit er an anderer Stelle eingefügt werden kann.

Siehe auch:

Mark\_Block

,

Insert\_Block

,

Copy\_Block

### 1.37 debug\_make

Debug\_Make

Debuggen des Make-Programmes

Es wird der Debugger gemäß dem Modul-Editor aufgerufen, um das Programm zu debuggen.

---

Siehe auch:

Modul-Editor

### 1.38 del

siehe Delete

### 1.39 del\_to\_line\_end

Del\_To\_Line\_End

Die Zeichen vom Cursor bis zum Linienende werden gelöscht

Siehe auch:

Del\_To\_Line\_Start

### 1.40 del\_to\_line\_start

Del\_To\_Line\_Start

Die Zeichen links vom Cursor werden gelöscht

Siehe auch:

Del\_To\_Line\_End

### 1.41 delay

Delay Time

Wartet

Hiermit veranlassen Sie den Texttask  $\text{Time} * 1/50$  Sekunden zu warten, aber höchstens eine Sekunde.

### 1.42 delete

Delete

Löscht Zeichen

Das Zeichen auf dem sich der Cursor befindet wird gelöscht. Ist der Cursor am Zeilenende, so wird diese Zeile mit der nachfolgenden verbunden.

---

Siehe auch:

Backspace

## 1.43 display\_prefs

Display\_Prefs

Öffnet Anzeige-Einstellungen Requester

Es öffnet sich ein Fenster um folgende Einstellungen zu treffen:

Screen-Modus: Hiermit bestimmen Sie auf welchem Screen MegaEd seine Fenster öffnet. Geben Sie hier "Public" an, so müssen Sie noch den exakten Namen des gewünschten Public-Screens in dem Stringgadget rechts eingeben.

Bildschirmmodus: Wenn Sie einen eigenen Screen öffnen können Sie hiermit den gewünschten Modus, die Größe und die Tiefe bestimmen (ab OS2.1). Bei OS2.0 oder kleiner, öffnet sich lediglich ein eigener Requester für Modus und Größe.

oder Workbench klonen: Modus, Größe und Tiefe des Screen werden von der Workbench übernommen.

Screen-Font: Hiermit wählen Sie einen Font für den Screen und das Menu aus.

oder Standard-Font: Es wird der Preferences-Font genommen.

Text-Font: Hiermit wählen Sie einen proportionalen Font, welcher benutzt wird, den Text zu editieren.

oder Standard-Font: Es wird der Preferences-Font genommen.

Public-Screen: Der geöffnete (eigene) Screen wird für öffentlich erklärt.

Shanghai: Hiermit können Sie einige Programme dazu bewegen, ihre Fenster anstatt auf der Workbench auf dem MegaEd-Screen zu öffnen.

Siehe auch:

Prefs

## 1.44 duplicate\_line

Duplicate\_Line

Die aktuelle Linie wird verdoppelt

## 1.45 edit\_a\_module

Edit\_A\_Module

Modul um editieren wählen

Gemäß der Daten, die mit dem

Modul-Editor

angegeben wurden, wird nun

ein Fenster geöffnet, in dem alle Module aufgelistet werden. Wenn Sie eins davon anwählen, wird es geladen. Dabei wird ein neues Textfenster benützt, wenn die Option Neues Fenster aktiv ist.

Siehe auch:

Modul-Editor

## 1.46 end

End

MegaEd wird beendet

Nach einer Sicherheitsabfrage wird MegaEd beendet.

Siehe auch:

End\_Now

## 1.47 end\_now

End\_Now

MegaEd wird sofort beendet

MegaEd wird sofort, ohne jegliche Sicherheitsabfrage beendet, so daß veränderte Texte verloren gehen, wenn Sie noch nicht gespeichert wurden. Also Vorsicht!

Siehe auch:

End

## 1.48 execute

Execute [Dateiname]

Startet ein Programm

Hiermit können Sie ein beliebiges ausführbares Programm starten. Es muß gewartet werden bis das Programm beendet wird. Wird [Dateiname] nicht angegeben, so erscheint ein Filerequester zur Auswahl einer Datei.

Beispiele:

```
Execute "SYS:Tools/Calculator";  
Execute;
```

Siehe auch:

---

Run

## 1.49 hot\_help

Hot\_Help [Projekt]

Öffnet HotHelp

Das Wort unter dem Cursor wird HotHelp (geniales Hilfesystem von Maxon!) übergeben, welches dafür dann die entsprechende Beschreibung anzeigt. Wird Projekt angegeben, so wird HotHelp mitgeteilt, das es nur in einem Projekt suchen soll.

Beispiele:

```
Hot_Help "Shell";
```

```
Hot_Help;
```

Siehe auch:

Last\_Hot\_Help

## 1.50 iconify

Iconify

Alle Fenster werden vorübergehend geschlossen

Alle Textfenster und der Text-Manager werden geschlossen. Statt dessen erscheint ein App-Icon auf der Workbench. Wenn Sie dieses doppelt anklicken, so werden wieder alle Fenster geöffnet.

Siehe auch:

Prefs

## 1.51 indent\_left

Indent\_Left

Zeilen werden verschoben

Um das Strukturieren speziell beim Programmieren zu erleichtern, können Sie hiermit alle markierte Zeilen um eine Einheit (Space oder Tab, siehe

```
Prefs
```

```
)
```

nach links verschieben.

Siehe auch:

---

Indent\_Right

## 1.52 indent\_right

Indent\_Right

Zeilen werden verschoben

Um das Strukturieren speziell beim Programmieren zu erleichtern, können Sie hiermit alle markierte Zeilen um eine Einheit (Space oder Tab, siehe

Prefs

)

nach rechts verschieben.

Siehe auch:

Indent\_Left

## 1.53 info

Info

Informationen über Texte und MegaEd

Es wird ein Requester geöffnet, welcher über Programmversion, Speicherplatz und globale Änderungen Auskunft gibt.

## 1.54 input\_asc

Input\_Asc

Ein beliebiger Asc-Wert wird in den Text eingefügt

## 1.55 insert\_block

Insert\_Block

Kopiert den Clipboardinhalt in den Text

Siehe auch:

Mark\_Block

,

Copy\_Block

,

Cut\_Block

## 1.56 insert\_bufferline

Insert\_Bufferline

Fügt die Linie aus dem Zwischenspeicher ein

Siehe auch:

Copy\_Line

## 1.57 insert\_file

Insert\_File [Dateiname]

Fügt Datei ein

Eine Datei wird in das Textfenster eingefügt, also zusätzlich zur bereits vorhandenen Datei. Mit Hilfe eines Requesters können Sie zusätzlich wählen, an welcher Position die neue Datei eingefügt wird.

Wird Dateiname nicht angegeben, so erscheint ein Filerequester zur Auswahl einer Datei.

## 1.58 insert\_string

Insert\_String String

Die Zeichenkette (String) wird in den Text eingefügt

## 1.59 jump

Jump Spalte,Zeile

Der Cursor springt an die angegebene Position

Beispiel:

Jump 1,1;

## 1.60 jump\_bottom\_view

Jump\_Bottom\_View

Der Cursor springt zur letzten, im Fenster sichtbaren Zeile

Siehe auch:

Jump\_Top\_View

,

```
Jump_Left_View  
,  
Jump_Right_View
```

## 1.61 jump\_end

```
Jump_End
```

Der Cursor springt zur letzten Zeile

Siehe auch:

```
Jump_Start
```

## 1.62 jump\_error

```
Jump_Error Fehlernummer
```

Springt Fehler an

Falls beim Übersetzen der Module Fehler auftraten, können sie hiermit direkt angesprungen werden, der Fehlertext wird auch ausgegeben.  
Die Fehlernummer reicht von 1 bis 10.

## 1.63 jump\_last\_change

```
Jump_Last_Change
```

Der Cursor springt zur Position der letzten Eingabe/Veränderung

Siehe auch:

```
Jump_Last_Position
```

## 1.64 jump\_last\_position

```
Jump_Last_Position
```

Der Cursor springt zur vorherigen Position

Siehe auch:

```
Jump_Last_Change
```

## 1.65 jump\_left\_view

Jump\_Left\_View

Der Cursor springt zur ersten, im Fenster sichtbaren Spalte

Siehe auch:

Jump\_Right\_View  
,  
Jump\_Top\_View  
,  
Jump\_Bottom\_View

## 1.66 jump\_line

Jump\_Line

Zeile anspringen

Es öffnet sich ein Requester mit dessen Hilfe Sie eine Zeilennummer eingeben können, die angesprungen wird.

Siehe auch:

Jump

## 1.67 jump\_line\_end

Jump\_Line\_End

Der Cursor springt zum letzten Zeichen der Zeile

Siehe auch:

Jump\_Line\_Start

## 1.68 jump\_line\_start

Jump\_Line\_Start

Der Cursor springt zum ersten Zeichen der Zeile

Siehe auch:

Jump\_Line\_End

---

## 1.69 jump\_mark

Jump\_Mark Markierungsnummer

Springt eine zuvor gesetzte Markierung an

Siehe auch:

Set\_Mark

## 1.70 jump\_next\_fold

Jump\_Next\_Fold

Springt zur nächsten Falte

Siehe auch:

Jump\_Prev\_Fold

## 1.71 jump\_next\_tab

Jump\_Next\_Tab

Springt zum nächsten Tabulator

Siehe auch:

Jump\_Prev\_Tab

## 1.72 jump\_next\_word

Jump\_Next\_Word

Springt zum nächsten Wort (anfang)

Siehe auch:

Jump\_Next\_Word\_End

,

Jump\_Prev\_Word

,

Jump\_Prev\_Word\_End

## 1.73 jump\_next\_word\_end

Jump\_Next\_Word\_End

Springt zum nächsten Wort (ende)

Siehe auch:

Jump\_Next\_Word  
,  
Jump\_Prev\_Word  
,  
Jump\_Prev\_Word\_End

## 1.74 jump\_prev\_fold

Jump\_Prev\_Fold

Springt zur vorherigen Falte

Siehe auch:

Jump\_Next\_Fold

## 1.75 jump\_prev\_tab

Jump\_Prev\_Tab

Springt zum vorherigen Tabulator

Siehe auch:

Jump\_Next\_Tab

## 1.76 jump\_prev\_word

Jump\_Prev\_Word

Springt zum vorherigen Wort (anfang)

Siehe auch:

Jump\_Prev\_Word\_End  
,  
Jump\_Next\_Word  
,  
Jump\_Next\_Word\_End

---

## 1.77 jump\_prev\_word\_end

Jump\_Prev\_Word\_End

Springt zum vorherigen Wort (ende)

Siehe auch:

Jump\_Prev\_Word  
,  
Jump\_Next\_Word  
,  
Jump\_Next\_Word\_End

## 1.78 jump\_right\_view

Jump\_Right\_View

Der Cursor springt zur letzten, im Fenster sichtbaren Spalte

Siehe auch:

Jump\_Left\_View  
,  
Jump\_Top\_View  
,  
Jump\_Bottom\_View

## 1.79 jump\_start

Jump\_Start

Der Cursor springt zur ersten Zeile

Siehe auch:

Jump\_End

## 1.80 jump\_top\_view

Jump\_Top\_View

Der Cursor springt zur ersten, im Fenster sichtbaren Zeile

Siehe auch:

Jump\_Top\_View  
,  
Jump\_Left\_View

---

,  
Jump\_Right\_View

## 1.81 kill\_all\_folds

Kill\_All\_Folds

Nach einer Sicherheitsabfrage werden alle Falten entfernt

Siehe auch:

Kill\_Fold

## 1.82 kill\_fold

Kill\_Fold

Die Falte, in der sich der Cursor befindet, wird entfernt

Siehe auch:

Kill\_All\_Folds

,  
Make\_Fold

## 1.83 kill\_line

Kill\_Line

Kopiert die aktuelle Linie in einen Zwischenspeicher und löscht diese dann

Siehe auch:

Insert\_Bufferline

## 1.84 kill\_mark

Kill\_Mark Markierungsnummer

Löscht eine zuvor gesetzte Markierung

Siehe auch:

Set\_Mark

---

## 1.85 last\_hot\_help

Last\_Hot\_Help

Öffnet letztes HotHelp-Fenster

Siehe auch:

Hot\_Help

## 1.86 load

Load [Dateiname]

Lädt eine Datei

Eine Datei wird in das Textfenster eingelesen. Wurde der alte Text verändert, erscheint ein Requester welcher die Aktion bestätigt haben will. Wird Dateiname nicht angegeben, so erscheint ein Filerequester zur Auswahl einer Datei.

Beispiele:

```
Load;
```

```
Load "Configs/MegaEd.config";
```

Siehe auch:

Open

,

Load\_Original

## 1.87 load\_default\_prefs

Load\_Default\_Prefs

Lädt die Standarteinstellung

Die Einstellungsdatei "ENV:MegaEd/MegaEd.prefs" wird geladen.

Siehe auch:

Save\_Default\_Prefs

,

Load\_Prefs

## 1.88 load\_makefile

Load\_Makefile [Dateiname]

---

Lädt ein Makefile

Ein Makefile wird eingelesen.  
Wird Dateiname nicht angegeben, so erscheint ein Filerequester zur Auswahl einer Datei.

Beispiele:

```
Load_Makefile;  
Load_Makefile "Hello.MKE";
```

Siehe auch:

```
Save_Makefile  
,  
Module_Editor
```

## 1.89 load\_original

```
Load_Original
```

Geladene Datei wieder lesen

Eine Datei, welche gerade im Textfenster editiert wird, wird erneut eingelesen, um z.B. ungewollte Änderungen rückgängig zu machen. Wurde der alte Text verändert, erscheint ein Requester welcher die Aktion bestätigt haben will.

Siehe auch:

```
Load
```

## 1.90 load\_prefs

```
Load_Prefs [Dateiname]
```

Lädt eine Einstellungsdatei

Wird Dateiname nicht angegeben, so erscheint ein Filerequester zur Auswahl einer Datei.

Siehe auch:

```
Load_Default_Prefs
```

## 1.91 make

```
Make
```

Erstellen des Make-Programmes

---

Gemäß der Daten, die mit dem Modul-Editor angegeben wurden, werden nun alle Module ggf. übersetzt und zu Schluß gelinkt. Ein Modul wird übersetzt, wenn eine der folgenden Bedingungen erfüllt ist:

- die Objektdatei existiert nicht
- die Sourcedatei besitzt ein früheres Datum als die Objektdatei
- das Modul wurde markiert (noch nicht implementiert)

Um den Betrieb nicht weiter aufzuhalten, wird ein neuer Task gestartet, welcher ein kleines Fenster öffnet. Dieses Fenster gibt immer an, welche Aktion gerade ausgeführt wird und weist auf aufgetretene Fehler hin. Um den Übersetzungsvorgang abzubrechen, befindet sich ein entsprechendes Gadget im Fenster. Allerdings muß das Modul zu Ende übersetzt werden.

Siehe auch:

```
Modul-Editor
,
Make_All
,
Compile
,
Run_Make
, Select_Modules
```

## 1.92 make\_all

```
Make_All
```

Neuerstellen des Make-Programmes

Gemäß der Daten, die mit dem Modul-Editor angegeben wurden, werden nun alle Module übersetzt, egal ob dieses verändert wurde oder nicht.

Siehe auch:

```
Modul-Editor
,
Make
```

## 1.93 make\_fold

```
Make_Fold
```

Erzeugt Falte

Hierfür müssen Sie zuerst einen Block zwischen den beiden Zeilen markieren, die den Faltenbegin bzw. -ende darstellen sollen. Dabei müssen Sie darauf achten, daß sich die Falten nicht überschneiden. Falten innerhalb Falten

---

sind (noch) verboten.

Siehe auch:

Kill\_Fold

## 1.94 mark\_block

Mark\_Block

Block ein/aus

Die Blockmarkierung wird hiermit ein- bzw. ausgeschaltet. Wurde sie eingeschaltet, erscheint im Infofenster ein "B". Wenn Sie nun den Cursor bewegen, wird der Bereich zwischen dem Blockanfang und der Cursorposition invertiert dargestellt.

Der hervorgehobene Bereich zwischen Cursor und der Stelle an der die Markierung aktiviert wurde wird nicht nur für Blockoperationen, sondern auch für Falt- und Formatierfunktionen benützt. Diese arbeiten jedoch nur zeilenorientiert, die Spalte des Cursors wird nicht beachtet.

Siehe auch:

Copy\_Block  
,  
Insert\_Block  
,  
Cut\_Block

## 1.95 matching\_bracket

Matching\_Bracket

Sucht entsprechende Klammer

Plazieren Sie den Cursor auf eine Klammer "(", ")", "[", "]", "{" oder "}". MegaEd springt nun zu der dazugehörigen Klammer bei Beachtung von Verschachtelungen.

Siehe auch:

Check\_Brackets

## 1.96 module\_editor

Module\_Editor

Make(file) editieren

Die "Make"-Funktion richtet sich speziell an Programmierer, kann aber auch benützt werden, wenn sie mit mehreren Datei arbeiten.

---

Sie können beliebig viele Module (Dateien, Sourcefiles, Texte, ...), zwei Compiler, einen Assembler, einen Debugger, einen Linker und weitere Attribute definieren. Alle Eingaben dazu werden mit dem Moduleditor gemacht.

Der Moduleditor öffnet ein relativ großes Fenster, in dem sich einige Gadgets befinden.

Auf der linken Seite befinden sich die Angaben über die einzelnen Module:  
 Quelltexte: Die Liste zeigt alle zuvor definierten Module an. Wenn Sie eins anklicken, können Sie weitere Angaben über das Modul machen (s. u.).  
 Neu: Mit Hilfe eines Filerequesters können Sie ein Modul (eine Datei) der Liste hinzufügen.

Löschen: Das aktivierte Modul wird aus der List der Liste entfernt.

Rauf: Da die Reihenfolge auch bei anderen Programmteilen eingehalten wird, können Sie das aktivierte Modul um eine Position nach oben verschieben.

Runter: Verschiebt das aktivierte Module um eine Position nach unten.

Übersetzer: Wenn das Modul Übersetzt(kompiliert) werden soll, geben Sie hiermit den zuständigen Übersetzer( an.

Objekt-Suffix: Die zum Quelltext gehörige Objektdatei erhält die angegebene Dateiendung (z.B. ".o"). Den trennenden Punkt sollten Sie hier weglassen.

Auf der rechten Seite befinden sich die weiteren Angaben:

Programm: Der Name des ausführbaren Programms nach dem Linken.

Argumente: Diese werden dem Programm, wenn es ausgeführt wird, übergeben.

Ausgabe: Geben Sie hier ein "CON:"-Fenster an, in dem alle Ausgaben des Programmes erfolgen sollen. Achten Sie darauf, daß die Flags "AUTO/WAIT/CLOSE" gesetzt sind. Bsp.:CON:0/100/640/200/Ausgabe/AUTO/WAIT/CLOSE"

Objektudir: Wenn Sie wollen, daß alle Objektdateien in ein separates Verzeichnis geschrieben werden sollen, so können Sie dieses hier angeben.

Debugger: Geben Sie ihren Debugger mit allen dazugehörigen Argumenten ein.

Link: Geben Sie ihren Linker mit allen dazugehörigen Argumenten ein.

Cmp1/Cmp2/Asm: Um ihre Module zu übersetzten, können Sie drei verschiedene Übersetzer eingeben. Achten Sie auch auf die vollständige Argumentliste.

Link/Cmp1/Cmp2/Asm-Trans: Hinter den String-Gadgets befindet sich ein Gadget("?") mit dem Sie einen "Fehler-Umsetzer" per Filerequester auswählen können. Dieser setzt die (Fehler-)Ausgabe von Linker oder Kompiler in ein MegaEd verständliches Format um. Somit können Fehler direkt im Quelltext angesprungen und sofort verbessert werden. Sie können auch einen eigenen Umsetzter schreiben falls Ihre Programmiersprache nicht unterstützt wird. Siehe auch:

Fehler-Umsetzer

Bei den Angaben Debugger, Linker, Compiler1&2, Assembler müssen ←  
 Sie neben

dem vollständigen Pfad auch die vollständige Parameterliste für die Programme angeben. Dabei verwenden sie folgende Platzhalter:

"%e" für das ausführbare Programm

"%s" für einen Quelltext

"%l" für ein linkbares Objektfile

Siehe auch:

Save\_Makefile

,

Load\_Makefile

,

Make

```
        ,  
        Compile  
        ,  
        Run_Make  
        ,  
        Edit_A_Module  
        ,  
Select_Modules
```

## 1.97 msg

Msg Text

Gibt (Text) im Infofenster aus

Beispiele:  
Msg "Hallo!"

Siehe auch:

Alert

## 1.98 new\_window

New\_Window

Öffnet neues Textfenster

Es wird ein neues (leeres) Textfenster geöffnet, damit Sie eine neue Datei erstellen können, ohne die bereits bestehenden Texte zu beeinflussen. Sie können übrigens so viele Fenster öffnen, wie es Ihr Speicher zulässt.

Siehe auch:

Close\_Window

## 1.99 next\_window

Next\_Window

Das nächste Textfenster wird in den Vordergrund gebracht

Siehe auch:

Prev\_Window

---

## 1.100 open

Open [Dateiname]

Öffnet eine Datei

Eine Datei wird in ein neues Textfenster eingelesen. Alle Texte bleiben unverändert

Wird [Dateiname] nicht angegeben, so erscheint ein Filerequester zur Auswahl einer Datei.

Beispiele:

Open;

Open "Configs/MegaEd.config";

Siehe auch:

Load

## 1.101 open\_all\_folds

Open\_All\_Folds

Öffnet alle Falten

Siehe auch:

Close\_All\_Folds

'  
Toggle\_Fold

## 1.102 open\_server

Open\_Server

Öffnet Textmanager

Das Fenster des Textmanagers wird geöffnet. Der Textmanager zeigt alle vorhandenen Texte in einer Liste an, aus welcher sie einen Text durch Anklicken aktivieren können. Wird ein aktivierter Text nochmal angeklickt, so wird das entsprechende Textfenster in den Vordergrund gebracht. Den aktiven Text können Sie auch mit den Cursortasten wählen und mit Return in den Vordergrund bringen.

Ist dem Dateinamen ein "\*" vorausgestellt, so bedeutet dies, daß der Text verändert wurde und noch gespeichert werden sollte.

Das Textmanager-Fenster besitzt noch ein eigenes kleines Menü, mit dem Sie ein neues Fenster öffnen können, den Textmanager wieder schließen oder MegaEd beenden können.

Siehe auch:

Close\_Server

---

## 1.103 prefs

### Prefs

Öffnet Editor-Einstellungen

Mit Hilfe des erscheinenden Fensters können Sie eine Reihe von MegaEd's Eigenschaften verändern.

Links befindet sich eine Reihe von An/Aus-Gadgets:

Screenleiste frei: Alle Textfenster werden unterhalb der Screenleiste geöffnet, damit der Screen verschoben und in den Hintergrund geklickt werden kann. Allerdings ist das Textfenster kleiner.

Kontroll-Leiste: Nicht benützt

Mauszeiger weg: Wenn Sie eine Taste drücken, verwandelt sich der Mauszeiger in einen winzigen Punkt. Somit stört er bei der Eingabe nicht mehr.

Fenster volle Größe: Nicht benützt

Neue Zeilen einrücken: Wenn Sie Return drücken, wird die neue Zeile genau wie die alte Zeile eingerückt, d.h. alle Leerzeichen und Tabs am Anfang der alten Zeile werden bei der neuen Zeile eingefügt.

App-Icon immer zeigen: Normalerweise erscheint das MegaEd-AppIcon wenn Sie

Iconify

wählen. Hiermit kann es immer benützt werden.

Wort: nur Buchstaben: Einige Funktionen (z.B.

Jump\_Next\_Word

)

sind auch Wörter ausgerichtet. Ist diese Gadget aktiv, so werden nur Textstücke als Wörter angesehen, die nur Buchstaben (A-Z, Ä, Ö und Ü) enthalten. Sonst sind Wörter nur durch das Leerzeichen oder den Tab abgegrenzt.

Alert bei Make-Fehlern: Wenn ein Compiler einen Fehler findet, so merkt sich MegaEd den Fehlertext. Dieser wird, wenn der Fehler angesprungen wird im Info-Fenster angezeigt. Da dieser oft zu lang ist, erscheint hiermit zusätzlich noch ein Fehlerrequester.

BackUp erzeugen: Wenn eine verändertete Datei gespeichert wird, wird die alte Datei nicht gelöscht, sondern umbenannt. Damit können fehlerhafte Veränderungen rückgängig gemacht und versehentlich gelöschte Dateien zurückgeholt werden. Allerdings wird der doppelte Speicherplatz auf dem Laufwerk benötigt.

BackUp bei AutoSave: Beim Auto-Save wird ein Back-Up erstellt.

TAB zu Space: Wenn Sie die Tab-Taste drücken wird kein Tabulator in den Text eingefügt, sondern die entsprechende Anzahl von Leerzeichen.

File-Req. speichern: Werden die Einstellungen gespeichert, so wird auch die Fensterposition und -größe und das Dateimuster des Filerequesters gespeichert.

Auf der rechten Seite befinden sich folgende Gadgets:

Auto-Save: Nach jedem Intervall, das in Minuten angegeben werden kann, werden automatisch alle veränderten Dateien gespeichert, auf Wunsch mit vorgehender Sicherheitsabfrage.

Scrollränder: Befindet sich der Cursor innerhalb der Scrollzone, wird automatisch gescrollt um den Cursor aus dieser herauszubringen. Diese Scrollzone wird mit den Ränder für oben, links, rechts und unten angegeben.

Tab-Weite: Hiermit verändern Sie die Tabgröße.

Extra Backup-Verzeichnis: Wenn es Sie stört, die Backups im gleichen Verzeichnis zu haben wie die eigentlichen Dateien, können Sie diese auch

in ein separates Verzeichnis umlenken. Das ?-Gadget erlaubt die Auswahl mit einem Requester.

Icons: Wollen Sie keine Icons erzeugen, wählen Sie hier "Nicht sichern", ansonsten "Sichern" oder "Mit Textmerkmalen". Bei letzterem werden mit Hilfe der Tooltypes noch verschiedene Textmerkmale gesichert, wie Cursorposition, Textrand, Markierungen und Falten. Speziell wenn Sie mit Falten arbeiten, ist diese Option unverzichtbar!

Nach {}: Ist das letzte Zeichen einer Zeile eine Klammer "{", so kann hiermit ein Space oder ein Tab eingefügt werden. Für C-Programmierer wird die Strukturierung vereinfacht.

Siehe auch:

Display\_Prefs

## 1.104 prev\_window

Prev\_Window

Das vorherige Textfenster wird in den Vordergrund gebracht

Siehe auch:

Next\_Window

## 1.105 print

Print

Druckt den gesamten Text

Ist genug Speicher vorhanden, so wird ein neuer Druck-Task gestartet. Dieser öffnet ein Fenster um anzuzeigen wieviel bereits gedruckt wurde. Mit dem Schließgadget kann der Druckvorgang abgebrochen und nach einem Fehler das Fenster geschlossen werden. Läuft alles nach Plan, wird das Fenster automatisch eine Sekunde nach Druckende geschlossen. Während des Druckens kann in allen Texten weitergearbeitet werden. Bei Speichermangel wird kein neues Fenster geöffnet. In diesem Fall muß gewartet werden bis der Druckvorgang abgeschlossen ist.

Siehe auch:

Print\_Block

## 1.106 print\_block

Print\_Block

Druckt den markierten Bereich

---

Aktivieren Sie die Blockmarkierung an der Textstelle, welche eine Grenze des Druckvorgangs darsellt. Setzen Sie nun den Cursor auf die andere Grenze, so daß die markierte Text der Passage entspricht, die Sie drucken möchten. Ist genug Speicher vorhanden, so wird ein neuer Druck-Task gestartet. Dieser öffnet ein Fenster um anzuzeigen wieviel bereits gedruckt wurde. Mit dem Schließgadget kann der Druckvorgang abgebrochen und nach einem Fehler das Fenster geschlossen werden. Läuft alles nach Plan, wird das Fenster automatisch eine Sekunde nach Druckende geschlossen. Während des Druckens kann in allen Texten weitergearbeitet werden. Bei Speichermangel wird kein neues Fenster geöffnet. In diesem Fall muß gewartet werden bis der Druckvorgang abgeschlossen ist.

Siehe auch:

Print

## 1.107 replace

Replace

Öffnet Ersetzen-Requester

Es wird ein Fenster geöffnet, welches identisch mit dem der "Suchen"-Funktion (siehe

Search

), nur daß nun hier das Texteingabefeld "Ersetze:" nicht blockiert ist. Hier müssen Sie einen Text, welcher den Suchtext ersetzen soll, angeben.

Nachdem Sie den Requester bestätigt haben, öffnet sich ein neues kleines Fenster. Hiermit leiten Sie alle weiteren Aktionen:

Ersetzen: Der Text, auf den der Cursor gerade zeigt, wird durch den neuen ersetzt.

Überspringen: Der Text, auf den der Cursor gerade zeigt, bleibt unverändert.

Alles ersetzen: Da es sich hierbei unter Umständen um eine fatale Funktion handeln kann, müssen Sie das Gadget ZWEIMAL anklicken, bevor sie ausgeführt wird. Danach werden alle gefundenen Textstellen ersetzt.

Neu: Es erscheint wieder der Ersetzen-Requester, mit dem Sie neue Texte eingeben können.

Siehe auch:

Search

## 1.108 reset\_prefs

Reset\_Prefs

Alle Einstellung werden auf interne Werten zurückgesetzt

Siehe auch:

Prefs

---

```
,  
Display_Prefs
```

## 1.109 restore\_line

Restore\_Line

Zeilen Undo/Redo

Wenn sie eine Zeile verändert haben, aber merken, daß Sie dies eigentlich gar nicht wollten, können Sie die alte Zeile hiermit wiederherstellen. Wenn Sie nun merken, daß Sie die Zeile doch ändern wollen, können Sie mit dieser Funktion auch die veränderte Version der Zeile zurückholen. (Und wenn Sie nun merken...)

## 1.110 return

```
Return
```

Zeilenumbruch

Die Zeile wird bei der Cursorposition umgebrochen, d. h. eine neue Zeile wird eingefügt. Alle Zeichen ab dem Cursor werden in die neue Zeile übernommen.

Siehe auch:

```
Return_Nosplit  
,  
Return_Revert_Insertmode
```

## 1.111 return\_nosplit

```
Return_Nosplit
```

Gehe zu Anfang der nächsten Zeile (ohne Umbruch)

Siehe auch:

```
Return  
,  
Return_Revert_Insertmode
```

## 1.112 return\_revert\_insertmode

---

Return\_Revert\_Insertmode

Zeilenumbruch

Führt dieselbe Funktion wie

Return

aus, nur daß die neue Zeile

eingerrückt wird, wenn dies normal NICHT der Fall ist. Die Einstellung "Neue Zeilen einrücken" wird also praktisch umgekehrt (siehe Prefs

Siehe auch:

Return

,  
Return\_Nosplit

### 1.113 return\_split

Return\_Split

siehe Return

### 1.114 run

Run [Dateiname]

Startet ein Programm

Hiermit können Sie ein beliebiges ausführbares Programm starten. Es muß nicht gewartet werden bis das Programm beendet wird, da es als separater Task gestartet wird.

Wird Dateiname nicht angegeben, so erscheint ein Filerequester zur Auswahl einer Datei.

Beispiele:

Run "SYS:Tools/Calculator";

Run;

Siehe auch:

Execute

### 1.115 run\_make

Run\_Make

Startet das Make-Programm

Siehe auch:

---

```
Debug_Make
,
Module_Editor
```

## 1.116 save

```
Save
```

Speichert eine Datei

Die Datei des Textfenster wird unter dem bereits existierenden Namen gespeichert. Ist die Datei noch namenlos, erscheint ein Filerequester.

Siehe auch:

```
Save_As
,
Save_All
```

## 1.117 save\_all

```
Save_All
```

Speichert alle Dateien

Alle veränderten Dateien werden unter ihren bereits existierenden Namen gespeichert.

Siehe auch:

```
Save
,
Save_As
```

## 1.118 save\_as

```
Save_As [Dateiname]
```

Speichert eine Datei unter neuem Namen

Es erscheint ein Filerequester zur Auswahl des neuen Dateinamens, unter dem die Datei gespeichert werden soll.

Ist Dateiname bereits angegeben, erscheint natürlich kein Filerequester.

Beispiele:

```
Save_As;
Save_as "Configs/MegaEd.config";
```

Siehe auch:

---

```
Save
,'
Save_All
```

### 1.119 save\_default\_prefs

```
Save_Default_Prefs
```

Speichert die Einstellungen als Standard

Die Einstellungen werden unter "ENVARC:MegaEd/MegaEd.prefs" und zusätzlich unter "ENV:MegaEd/MegaEd.prefs" gespeichert. Beim nächsten Start von MegaEd werden diese Einstellungen geladen.

Siehe auch:

```
Save_Prefs
,'
Load_Default_Prefs
,'
Load_Prefs
```

### 1.120 save\_icon

```
Save_Icon
```

Erstellt ein Icon zur Datei

Die gerade bearbeitete Datei erhält ein Icon. Besitzt sie bereits eins, so wird es durch ein neues ersetzt.

Siehe auch:

```
Prefs
```

### 1.121 save\_makefile

```
Save_Makefile
```

Speichert das Makefile

Das Makefile wird unter dem bereits existierenden Namen gespeichert. Ist die Datei noch namenlos, erscheint ein Filerequester.

Siehe auch:

```
Save_Makefile_As
```

---

## 1.122 save\_makefile\_as

Save\_Makefile [Datei]

Speichert das Makefile unter neuem Namen

Das Makefile wird unter dem angegebenen Namen oder, wenn dieser nicht angegeben wurde, durch den mit Hilfe eines Filerequester gewählten Namen gespeichert.

Siehe auch:

Save\_Makefile

## 1.123 save\_prefs

Save\_Prefs [Datei]

Speichert die Einstellungen

Die Einstellungen werden unter dem angegebenen Namen oder, wenn dieser nicht angegeben wurde, durch den mit Hilfe eines Filerequester gewählten Namen gespeichert.

Siehe auch:

Save\_Default\_Prefs  
,  
Load\_Prefs  
,  
Load\_Default\_Prefs

## 1.124 save\_program

Save\_Program

Speichert die Make-Programm

Das Programm, welches durch den  
Make  
-Vorgang bereits erstellt wurde, wird  
hiermit gespeichert. Den neuen Dateinamen geben Sie mit einem Filerequester  
an.

Siehe auch:

Module\_Editor

## 1.125 scroll

---

Scroll Spalten, Zeilen

Scrollen

Der Fensterinhalt wird um Spalten nach rechts bzw. bei negativen Werten nach links und um Zeilen nach unten bzw. bei negativen Werten nach oben gescrollt ("gerollt").

Beispiel:

Scroll 0,1;

Siehe auch:

Scroll\_Page\_Up, Scroll\_Page\_Down

### 1.126 scroll\_down\_page

Scroll\_Down\_Page

Der Fensterinhalt wird um eine Seite nach unten gescrollt

Siehe auch:

Scroll\_Page\_Up

Scroll

### 1.127 scroll\_up\_page

Scroll\_Up\_Page

Der Fensterinhalt wird um eine Seite nach oben gescrollt

Siehe auch:

Scroll\_Page\_Down

Scroll

### 1.128 search

Search

Öffnet Such-Requester

Das sich öffnende Fenster enthält folgende Gadgets:

Suche: Der Text nach dem gesucht werden soll

1, 2, 3, 4, 5, 6, T: MegaEd erlaubt pro Text sechs verschiedene Paare für Such- und Ersetztext. Um zwischen diesen umschalten zu können, wählen sie eins der Gadgets "1" bis "6" an, wobei das gerade gewählte Paar durch eine doppelte Umrandung des jeweiligen Gadgets gekennzeichnet ist. Mit "T" können Sie Such- und Ersetztext tauschen.

Groß/Klein beachten: Wenn dies gewählt ist, wird auf die Groß- und Kleinschreibung geachtet, welche exakt übereinstimmen müssen, d. h. das Wort

---

"Banal" wird nicht gefunden, wenn der Suchtext "banal" lautet.

Muster (\*) verwenden: geplant für V2.0

Nur ganze Wörter: Der Suchtext bezieht sich nur auf ganze Wörter. Wenn der Suchtext in einem Wort enthalten ist, wird dieser nicht gefunden. Bsp.: Der Suchtext "ist" wird in dem Text "Schule ist Mist" nur einmal gefunden.

In allen Texten: geplant für V2.0

Cursor, vorwärts: Der Ausgangspunkt der Suche ist die Cursorposition, die Suchrichtung ist vorwärts.

Cursor, rückwärts: Der Ausgangspunkt der Suche ist die Cursorposition, die Suchrichtung ist rückwärts.

Textanfang: Der Ausgangspunkt der Suche ist der Textanfang, die Suchrichtung ist vorwärts.

Textende: Der Ausgangspunkt der Suche ist das Textende, die Suchrichtung ist rückwärts.

Siehe auch:

```
Search_Next
,
Search_Prev
,
Replace
```

## 1.129 search\_next

```
Search_Next
```

Das nächste Vorkommen des Suchtextes wird gesucht

Siehe auch:

```
Search_Prev
,
Search
,
Replace
```

## 1.130 search\_prev

```
Search_Prev
```

Das vorherige Vorkommen des Suchtextes wird gesucht

Siehe auch:

```
Search_Next
,
Search
,
Replace
```

---

### 1.131 set\_clip

Set\_Clip [Clip]

Die Clipboard-Unit wird verändert

Mit dieser Funktion können Sie 256 verschiedene Blöcke benutzen, indem Sie hier einen Clip zwischen 0 und 255 angeben. Der Clip 0 wird z. B. auch für die Shell verwendet, so daß ein Austausch problemlos ist.

Wird Clip nicht angegeben, so erscheint ein Requester, welcher die Eingabe nachträglich ermöglicht.

Beispiel:

```
Set_Clip 1;
```

```
Set_Clip;
```

Siehe auch:

Mark\_Block

### 1.132 set\_mark

Set\_Mark Markierungsnummer

Setzt Markierung

Die Markierung Markierungsnummer wird gesetzt, d. h. MegaEd merkt sich die Cursorposition und kann später wieder angesprungen werden. Es werden maximal 20 Markierungen gespeichert.

Siehe auch:

Jump\_Mark

,

Kill\_Mark

### 1.133 set\_right\_border

Set\_Right\_Border Spalte

Setzt rechten Rand

Einige Funktionen (Wordwrap, Formatierungen) benötigen einen rechten Rand. Dieser wird hiermit auf Spalte gesetzt. Fehlt diese Angabe, so wird ein Requester geöffnet der eine Eingabe erlaubt.

Siehe auch:

Format\_Left

---

## 1.134 sort\_lines

Sort\_Lines

Zeilen sortieren

Alle Zeilen von der Blockmarkierung bis zur aktuellen Zeile werden gemäß der ASCII-Tabelle sortiert.

Siehe auch:

Mark\_Block

## 1.135 tab

Tab

Ein Tabulator wird in den Text eingefügt

## 1.136 toggle\_fold

Toggle\_Fold

Falte schließen/öffnen

Befindet sich der Cursor auf einer geschlossenen Falte, wird diese geöffnet, ist sie geöffnet wird sie geschlossen.

Siehe auch:

Close\_All\_Folds

,

Open\_All\_Folds

## 1.137 write\_mode

Write\_Mode [Modus]

Einfügen/Überschreiben

Hiermit wechseln Sie den Schreibmodus. Für Modus können Sie entweder Replace (Überschreiben), Insert (Einfügen) oder auch nichts angeben, dann wird der Modus getauscht.

Beispiel:

```
Write_Mode Replace;
```

```
Write_Mode;
```

---

### **1.138 zoom\_all\_windows**

Zoom\_All\_Windows

Alle Fenster werden auf maximale Größe gebracht

Siehe auch:

Zoom\_Window

### **1.139 zoom\_window**

Zoom\_All\_Windows

Das aktuelle Fenster wird auf maximale Größe gebracht

Siehe auch:

Zoom\_All\_Window

---