

AmisED

Willem Mestrom

COLLABORATORS

	<i>TITLE :</i> AmisED		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Willem Mestrom	April 14, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AmisED	1
1.1	AmisED manual	1
1.2	The Amis project	2
1.3	WYSIWYG	3
1.4	Requirements	4
1.5	Registration	4
1.6	What's new	6
1.7	What's new in AmisED	6
1.8	What's new in this guide	6
1.9	Disclaimer	7
1.10	The AmisED homepage	7
1.11	Startup options	7
1.12	The editor	8
1.13	The menus explained	9
1.14	Menu explanation	12
1.15	Windows	16
1.16	'Find...' window	17
1.17	'Find functions...' window	18
1.18	'Undo list...' window	18
1.19	'Macro...' window	19
1.20	'Information...' window	19
1.21	'About' window	20
1.22	'Project Manager' window	20
1.23	'Window Manager' window	20
1.24	'Toolbar...' window	21
1.25	'Editor...' window	21
1.26	'Smart Indent...' window	22
1.27	'Current window settings...' window	23
1.28	'Screen...' window	23
1.29	'Palette...' window	24

1.30	'Default fonts...' window	24
1.31	'Keyboard...' window	25
1.32	'Menu...' window	25
1.33	'Programming' window	26
1.34	'Miscellaneous...' window	27
1.35	'Internal command...' window	27
1.36	'Execute Command...' window	27
1.37	'Edit commands...' window	28
1.38	Getting started	28
1.39	Save your files after a failure	29
1.40	Credits	29
1.41	Known bugs	30
1.42	ARexx	30
1.43	Internal Commands	30
1.44	Alphabetical list of the Internal Commands	32
1.45	Available Internal Commands	35
1.46	Internal command: CursUp	38
1.47	Internal command: CursDown	38
1.48	Internal command: CursLeft	38
1.49	Internal command: CursRight	39
1.50	Internal command: CursMode	39
1.51	Internal command: WordWrap	39
1.52	Internal command: Goto	40
1.53	Internal command: SetBookmark	40
1.54	Internal command: Backspace	41
1.55	Internal command: Delete	41
1.56	Internal command: Line	41
1.57	Internal command: Insert	41
1.58	Internal command: File	42
1.59	Internal command: EOLMode	42
1.60	Internal command: CursorWrap	43
1.61	Internal command: WriteProtect	43
1.62	Internal command: SetTab	43
1.63	Internal command: BackUps	44
1.64	Internal command: AutoSave	44
1.65	Internal command: AutoIndent	44
1.66	Internal command: CreateIcons	45
1.67	Internal command: FastMode	45
1.68	Internal command: OverWrite	45

1.69 Internal command: Query	46
1.70 Internal command: GetWord	46
1.71 Internal command: GetLine	46
1.72 Internal command: GetASCII	46
1.73 Internal command: Toolbar	47
1.74 Internal command: Codebar	47
1.75 Internal command: Statusbar	47
1.76 Internal command: Viewerbar	48
1.77 Internal command: Viewer	48
1.78 Internal command: Request	48
1.79 Internal command: Message	49
1.80 Internal command: Find	49
1.81 Internal command: Replace	50
1.82 Internal command: FindFunctions	50
1.83 Internal command: Block	51
1.84 Internal command: BlockLayout	51
1.85 Internal command: BlockMode	52
1.86 Internal command: CutBlock	52
1.87 Internal command: CopyBlock	52
1.88 Internal command: PasteBlock	53
1.89 Internal command: Undo	53
1.90 Internal command: Redo	53
1.91 Internal command: Macro	54
1.92 Internal command: Select Window	54
1.93 Internal command: Window	54
1.94 Internal command: ArrangeWindows	55
1.95 Internal command: Compile	55
1.96 Internal command: Assemble	56
1.97 Internal command: Link	56
1.98 Internal command: Debug	57
1.99 Internal command: Run	57
1.100 Internal command: ParserError	58
1.101 Internal command: Project	58
1.102 Internal command: ExecuteCommand	59
1.103 Internal command: InternalCommand	59
1.104 Internal command: Shell	59
1.105 Internal command: Rexx	59
1.106 Internal command: ShowStatistics	60
1.107 Internal command: ShowWindowManager	60

1.108	Internal command: ShowAbout	60
1.109	Internal command: ToolbarSettings	60
1.110	Internal command: EditorSettings	61
1.111	Internal command: KeyboardSettings	61
1.112	Internal command: MenuSettings	61
1.113	Internal command: FontSettings	62
1.114	Internal command: ScreenSettings	62
1.115	Internal command: ProgramSettings	62
1.116	Internal command: SmartIndentSettings	63
1.117	Internal command: MiscSettings	63
1.118	Internal command: Settings	63
1.119	Internal command: Iconify	64
1.120	Internal command: Quit	64
1.121	Internal command: ResultToVar	64
1.122	Internal command: Help	64
1.123	Internal command: Graphics	65
1.124	Internal Command: Fold	65
1.125	Internal Command: UnFold	65
1.126	Internal Command: Print	66
1.127	AmigaDOS Arguments	66
1.128	Index	66

Chapter 1

AmisED

1.1 AmisED manual

AmisED V1.20
Copyright © 1993-1999 Willem Mestrom

The Amis~project ~~
 Purpose, philosophy and features.

What's new
 What's new since version 1.03.

Requirements ~ ~~
 The minimum required system.

Registration
 How to obtain the full version.

Homepage & maillist
 Stay up-to-date.

Disclaimer
 Just in case.

Startup options
 CLI and Workbench startup options.

The editor
 A general overview.

Menus
 The menus explained.

Windows ~~
 The windows explained.

Internal~Commands ~

All about the internal commands.

ARexx

The ARexx facilities of AmisED.

Save your files

Important after a software failure.

Credits~~~~~ ~~~~~

The people who made it possible.

Bugs~~~~~ ~~~~~

Where's the anti-spray. (all 68060
owners, please look here!).

Index

Where to find...

Manual by: Willem Mestrom & Vincent Groenewold (supernov)

1.2 The Amis project

Philosophy:

Some men see things as they are, and say 'why ?'.
I dream things that never were, and say 'why not ?'.

After years of using my Commodore 64 and my Amiga to learn all about the assembler language, I dreamt of a program which had it all. Now, after a development period of 3 years, you have the first steps of this dream in your hands. Amis (AMiga Integrated System) is a program capable of editing multiple file formats with the power of an unique, easy to use graphical user interface using normal intuition functions.

A list of the most important features of AmisED:

- o WYSIWYG
 handling of AmigaGuide, ANSI, ASCII, C and HTML documents.
- o A special
 codebar
 beneath the WYSIWYG environment allowing normal ASCII
 editing. The codebar height is adjustable.
- o Each viewer has its own special preferences.
- o Recognition of the type of text you want to load.
- o Possibility to program your own viewer. Examples are included.
- o A Project manager
 for documents which belong to a certain subject.
- o A Window manager

- o for easy handling opened/hidden windows.
- o Many user-definable preferences to let the user create his own best suitable editor environment (toolbar, menu, keyboard, etc.).
- o Folding
 - .
- o Undo/Redo
 - .
- o Real tabs with a possibility to use spaces.
- o Several ways of wordwrapping
 - .
- o Hunt file option for easy opening include files used in C documents.
- o Many internal commands which simultaneously represent the ARexx commands.
- o Each window has its own preferences.
- o Window position snap-/unsnapshotting
 - .
- o Macro recording/playing and adding short-keys to macros.
- o Bookmark support.
- o Smart indent with unique user-definable preferences.
- o Fast search/replace function with wildcards option.
- o Fast loading and initializing of documents.
- o Extra Dutch, German and Italian localization.
- o XPK crunch support.
- o Powerfull compiler and assembler options such as an error parser.

1.3 WYSIWYG

WYSIWYG stands for What You See Is What You Get. This means that the text you are editing and is shown by AmisED will look the same as in the programs where it is meant to be shown. In practice this is not entirely true, because there are many programs which have a style of their own. Besides that they aren't always following standard rules. So what you are editing in AmisED will look the same for many programs, but a little different in others. The functions (like buttons in AmigaGuide or links in HTML) however should work exactly the same in all programs. With the

making of AmisED we tried to follow the standards as much as possible and to look at some of the most used programs:

- AmigaGuide for the AmigaGuide viewer
- AWeb for the HTML viewer and
- Various comm. programs for the ANSI viewer.

1.4 Requirements

The minimum required system to run AmisED:

- o An Amiga with 68020 micro-processor or the UAE '020 compatible emulator.
- o Workbench/Kickstart 3.x.

This system is fast enough for a smooth way of editing your text in most editors. However, if you think it should be faster, just turn on

```

'fast mode'
  by pressing the key combination Ctrl-f or to select it via the
settings requester
'Editor (defaults)...'
.

```

1.5 Registration

To improve the quality of AmisED, a better development system is required. Because I'm still a student and don't have a fixed income I can't buy this system without your help. If you like AmisED and want it to get better in future releases then please support it by donating a small fee. In return you will get a personal registered keyfile send to you on disk or by e-mail. Please allow a delivery period of 1 month.

Because of account transfer and Post&Package costs, the price of AmisED differs for some parts of the world. Please look in the price table below for your region:

Your region	Price (by cash)	Price (by account)
The Netherlands	F1 25,-	F1 25,-
Europe	DM 25,-	DM 39,-
	UK £12	UK £18
	US \$18	US \$28
	F1 27,-	F1 42,-
Outside Europe	DM 28,-	DM 42,-
	UK £13	UK £19
	US \$20	US \$30
	F1 32,-	F1 47,-

> If you want to pay AmisED by cash, fill in the following orderform and send it together with the money to the following address:

Willem Mestrom
Zevenwouden 237
3524 CR Utrecht
The Netherlands

I want to become a registered user of AmisED.
Here is the information you need to know about me:

Name: _____

Address: _____

Country: _____

E-mail: _____

Hardware: _____

Suggestions: _____

I'll pay the shareware fee by cash...

- from The Netherlands Fl 25,-
- from Europe DM 25 / £ 12 / US \$18 / Fl 27,-
- from outside Europe DM 28 / £ 13 / US \$20 / Fl 32,-

Please send me the keyfile...

- in uuencoded form to my E-mail address.
 - on a disk to my postal address.
- I promise not to spread my keyfile or install it on computers other than my own.

(Date)

(Signature)

> If you want to pay by account from within the Netherlands:

Maak Fl 25,- over op giro rekening 4302419 van de POSTBANK. Vermeld bij de overschrijving je naam en adres waar je de keyfile wilt ontvangen. Je hoeft dan geen 'orderform' in te vullen. Eventueel kan je de keyfile ook via het Amiga Magazine BBS verkrijgen. Stuur me dan even een E-mailtje en vermeldt dat bij de overschrijving.

> If you want to pay by account from outside the Netherlands:

Transfer the fee to the following account...

Account Nr: 4302419
Willem Mestrom
Zevenwouden 237
3524 CR Utrecht
The Netherlands
Swift code: ING B NL 2-A

Don't forget to mention your full name and address !

1.6 What's new

What's new in
AmisED
and what's new in
this guide
.

1.7 What's new in AmisED

Look
here
for a list of the bug-fixes in this version.

Improved since AMIS 1.03

- o AmisED is now a lot more stable with blocks cutting / pasting and file insertion.
- o The internal working of AmisED is changed quite a lot for faster and even more 'natural' working.
- o Memory usage is reduced considerably.

1.8 What's new in this guide

- o New/improved subjects since AMIS.guide 1.03
- o What's new
 - What's new since version 1.03.
- o Internal commands
 - All internal commands (with some new options)

1.9 Disclaimer

This software is provided as-is, without warranty of any kind, either expressed or implied. In no event will the author be liable for direct, indirect, incidental or consequential damages or data loss resulting from the use or application of this software. The entire risk as to the results and performance of this software is assumed by the user! As for the registration procedure, the author can not be held responsible for any loss of money during the transfer.

If you experience any problems during the use of AmisED or the registration procedure, please write a letter containing the exact problem to the author's postal address or to supernov@zeelandnet.nl.

1.10 The AmisED homepage

AmisED is in constant development. If you want to stay up-to-date about the current releases and those in development, than please have a look at the official AmisED homepage <http://people.zeelandnet.nl/supernov/amis.htm>.

It's also possible to ask the author and beta-testers questions about problems or anything else related to AmisED via the new mailing list. To subscribe, send your request to majordomo@thule.no. For help, send HELP in the bodytext.

1.11 Startup options

AmisED can be started from the Workbench or from the Shell. If you start AmisED from the workbench you can open files with the normal shift-click procedure. The tooltypes will be read when started from the Workbench, but also when started from the Shell. The tooltypes are:

PROJECT	The project you want to load on startup.
PORTNAME	The name of the ARexx port (Default: "AMISPORT.x")
PROJECTMANAGER	Show the 'Project Manager' after bootup (Default: OFF).
STARTICONIFIED	Start AmisED in iconified mode (no screen or files will be
SETTINGS	The preference file you want to be loaded.
TAB	The default tab size (Default: 8).
	opened, default: OFF).
PROJECTDIR	The directory where you keep your project files. The
	projects will be displayed in the Project Manager.

From the Shell you can also give some startup options. AmisED uses the normal DOS template function for the argument parsing, so you can give the arguments in the same way as for normal DOS commands.

The template: `FILE,PROJECT/K,PORTNAME/K,PROJECTMANAGER/S,ICONIFIED/S,STARTNEW/S,SETTINGS/K,TAB/N`

All options are the same as the tooltypes, where FILE is the file you want to open on startup. STARTNEW will make sure AmisED will always start a new

copy, no matter whether an AmisED already runs or not. This is very useful for programs who call the editor and expect them to return when you are ready. If STARTNEW is not given and there is already a copy of AmisED running, the new AmisED will return almost immediately and cause these programs to fail. The STARTNEW option is also useful if you want to start it from within an ARExx script without a requester being displayed if there is already an AmisED running.

1.12 The editor

The basic editor is quite simple. There is a cursor and a scroll bar on the right side. You can simply type your text and save it using the menu. But there are a lot of extra's available. If you look in the 'Window' menu you will find the item 'Bars'. Here you can switch the following bars on and off: the toolbar, the viewerbar, the statusbar and the codebar.

The toolbar is a bar in top of the editor window with some buttons in it. Each button represents a function, like 'Save', 'Load' or 'Close'. Using this bar you have some functions available with one mouse-click. This bar is user definable, it can be changed using the 'Toolbar...' item in the 'Settings' menu.

The viewerbar is a bar under the toolbar. Here you can find some functions that are specifically for the active editor. In normal viewing mode no bar will be visible. In ANSI mode there are buttons for the text color and style. In AmigaGuide mode you'll find a more complex viewerbar with functions to create and modify buttons, text color and text styles. The HTML viewerbar is the most complex one.

The statusbar is a bar under the viewerbar. In this bar some information is displayed like the line and column of the cursor position. It looks like this:

(example)

```
C: 001    L: 00001    P: 000000    V: Normal    LOCK FAST OVER DWRAP VERT
```

C: The column the cursor is in.

L: The line the cursor is on.

P: The character number of the character under the cursor.

V: The viewer this text is displayed with.

LOCK: Displayed when the window is 'locked' from user input.

FAST: Displayed when the text is displayed in fast mode (No fancy graphics).

WRAP: In this mode AmisED automatically breaks a line when you reach the window border (or the column set in the editor settings).

DWRAP: In this mode AmisED automatically breaks lines when they are too long, the break position is automatically changed when you change the line.

HORZ: Blocks are now in the normal mode.

VERT: Blocks are now in the 'vertical' mode (just try and you'll see the difference).

The codebar is a bar at the bottom of the window. In this bar you can see the same line as the one where the cursor is on, but not in the same style. For example commands in an AmigaGuide text will not be visible in the editor window, but will be visible in the codebar. On the right side of this bar

there is number showing the ASCII code of the character the cursor is on. In the codebar you can also see the cursor (in a different color). If you click in the codebar the color of this cursor will change indicating that it is active. Now you can move the cursor to characters the cursor would normally 'jump over', in other words...this bar is a little ascii editor. You are able to adjust its height via the 'Editor (defaults)...' item in the settings menu.

An other 'feature' of the editor is the pulldown menu . This menu provides a lot of editing functions like 'Find', 'Cut', 'Copy', etc. This menu is also user definable, it can be changed via the 'Menu...' settings window.

There are also a lot of information and configuration windows available, for a complete list please look at the Windows section.

1.13 The menus explained

File operations:	Editing:	Searching:
New		
Cut		
Find...		
Open...		
Copy		
Load...		
Paste		
Find next		
Revert...		
Save block...		
Find prev		
Erase		
Replace & next		
Save		
Replace & prev		
Save As...		

Block layout »

Select all

Go to line...

Print

Select from cursor

Go to bookmark »

Set bookmark »

Clear

Undo

Close

Undo list...

Find functions

Close all

Redo

Hide

Insert file...

Add to project

Insert ASCII code

Show ASCII code

Project Manager

About

Information...

Iconify

Quit

Window: Miscellaneous: Macro:

Viewer »

Internal command...

Start recording

Stop recording

Bars »

Shell

Execute command...

Play

Next

Play many...

Previous

Help

Hunt file

Macro window

Arrange »

Create Guide Index

Window manager

Expand Tabs

Insert current date

Insert current time

Folding »

Programming: Settings:

Run

Toolbar...

Editor (defaults)

Compile

Smart Indent...

Assemble

Keyboard...

Link

Menu...

Screen...

Debug

Palette...
 Default fonts...
 First error
 Programming...
 Next error
 Miscellaneous...
 Prev. error
 Current window »
 Load settings...
 Save settings
 Save settings as...

1.14 Menu explanation

	File operations:
New	- Opens a new, blank window.
Open	- Opens a new window and filerequester and loads the selected file in the new window.
Load	- Opens the filerequester and loads the selected file in the current window.
Revert	- Loads the last saved version of the current loaded file.
Save	- Saves the current edited text.
Save as	- Opens the filerequester to be able to save the current edited text under a different name.
Print	- Prints the current edited text.
Clear	- Clears the current window. You'll be warned first.
Close	- Closes the current window.
Close all	- Closes all opened windows.
Hide	- Hides the current window. Select 'Reveal' in the window manager to show it again.
Add to project	- Adds the current file to an opened project.
Project Manager	- This opens the project manager in which you are able to manage the documents which belong together.
About	- Opens the about requester with AmisED information.
Iconify	- Hides the AmisED environment and puts the amis icon on the workbench. Double click to activate the environment again. When iconifying no data will be lost (read disclaimer).

- Quit - This is the way to exit the AmisED program. You will be warned when edited data wasn't saved.
- Editing:
- Cut - Removes the selected piece of text and saves it on the clipboard.
- Copy - Copies the selected piece of text to the clipboard.
- Paste - Pastes the contents of the clipboard.
- Save block - Saves the selected piece of text via a filerequester.
- Erase - Removes the selected piece of text. This function does not copy the text to the clipboard.
- Block layout - Allows center, left or right alignment for the selected piece of text.
- Select all - Selects all text.
- Select from cursor - Selects all text from the current cursor position to the end of the document.
- Undo - Undoes a keyboard action.
- Undo list - Show a list of keyboard actions in the undo buffer
- .
- Redo - 'Redoes' a keyboard action removed with the undo function.
- Insert file - Opens a filerequester and inserts the selected file after the current cursor position.
- Insert ASCII code - Opens a requester in which you are able to type an ASCII code. The corresponding character will be inserted after the current cursor position.
- Show ASCII code - Displays the ASCII code corresponding to the character on the current cursor's position.
- Information - Displays information about the current document. In here you are also able to change its tooltypes.
- Searching:
- Find - Opens the find-requester in which you are able to fill in the text you want to search or replace.
- Find next - Searches the next occurrence (relative to the cursor position) of the text you want to find.
- Find prev - Searches for the previous occurrence (relative to the cursor position) of the text you want to find.
- Replace & next - (See Find next) and replaces it with the text you entered in the find-requester
- .
- Replace & prev - (See Find prev) and replaces it with the text you entered in the find-requester
- .
- Go to line - Opens a requester in which you are able to fill in the line number you want to go to.
- Go to bookmark - Goes to the bookmark you've set with Set bookmark.
-

Set bookmark - Sets a bookmark at the beginning of the current line. A bookmark is a kind of flag which represents the current line number.

Find functions - Opens the find functions window.

Window:

Viewer - Here you will find all the viewers AmisED supports. Selecting one of them will switch to that viewer.

Bars - Here you are able to turn the 4 different bars on or off.

Next - This will activate and pop up the next window.

Previous - This will activate and pop up the previous window.

Arrange - Here you will find 5 different ways of arranging the opened windows for a better overview.

Window Manager - This will open the Window Manager.

.

Miscellaneous:

Internal command - Opens the Internal Command window.

Shell - Opens a shell window on the AmisED public screen.

Execute command - Opens a window in which you are able to execute a normal DOS command.

Help - Opens the AmisED.guide on the AmisED public screen.

Hunt file - Used in the C-viewer. If you place the cursor on an include command and select this option, AmisED will 'hunt' and open the include file.

Create Guide Index - Opens a new window and creates an AmigaGuide index.

Expand Tabs - Replaces all tabs inside a document with normal spaces.

Insert current date - Inserts the current date.

Insert current time - Inserts the current time.

Folding - Used in the AmigaGuide viewer. If you place the cursor inside a node and select this option, the node will be hidden with only the node-name and lines remaining. Selecting it again will reveal the text.

Macro:

Start recording - Puts AmisED in macro recording mode. Keyboard actions will be recorded.

Stop recording - Stops recording keyboard actions.

Play - Plays the current active macro. See macro window.

-
- Play many - Opens a window in which you are able to give how many times AmisED should replay the current active macro.
- Macro window - Opens the macro window
.
- Programming:
- Run - Runs (if compiled) the program belonging to the current loaded document.
- Compile - Executes the chosen compiler to compile the current loaded document.
- Assemble - Executes the chosen assembler to assemble the current loaded document.
- Link - Executes the chosen linker to link the current loaded document.
- Debug - Executes the chosen debugger to debug the current loaded document.
- First error - Jumps to the first error encountered during compiling or assembling.
- Next error - Jumps to the next error encountered during compiling or assembling.
- Prev error - Jumps to the previous error encountered during compiling or assembling.
- Settings:
- Toolbar - Opens the toolbar settings window.
- Editor - Opens the editor settings window.
- Smart indent - Opens the smart indent settings window.
- Keyboard - Opens the keyboard settings window.
- Menu - Opens the menu settings window.
- Screen - Opens the screen settings window.
- Palette - Opens the palette settings window.
- Default fonts - Opens the default fonts
-

settings window.
Programming - Opens the programming settings window.
Miscellaneous- Opens the miscellaneous settings window.
Current window - Opens the current window settings window.
Load settings - Opens filerequester and loads the selected settings file.
Save settings - Saves the current settings.
Save settings as - Opens filerequester and saves the current settings under the selected name.

1.15 Windows

Editing:
Find...
Find functions...
Undo list...
Macro...
Information:
Information...
About
Managers:
Project Manager
Window Manager
Settings:
Toolbar...
Editor...
Smart Indent...
Keyboard...
Menu...
Screen...

Palette...
 Default fonts...
 Programming...
 Miscellaneous...
 Current window...
 Other:
 Internal command...
 Execute command...
 Edit commands...

1.16 'Find...' window

The 'Find...' window can be used for find and replace functions.

The gadgets:

Gadget	Function
Find	The string you want to find.
Change To	The string you want to replace the find-string with.
Case sensitive	Upper case is not the same as lower case ('a' != 'A').
Whole words	Only look for the find-string as a whole word.
Use patterns	When this options is selected AmisED will use the AmigaDOS pattern matching functions to find the pattern you entered in the find gadget. (See below for more information about AmigaDOS patterns).
Find Next	Find the next occurrence of the find-string in the text.
Find Prev	Find the previous occurrence of the find-string in the text.
Repl Next	Replace the string and find the next.
Repl Prev	Replace the string and find the previous.
Replace All	Replace all occurrences of the find-string in the text by the replace string.
Cancel	Close this window.

AmigaDOS patterns:

?	Matches a single character.
#	Matches the following expression 0 or more times.
(ab cd)	Matches any one of the items seperated by ' '. Note: The original document has a typo "seperated".
~	Negates the following expression. It matches all strings that do not match the expression (aka ~(foo) matches all strings that are not exactly "foo").
[abc]	Character class: matches any of the characters in the class.
[~bc]	Character class: matches any of the characters not in the class.
a-z	Character range (only within character classes).
%	Matches 0 characters always (useful in "(foo bar %)").
*	Synonym for "#?", not available by default in 2.0. Available

as an option that can be turned on.

"Expression" in the above table means either a single character (ex: "#?"), or an alternation (ex: "#(ab|cd|ef)"), or a character class (ex: "#[a-zA-Z]").

Sometimes you might have to use some tricks to find the things you want, for example:

- searching for '(' becomes: [(]
- searching for ')' becomes: [)]
- searching for '[' becomes: [[]
- searching for ']' becomes: ~ (~ ([\ - ^]) | \ | ^)
 - > all that is not not '\', ']' or '^' and not '\ or '^'.
 - (ASCII: '\'=92 '['=93 and '^'=94).
- searching for '~' becomes: ~ (~ ([a ~]) | a)
 - > all that is not not 'a' or '~' and not 'a'.

Ofcourse you can also switch of the use of patterns, then you can simple search for ']' using ']'. :-))

1.17 'Find functions...' window

The 'Find functions...' window can be used for some special find functions. Built in there is a find function for AmigaGuide nodes. If you select this one and then click on the 'Ok' button you will get a list of all AmigaGuide nodes in the document you are editing. Other find functions can be added with the 'Add' button. The find functions are GoldED compatible.

The gadgets:

Gadget	Function
Ok	Activate this find function.
Add	Add a find function.
Remove	Remove the selected find function.
Cancel	Close this window.

1.18 'Undo list...' window

The 'Undo list...' window shows the list of undo's that are still in the undo buffer.

The gadgets:

Gadget	Function
Undo	Perform the LAST command in the undo list (normal Undo).

1.19 'Macro...' window

The 'Macro...' window can be used to create and edit macro's.

The gadgets:

Gadget	Function
Add	Create a new macro (Name: 'New Macro').
Remove	Remove the selected macro.
Move up	Move the selected macro one place up in the list.
Move down	Move the selected macro one place down in the list.
Play Macro	Play the selected macro.
Record	Start or continue recording the selected macro.
Stop recording	Stop recording the macro.
Edit	Edit the selected macro, you will get an Edit commands... window to do this.
Short cut	Assign a short-cut to the selected macro, you will get the ' Keyboard... ' window to do this (all you have to do is press the key / key-combination you want the macro to be assigned to).
No short cut	Remove the short-cut for the selected macro.
Load	Load a macro from disk.
Save	Save a macro to disk.

1.20 'Information...' window

The 'Information...' window gives you information about the text you are editing. ←

Gadget	Function
Name	The name of the current document.
Path	The location where the current document is saved.
Characters	The amount of characters in the current document.
Lines	The amount of lines in the current document.
Words	The amount of words in the current document.
(and)	How many occurrences of this character are present. Useful for C documents and installer scripts.
Tab size	Change the tab size for this text (the global tab size is defined in the ' Editor... ' window).
Av. Length	The average length of the lines in the current document.
Av. Length	The average length of the words in the current document.
{ and }	How many occurrences of this character are present. Useful for C documents.
Tooltypes	Here you are able to change the tooltypes of the current document in the same manner as on the Workbench.
Ok	Close this window.
Next	Some information about the next document in memory.

1.21 'About' window

The 'About' window shows information about the version and the makers of the AmisED version you are using.

Gadget	Function
Continue	Close this window.
More	Show an other about window.

1.22 'Project Manager' window

The 'Project Manager' window can be used to create a list of files that in some way 'belong' together, for example the different source files of a program or all files about (whatever you want). What makes this window extra convenient is that the full path name is also saved in this list. So all you have to do is click on the file you want to open and click on the 'Open' button (a double click will have the same effect).

This window also has a pull-down window. Please be aware that the items in the 'Project' menu are meant for the list of files. The items in the 'File' menu are meant for the selected file. And the items in the 'Editor' menu have nothing to do with the list of files, but can be used to start or open a new file that is not in the list of files.

Gadget	Function
Projects	(This is the left listview) Here you can select a project.
Load	Load a previously saved project. (With filerequester).
Save	Save the selected project. (With filerequester).
New	Create a new project. (With filerequester).
Close	This closes the selected project.
Source Files	Here you see the list of files in the selected project, the type of the file and the status are also displayed.
Add	Add a file to the list of files.
Delete	Remove the selected file from the list.
Move up	Move the selected file one place up in the list.
Move down	Move the selected file one place down in the list.
Open	Open the selected file.
Close	Close the selected file.
Hide	Hide the selected file (the editor window will be closed, but the file will remain in memory).
Reveal	Reveal the selected file (Reveal = 'unhide').

1.23 'Window Manager' window

The 'Window Manager' window shows a list of all files that are loaded in the memory. The status of each file is also display, this can be either 'Opened' or 'Hidden'.

Gadget	Function
Windows	The list of files currently in memory.
Close	Close the selected file (and window).
Hide	Hide the selected file (the editor window will be closed, but the file will remain in memory).
Reveal	Reveal the selected file (Reveal = 'unhide').

1.24 'Toolbar...' window

The 'Toolbar...' window can be used to change the toolbar of the editor windows (for more information about this toolbar see Editor).

Gadget	Function
Icons	A list of all available buttons.
Toolbar	The list of buttons in the toolbar.
>>	Copy the selected icon to the toolbar list.
<<	Remove the selected button from the toolbar list.
Move up	Move the selected button one place up in the toolbar list.
Move down	Move the selected button one place down in the toolbar list.
Ok	Close this window.
Save	Save the new toolbar settings in a separate file. If you want these settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the toolbar settings from a preference file, this can either be a file previously saved with the 'Save' button or a complete settings file (only toolbar settings will be loaded).

1.25 'Editor...' window

The 'Editor...' window can be used to change the editor settings that apply to all editor windows.

Gadget	Function
Tab size	The default tab size for a new editor window. (See also Statistics).
Buffer size	The size of the text buffer for a new editor window (every time the buffer is full a new (bigger) one has to be allocated which may slow down editing). 4096 is a nice setting.
End of line	What should the cursor do at the end of the line when you press the right arrow key ? 'Stop' will stop the cursor, 'Move on' will move the cursor further to the right

	(although there is no more text there) and 'To next line' will wrap the cursor to the next line.
AutoSave	'On' will automatically save your file after some time (counting from the first change you made). 'Off' will prevent this, 'Ask' will display a requester instead of saving it directly.
AutoSave Time	The time between the first change and the moment of the AutoSave.
Max undo level	The maximal number of undo's in memory (more undo's will take more memory and slow down the system a little).
Wordwrap column	At what cursor position should AmisED 'see' the end of the line ? 0 means this function is inactive.
Codebar height	This is the amount of lines the codebar should display.
Flashing curs.	Flash the cursor instead of a cursor that is always visible.
Auto-Indent	This will move the cursor to the position under the first character (that is not a space or tab) on the previous line when you press enter.
Wordwrap	This will activate wordwrapping at the end of a line.
Cursor Up/Down Wrap	This will move the cursor back to the end of the line if the cursor is behind the end of the line when you move the cursor up or down.
Fast mode	Everything will be displayed in black and gray, this will speed up the graphical routines like scrolling and redrawing.
Create backups	AmisED now saves an extra backup file of the document you save.
Create icons	Save icons with your files (if the file doesn't have an own icon the icon 'def_text.info' from the drawer 'ENV:AMIS' will be used).
Enable numeric keypad	Now you are able to use functions like 'Page down/up' and 'Home' via the numeric keypad.
Automatic viewer switching	AmisED will not ask to switch to the appropriate viewer anymore, but will act immediately.
Insert tab as spaces	Use spaces instead of real tabs.
Toolbar	Indicates wether or not the toolbar should be ON or OFF when you open a new editor window.
Viewerbar	Indicates wether or not the viewerbar should be ON or OFF when you open a new editor window.
Statusbar	Indicates wether or not the statusbar should be ON or OFF when you open a new editor window.
Codebar	Indicates wether or not the codebar should be ON or OFF when you open a new editor window.
Ok	Close this window.
Save	Save the new editor settings in a separate file. If you want these settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the editor settings from a preference file, this can either be a file previously saved with the 'Save' button or a complete settings file (only editor settings will be loaded).

1.26 'Smart Indent...' window

The 'Smart Indent...' window can be used to define words or characters which should cause an indent. For example in a C source code you might want to indent automatically after a '{' character, this can be set in this window.

Gadget	Function
[Upper cycle]	Selects the viewer/editor module for which you want to change the smart indent settings.
[Listview]	A list of the defined strings to be recognized.
Add	Add a string to the indent list.
Remove	Remove the selected string from the indent list.
Move up	Move the selected string one place up in the indent list.
Move down	Move the selected string one place down in the indent list.
Position	Where in the line should the string be found.
Line to indent	Which line should be indented.
Direction	Should that line be indented one step forward or backward.
Allow multiple	Should AmisED indent more steps if the string was found several times.
Type	Should the indent be done with tabs or spaces.
Ok	Close this window.
Save	Save the new smart indent settings in a separate file. If you want these settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the smart indent settings from a preference file, this can either be a file previously saved with the 'Save' button or a complete settings file (only smart indent settings will be loaded).

1.27 'Current window settings...' window

The 'Current window settings...' window can be used to change some settings for only the selected editor window.

Gadget	Function
Current window	Display's the name of the window of which the settings can now be changed.
Font	With this gadget you can change the font to be used for displaying your text.
Tab size	Change the tab size.
Ok	Close this window.
Next	Select the next editor window.
Prev	Select the previous editor window.

1.28 'Screen...' window

The 'Screen...' window can be used to change the screen AmisED should use.

Gadget	Function
Screen mode...	Change the screen mode of the screen AmisED should open.
On pubscreen	The name of the public screen AmisED should open upon (No name indicates the default public screen, usually this is

	the Workbench).
Own screen	AmisED should open an own screen.
Clone Workb	Use the screen mode, palette, etc. from the workbench screen.
Ok	Close this window.
Apply	Applies the current settings to the current AmisED screen.
Save	Save the new screen settings in a separate file. If you want these settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the screen settings from a preference file, this can either be a file previously saved with the 'Save' button or a complete settings file (only screen settings will be loaded).

1.29 'Palette...' window

The 'Palette...' window can be used to change the colors used by AmisED. ←

Gadget	Function
Base colors	The colors used by AmisED for the GUI etc.
ANSI colors	The standard ANSI colors.
ANSI Shine col	The colors used by AmisED for bold text in ANSI files when Bold=Shine is selected in the Editor... settings window.
Force colors to change	AmisED will now ignore locks on the current palette colours. Programmes running on the AmisED public screen are forced to change colours too.
Upper slider	Changes the red value of the selected color.
Mid slider	Changes the green value of the selected color.
Lower slider	Changes the blue value of the selected color.
Ok	Close this window.
Save	Save the new palette settings in a separate file. If you want these settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the palette settings from a preference file, this can either be a file previously saved with the 'Save' button or a complete settings file (only palette settings will be loaded).

1.30 'Default fonts...' window

The 'Default fonts...' window can be used to tell AmisED which font is to be used for the different viewers.

Gadget	Function
Normal	Change the default font for the normal viewer.
ANSI	Change the default font for the ANSI viewer.
AmigaGuide	Change the default font for the AmigaGuide viewer.
Ok	Close this window.

Save	Save the new font settings in a separate file. If you want these settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the font settings from a preference file, this can either be a file previously saved with the 'Save' button or a complete settings file (only palette settings will be loaded).

1.31 'Keyboard...' window

The 'Keyboard...' window can be used to change the functions assigned to a key. ↔

Gadget	Function
Cycle	This cycle gadget shows the keyboard settings for the different viewers available.
Change	Change the selected key to an other key or key-combination, the functions assigned to it will stay the same.
Add	Add a key or key-combination you want to give a special function.
Remove	Remove the selected key from the list.
Edit	Edit the commands assigned to the selected key, to do this you will get an 'Edit commands...' window.
Sort	Sort the key list.
Default Keys	- Not implemented in version 1.03 -
Ok	Close this window.
Save	Save the new keyboard settings in a separate file. If you want this settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the keyboard settings from a preference file, this can either be a file previously saved with the 'Save' button, or a complete settings file (only keyboard settings will be loaded).

1.32 'Menu...' window

The 'Menu...' window can be used to change to pull-down menu of the editor windows.

Gadget	Function
Menus	A list of the menus in the pull-down menu.
Items	A list of the menu items in the selected menu.
Sub items	A list of the sub items in the selected menu item.
Add (3x)	Add a menu / menu item / sub item to the list.
Remove (3x)	Remove the selected menu / menu item / sub item from the list.
Move up (3x)	Move the selected menu / menu item / sub item one place up in the list.

Move down (3x)	Move the selected menu / menu item / sub item one place down in the list.
Edit	Edit the commands assigned to the selected menu item / sub item.
A-Key	Assign an Amiga-... short-cut to the selected menu item / sub item.
Barlabel	Change the selected menu item / sub item to a barlabel.
Subs	Indicate that the selected menu item has some sub items.
Ok	Close this window.
Save	Save the new menu settings in a separate file. If you want this settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the menu settings from a preference file, this can either be a file previously saved with the 'Save' button, or a complete settings file (only menu settings will be loaded).

1.33 'Programming' window

The 'Programming...' window can be used to change the settings for the compile, assemble, link and debug functions.

Gadget	Function
Compiler draw.	The directory AmisED should go to before executing the compile command.
Compiler comm.	The command AmisED should execute when a compile command was given.
Precompile script	An Arexx script AmisED executes when compilation is activated.
PostCompile script	An Arexx script AmisED executes after compilation is ready.
Assembler draw.	The directory AmisED should go to before executing the assemble command.
Assembler comm.	The command AmisED should execute when a assemble command was given.
Linker draw.	The directory AmisED should go to before executing the link command.
Linker comm.	The command AmisED should execute when a link command was given.
C-Debug. draw.	The directory AmisED should go to before executing the C-debug command.
C-Debug. comm.	The command AmisED should execute when a debug command was given (on a file with a name indicating that it is a C source code).
A-Debug. draw.	The directory AmisED should go to before executing the ASM-debug command.
A-Debug. comm.	The command AmisED should execute when a debug command was given (on a file with a name indicating that it is a ASM source code).
Use text draw. (5x)	Indicate that AmisED should go to the directory where the source was saved before executing the command displayed left from this gadget.
Use own proc. (5x)	Indicate that AmisED should detach this program (otherwise AmisED will wait until it returned).
Ok	Close this window.

Save	Save the new programming settings in a separate file. If you want this settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the programming settings from a preference file, this can either be a file previously saved with the 'Save' button, or a complete settings file (only programming settings will be loaded).

1.34 'Miscellaneous...' window

The 'Miscellaneous...' window can be used to change the settings for the startup script.

Gadget	Function
Startup script	The filename of the script to be execute when AmisED is booting up. (This must be an ARexx script).
CON Window	The so called 'CON' window which AmisED uses to open the shell for example.
Ok	Close this window.
Save	Save the new miscellaneous settings in a separate file. If you want this settings to be saved as the default use the 'Save settings' item from the 'Settings' menu.
Load	Load the miscellaneous settings from a preference file, this can either be a file previously saved with the 'Save' button, or a complete settings file (only miscellaneous settings will be loaded).

1.35 'Internal command...' window

The 'Internal command...' window can be used to give AmisED an internal command (for a full list see Internal Commands).

Gadget	Function
Command string	The command you want AmisED to execute.
? (the upper)	Shows a list of all available internal commands.
Syntax	Shows the arguments you can give to the selected command.
? (the lower)	Shows the help text in this file about the selected command.
Result	The result that the command returned (after you pressed 'Ok').
Ok	Execute the command.
Cancel	Close this window.

1.36 'Execute Command...' window

The 'Execute Command...' window can be used to run an AmigaDOS command from within AmisED.

Gadget	Function
Command string	The command you want to execute.
[Filereq]	Select a command from a filerequester.
Ok	Execute the selected command (it will be detached so you can continue working with AmisED while the command is being executed).
Cancel	Close this window.

1.37 'Edit commands...' window

The 'Edit commands...' window can be used to change a list of ←
command from
a key-definition, a macro or a menu definition. (For more information about
the commands you can use see
Internal Commands
).

Gadget	Function
Editing	The key, macro or menu the list of commands is assigned to.
Command list	The list of commands.
[String gadget]	The selected command.
? (The upper)	Shows a list of all available internal commands.
[Text gadget]	Shows the arguments you can give to the selected command.
? (The lower)	Shows the help text in this file about the selected command.
Add	Add a command to the command list.
Remove	Remove the selected command from the command list.
Move up	Move the selected command one place up in the command list.
Move down	Move the selected command one place down in the command list.
Optimize	Optimize the command list (this is very useful to speed up macro's, just try it and you will see).
Ok	Close this window.

1.38 Getting started

If you start AmisED as you find it after using the 'Install' script, AmisED will open with an empty editor window. Here you can type your text and edit it using the menus.

You can get this help file using the 'Help' option in the 'Misc' menu, pressing 'Amiga-H' or the help-key. You can customize AmisED using the options in the 'Settings' menu. Most special functions are easily available by menu and/or toolbar. The best way to find out what is possible is just try all the different menu options and buttons.

1.39 Save your files after a failure

Although AmisED has been tested quite a lot, there might still be some bugs in it. If you are editing a file and after a while, for some reason, AmisED isn't responding any more, you can still save your file!!

This can be done with a very special feature of AmisED. When you start AmisED, it will look for an other AmisED that might already be running. If there is an other AmisED it will be signalled to show a requester asking the user wether or not he wants to start a new copy of AmisED. If the other AmisED is not responding to this signal, the new copy will start up. Then it will ask the user if he wants to add the files that were opened in the other AmisED to the file list of the new AmisED. If so, the files will be available via the 'Window Manager' (this might not work when to much data was damaged, in that case your files are lost!). Now you can access the files by just 'revealing' them.

1.40 Credits

This program is designed and written by:

Willem Mestrom
Zevenwouden 237
3524 CR Utrecht
The Netherlands
E-mail: nuwanda@utrecht.nederlands.com

Advice and beta testing:

Vincent Groenewold (supernov)
E-mail: supernov@zeelandnet.nl or vincentg@freemail.nl
Homepage: <http://people.zeelandnet.nl/supernov/>

Robert Nienkemper
The Netherlands

Jos van Oijen
The Netherlands

Mirko Lukas
Germany

Help for the listview programming:

Leon Makkink
The Netherlands

The german translation:

Mirko Lukas
Germany

The italian translation:

Basso Alessandro
Italy

1.41 Known bugs

Bug fixes since AMIS 1.03

- o Fixed a bug in the block pasting / file inserting function, AmisED now workes a lot more stable with these functions.
- o Several small bugs are fixed, mostly just this that where 'illogical'.

Known bugs:

- o Some Mungwall hits are present.
- o AmisED doesn't work on all 68060 boards, the strange thing is that it works on some A4000's with a Cyberstorm MK-II and not on others, yet there doesn't seem to be any real difference. If you have more information about this, please contact the author. (The clue that will lead to a bug-fix will be rewarded with a free registration).

For up-to-date information have a look at the homepage of my beta-tester Vincent Groenewold (supernov): <http://people.zeelandnet.nl/supernov/>

1.42 ARexx

AmisED has a very complete ARexx port (if you want to know the name of this port, you can open the about window. Usually it will be AMISPORT.x, where x is '1' for the first AmisED that is running, '2' for the second, etc.). Almost all editing functions can be accessed through this port. The commands you can give are the same as the Internal Commands of AmisED. Since these are the commands that are used for menu, macro and key definitions you can understand these are quite powerful.

One way to add new functions to AmisED is by creating an ARexx script and assign it to a key or menu with the

Keyboard
or
Menu
settings using the

Rexx
command.

Before you start with creating ARexx scripts it would be wise to take a good look at the commands that are available, this might save you a lot of work trying to create a function that is already supported!

1.43 Internal Commands

AmisED has got quite a few Internal Commands available, which can be used for macro's, special key functions, menu functions and from the Internal Command window. The Internal Commands are at the same time the ARexx commands (See ARexx).

Groups of commands:

Cursor~control~~~~~
Everything that controls cursor movement.

Editing~~~~~
Commands to edit your text.

Folding
Fold and unfold your text.

Blocks~~~~~
Copy, cut, paste, etc.

Search~~~~~
Search (& replace) commands.

Programming ~~~~
Everything for Compiling, Assembling, etc.

File&~projects~~~~
Loading, saving, etc.

Messages~~~~~
Say something to the user.

Info windows~~~~~
About, statistics, etc.

Editor windows
Moving windows around.

Commands~~~~~
DOS, REXX, etc.

Editor options ~
FastMode, OverWrite, Tab size, etc.

Settings
Loading, saving & editing settings.

Other~~~~~
Everything else.

Complete lists:

Full~alphabetical~list~~~~~

Full~list~arranged~by~subject~

1.44 Alphabetical list of the Internal Commands

All Internal Commands in alphabetical order:

ArrangeWindows

Assemble

AutoIndent

AutoSave

Backspace

BackUps

Block

BlockLayout

BlockMode

Codebar

Compile

CopyBlock

CreateIcons

CursDown

CursMode

CursLeft

CursorWrap

CursRight

CursUp

CutBlock

Debug

Delete

EditorSettings

EOIMode

ExecuteCommand
FastMode
File
Find
FindFuncions
Fold
FontSettings
GetASCII
GetLine
GetWord
Goto
Graphics
Help
Iconify
Insert
InternalCommand
KeyboardSettings
Line
Link
Macro
 Make
Message
MenuSettings
MiscSettings
OverWrite
ParserError
PasteBlock
Print
Project

ProgramSettings

Query

Quit

ResultToVar

Redo

Replace

Request

Rexx

Run

ScreenSettings

SelectWindow

SetBookmark

SetTab

Settings

Shell

ShowAbout

ShowStatistics

ShowWindowManager

Statusbar

Toolbar

ToolbarSettings

Undo

UnFold

Viewer

Viewerbar

Window

WordWrap

WriteProtect

1.45 Available Internal Commands

All Internal Commands arranged by subject.

Cursor control:

CursUp

CursDown

CursLeft

CursRight

Goto

SetBookmark

Editing:

Backspace

Delete

Insert

Line

Macro

Folding:

Fold

UnFold

Blocks:

Block

CutBlock

CopyBlock

PasteBlock

BlockLayout

BlockMode

Search:

Find

Replace

FindFunctions

Programming:

Compile

Assemble

Link

Run

Debug

ParserError

File & projects:

File

Print

Project

Messages:

Request

Message

Info Windows:

ShowStatistics

ShowWindowManager

ShowAbout

Editor windows:

ArrangeWindows

SelectWindow

Window

Commands:

ExecuteCommand

InternalCommand

Shell

Rexx

Editor options:

CursMode

EOLMode

CursorWrap

WriteProtect

SetTab
BackUps
AutoSave
AutoIndent
CreateIcons
FastMode
OverWrite
WordWrap
Toolbar
Viewerbar
Statusbar
Codebar
Viewer
Settings:
ToolbarSettings
EditorSettings
 SmartIndentSettings
KeyboardSettings
MenuSettings
ScreenSettings
FontSettings
ProgramSettings
MiscSettings
Settings
Other:
Graphics
Query
GetASCII
GetWord
GetLine

ResultToVar

Help

Undo

Redo

Iconify

Quit

1.46 Internal command: CursUp

CursUp

Function: Move the cursor upward.
Syntax: LINES/N,PAGE/S, TOP/S, SELECT/S, SCROLL/S
Result: 0 - Done (if possible).
10 - No editor window open / selected.

Options: LINES: Move the cursor LINES lines up.
PAGE: Move the cursor 1 page up.
TOP: Move the cursor to the top.
SELECT: Move the cursor extend the block to the new cursor position.
SCROLL: Keep the cursor on the same line, but scroll the whole window 1 line down.

1.47 Internal command: CursDown

CursDown

Function: Move the cursor downward.
Syntax: LINES/N,PAGE/S,BOTTOM/S, SELECT/S, SCROLL/S
Result: 0 - Done (if possible).
10 - No editor window open / selected.

Options: LINES: Move the cursor LINES lines down.
PAGE: Move the cursor 1 page down.
BOTTOM: Move the cursor to the bottom.
SELECT: Move the cursor extend the block to the new cursor position.
SCROLL: Keep the cursor on the same line, but scroll the whole window 1 line down.

1.48 Internal command: CursLeft

CursLeft

Function: Move the cursor to the left.
Syntax: CHARS/N,WORD/S,SOL/S,SELECT/S
Result: 0 - Done (if possible).
10 - No editor window open / selected.

Options: CHARS: Move the cursor CHARS characters to the left.
WORD: Move the cursor 1 word to the left.
SOL: Move the cursor to the Start Of the Line.
SELECT: Move the cursor extend the block to the new cursor position.

1.49 Internal command: CursRight

CursRight

Function: Move the cursor to the right.
Syntax: CHARS/N,WORD/S,EOL/S,SELECT/S
Result: 0 - Done (if possible).
10 - No editor window open / selected.

Options: CHARS: Move the cursor CHARS characters to the right.
WORD: Move the cursor 1 word to the right.
EOL: Move the cursor to the End Of the Line.
SELECT: Move the cursor extend the block to the new cursor position.

1.50 Internal command: CursMode

CursMode

Function: Switch between normal cursor and the cursor in the codebar. When the cursor in the codebar is active you can edit any text as if you were using the normal viewer/editor module even when that's not the case. This is very handy for editing unsupported HTML tags and things like that.

Syntax: NORMAL/S,CODEBAR/S,TOGGLE/S
Result: 0 - Ok.
10 - No editor window open / selected.

Options: NORMAL: Activate normal cursor.
CODEBAR: Activate cursor in the codebar.
TOGGLE: Toggle between the normal cursor and the cursor in the codebar.

1.51 Internal command: WordWrap

WordWrap

Function: Switch between the available wordwrap modes.
Syntax: ON/S, OFF/S, DYNAMIC/S, TOGGLE/S
Result: 0 - Ok.
10 - No editor window open / selected.

Options: ON: Normal wordwrap on.
OFF: All wordwrap off.
DYNAMIC: Dynamic wordwrap on.
TOGGLE: Toggle between normal wordwrap, dynamic wordwrap
and no wordwrap.

1.52 Internal command: Goto

Goto

Function: Move the cursor to specific places in the document.
Syntax: LINE/N, COLUMN/N, POS/N, LASTCHANGE/S, BOOKMARK/N, INTOP/S,
BLOCKSTART/S, BLOCKEND/S
Result: 0 - Cursor moved.
10 - No editor window open / selected.

Options: LINE: Move the cursor to line LINE.
COLUMN: Move the cursor to column COLUMN.
POS: Move the cursor POS characters from the start.
LASTCHANGE: - not implemented in version 1.03 -
BOOKMARK: Move the cursor to bookmark BOOKMARK.
INTOP: Make sure the line the cursor will be on is in top
of the editor window.
BLOCKSTART: Move the cursor to the start of the selected
block.
BLOCKEND: Move the cursor to the end of the selected
block.

1.53 Internal command: SetBookmark

SetBookmark

Function: Set a bookmark.
Syntax: BOOKMARK/N
Result: 0 - Done.
10 - No editor window open / selected.

Options: BOOKMARK: Which bookmark is to be set (1-10).

See also: Goto

1.54 Internal command: Backspace

Backspace

Function: Remove text on the left side of the cursor.
Syntax: CHARS/N,WORD/S,SOL/S
Result: 0 - Done (if possible).
10 - No editor window open / selected.

Options: CHARS: Remove CHARS characters.
WORD: Remove 1 word.
SOL: Remove everything left of the cursor.

1.55 Internal command: Delete

Delete

Function: Remove text on the right side of the cursor.
Syntax: CHARS/N,WORD/S,EOL/S
Result: 0 - Done (if possible).
10 - No editor window open / selected.

Options: CHARS: Remove CHARS characters.
WORD: Remove 1.
EOL: Remove everything right of the cursor.

1.56 Internal command: Line

Line

Function: Edit a document on a level of lines.
Syntax: INSERT/S,REMOVE/S,SPLIT/S,JOIN/S,SWAP/S,DOUBLE/S
Result: 0 - Ok.
10 - No editor window open / selected.

Options: INSERT: Insert a line before the line the cursor is on.
REMOVE: Remove the line the cursor is on.
SPLIT: Same as a return.
JOIN: Join the next line with the current line.
SWAP: Swap the current line with the previous one.
DOUBLE: Insert a line after the current one and copy the current line to it.

1.57 Internal command: Insert

Insert

Function: Insert a string, file or character at the cursor position.
Syntax: TEXT/K,FILE/K,ASCII/N,NOMOVE/S

Result: 0 - Ok.
10 - No editor window open / selected.

Options: TEXT: Insert a string (example: Insert TEXT="Hello", this will insert the string 'Hello'). (For some more information about how to enter a string please see: AmigaDOS arguments).

FILE: Insert a file.
ASCII: Insert a character with this ASCII number.
NOMOVE: If this option is selected the cursor won't move, otherwise the cursor will move as if the text was typed on the keyboard.

1.58 Internal command: File

File

Function: Open, load or save a file, change it's name, open a new one and some other useful things.

Syntax: NAME/K,NEW/S,OPEN/S,LOAD/S,SAVE/S,CLOSE/S,CLEAR/S,REVERT/S,ADDTOPROJECT/S,ASK/S

Result: 0 - Done.
10 - Failed or
No editor window opened / selected.

Options: NAME: The name of the file you want to OPEN, LOAD or create with NEW. The name that should be given to the current file. The name of the file that should be saved (This will change the name of file you are editing).

NEW: Open a new editor window.
OPEN: Open a new editor window and load the file [NAME] in it.
LOAD: Load the file [NAME] in the current editor window.
SAVE: Save the text in the current editor window in the file [NAME].
CLOSE: Close the current editor window.
CLEAR: Clear the text in the current editor window.
REVERT: Revert the text in the current editor window to the last saved version.
ADDTOPROJECT: Add the file in the current editor window to the current window.
ASK: Don't use the file [NAME] but ask for a file using a filerequester.

1.59 Internal command: EOLMode

EOLMode

Function: Define what the cursor should do when it is on the end of the line and a 'CursRight' command is given (or when the right arrow is pressed on the keyboard).

Syntax: WRAP/S,STOP/S,IGNORE/S

Result: 0 - Ok

Options:

- WRAP: The cursor will move to the start of the next line, if you give a 'CursLeft' command at the start of a line the cursor will move to the end of the previous line.
- STOP: The cursor will never jump to the next or previous line when a 'CursRight' or 'CursLeft' command is given.
- IGNORE: The cursor will move on more step to the right, 'CursLeft' will do the same as in WRAP mode.

1.60 Internal command: CursorWrap

CursorWrap

Function: Define what the cursor should do when a 'CursUp' or 'CursDown' command is given which would move the cursor to a place after the end of the previous or next line.

Syntax: ON/S,OFF/S

Result: 0 - Ok

Options:

- ON: The cursor will now move to the last character of the next or previous line.
- OFF: The cursor will only move up and down regardless of where that would be.

See also:

- EOLMode

1.61 Internal command: WriteProtect

WriteProtect

Function: This function allows you to lock a document and prevent it from being edited (useful for include files for example).

Syntax: ON/S,OFF/S,TOGGLE/S

Result: 0 - Ok.
10 - No editor window open / selected.

Options:

- ON: Lock the current document.
- OFF: Unlock the current document.
- TOGGLE: Switch between locked and unlocked mode.

1.62 Internal command: SetTab

SetTab

Function: Change the tab size used in the current editor window.
Syntax: TAB/N/A
Result: 0 - Ok.
10 - No editor window opened / selected.
Options: TAB: The size of the tabs.

1.63 Internal command: BackUps

BackUps

Function: Switch the making of backups on or off.
Syntax: ON/S,OFF/S
Result: 0 - Ok.
Options: ON: Backups will be made.
OFF: No backups will be made.

1.64 Internal command: AutoSave

AutoSave

Function: Switch AutoSave mode on or of, when switched on your file will automatically be saved once in a while. You can also let AmisED ask you wether or not you want to save your file.
Syntax: TIME/N,ON/S,OFF/S,ASK/S
Result: 0 - Ok.
Options: TIME: Set the time that has to elapse before Amised will save your file.
ON: Switch AutoSave mode on.
OFF: Switch AutoSave mode off.
ASK: Let Amised ask you wether or not you want to save your file.

1.65 Internal command: AutoIndent

AutoIndent

Function: Switch AutoIndent mode on or off. When switched on and you insert a new line (with return) Amised will set the cursor under the first text on the previous line.
Syntax: ON/S,OFF/S
Result: 0 - Ok.
Options: ON: Switch AutoIndent on.
OFF: Switch AutoIndent off.

1.66 Internal command: CreateIcons

CreateIcons

Function: Switch the creating of icons on or off. When switch on AmisED will save an icon and tooltypes with your file. If your file already has an icon, this icon will be used. Otherwise the icon "def_text" in the "ENV:AMIS" directory will be used.

Syntax: ON/S,OFF/S

Result: 0 - Ok.

Options: ON: Switch CreateIcons on.
OFF: Switch CreateIcons off.

1.67 Internal command: FastMode

FastMode

Function: Switch FastMode on or off. When switched on everything will be displayed in one color, doing this scrolling will be much faster on 'slower' machines.

When switched on 'FAST' will be displayed in the statusbar.

Syntax: ON/S,OFF/S

Result: 0 - Ok.

Options: ON: Switch FastMode on.
OFF: Switch FastMode off.
TOGGLE: Toggle between FastMode on and FastMode off.

1.68 Internal command: OverWrite

OverWrite

Function: Switch OverWrite on or off. When switched on AmisED will replace the character the cursor is on when you press a key. Otherwise an character will be inserted before the character were the cursor is on.

When switched on 'OVER' will be displayed in the statusbar.

Syntax: ON/S,OFF/S,TOGGLE/S

Result: 0 - Ok.

Options: ON: Switch OverWrite on.
OFF: Switch OverWrite off.
TOGGLE: Toggle between OverWrite on and OverWrite off.

1.69 Internal command: Query

Query

Function: Get information about the current situation.
Syntax: LINE/S, LINES/S, CHAR/S, LINELENGTH/S, POS/S, CHARS/S, NAME/S, PATH/S, FLAG/N
Result: 0 - Ok.
10 - No editor window opened / selected.

Options:

- LINE: Return the line number.
- LINES: Return the number of lines.
- CHAR: Return the character number of the cursor relative to the start of the line (0 = first of the line).
- LINELENGTH: Return the length of the line.
- POS: Return the character number of the cursor relative to the start of the text (0 = first of the text).
- CHARS: Return the number of characters.
- NAME: Return the file name of the text.
- PATH: Return the path were this file can be found.
- FLAG: Return the status of an internal flag. Currently the following values are supported:
 1. Overwrite ON/OFF (1/0)

1.70 Internal command: GetWord

GetWord

Function: Return the word the cursor is on in the second result variable (ARexx only).
Syntax: -
Result: 0 - Ok.
10 - No editor window opened / selected.

Options: -

1.71 Internal command: GetLine

GetLine

Function: Return the line the cursor is on in the second result variable (ARexx only).
Syntax: -
Result: 0 - Ok.
10 - No editor window opened / selected.

Options: -

1.72 Internal command: GetASCII

GetASCII

Function: Return the ASCII code of the character the cursor is on in the second result variable (ARexx only).
Syntax: -
Result: 0 - Ok.
10 - No editor window opened / selected.
Options: -

1.73 Internal command: Toolbar

Toolbar

Function: Switch the toolbar of the selected editor window on or off.
Syntax: ON/S,OFF/S,TOGGLE/S
Result: 0 - Ok.
10 - No editor window opened / selected.
Options: ON: Switch the toolbar on.
OFF: Switch the toolbar off.
TOGGLE: Toggle between toolbar on and off.

1.74 Internal command: Codebar

Codebar

Function: Switch the codebar of the selected editor window on or off.
Syntax: ON/S,OFF/S,TOGGLE/S
Result: 0 - Ok.
10 - No editor window opened / selected.
Options: ON: Switch the codebar on.
OFF: Switch the codebar off.
TOGGLE: Toggle between codebar on and off.

1.75 Internal command: Statusbar

Statusbar

Function: Switch the statusbar of the selected editor window on or off.
Syntax: ON/S,OFF/S,TOGGLE/S
Result: 0 - Ok.
10 - No editor window opened / selected.
Options: ON: Switch the statusbar on.
OFF: Switch the statusbar off.
TOGGLE: Toggle between statusbar on and off.

1.76 Internal command: Viewerbar

Viewerbar

Function: Switch the viewerbar of the selected editor window on or off.

Syntax: ON/S, OFF/S, TOGGLE/S

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: ON: Switch the viewerbar on.
OFF: Switch the viewerbar off.
TOGGLE: Toggle between viewerbar on and off.

1.77 Internal command: Viewer

Viewer

Function: Change the viewer for the selected editor window.

Syntax: NORMAL/S, ANSI/S, AMIGAGUIDE/S

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: NORMAL: Switch to normal viewing mode.
ANSI: Switch to ANSI viewing mode.
AMIGAGUIDE: Switch to AmigaGuide viewing mode.

1.78 Internal command: Request

Request

Function: Display a requester with info and or a question for the user. The number of the button the user selected will be passed in the second result variable. If a value is requested, this value will be passed along with this text (separated by an '|' character, for example: Result2='243|1' -> number: 243, button: 1).

Syntax: BODY, BUTTON/K, LONG/S, VALUE/N, NOCENTRE/S, VAR/K, CANCEL/N, RESULTNUMBER/S, LOCALE/N, TITLE/K

Result: 0 - Ok.
10 - Failed (for any reason).

Options: BODY: Here you can give the text you want to be displayed in the requester. You can separate lines by an '|' character.
BUTTON: Here you can give the text you want in your buttons. Different buttons can be separated by an '|' character. An '_' character will make the letter behind this character a short-cut for this button. If the button name starts with an '␣' character, this button will be displayed in bold text and can be selected with the return key.

LONG: Indicates that you request a number from the user. This will be done by adding a integer gadget to the requester.

VALUE: Here you can give the starting value of the integer gadget.

NOCENTRE: By default all text in the requester is centred. If you select this option all text will be left-aligned.

VAR: Replaces a %s in the BODY text.

CANCEL: If you select this option, the macro or key defenition you use this in will be canceled if the user selects the button with the number you pass in this option.

RESULTNUMBER: If you are expecting a value to be passed (you select the LONG option) you can use this option to get only this value and not the button number.

LOCALE: A locale string number, which will be used instead of the text givin in BODY when available.

TITLE: The title of the requester.

See also:

Message

1.79 Internal command: Message

Message

Function: Display a message for the user in the title bar of the editor window.

Syntax: TEXT/A, TIME/N

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: TEXT: The text to be displayed in the title bar.
TIME: The time the message should stay in the bar, after this time the message will disappear again. (With a value of nul the message will stay until it is overwritten by an other message).

See also:

Request

1.80 Internal command: Find

Find

Function: Find a string in the text.

Syntax: STRING/K, NEXT/S, PREV/S, FIRST/S, CASE/S, NOCASE/S, WORDS/S, NOWORDS, ASK/S

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: STRING: Here you can give the string you want to find.
NEXT: Find forward from the cursor.
PREV: Find backward from the cursor.
FIRST: Find forward from the start of the text.
CASE: Case sensitive: 'a' is not the same as 'A'.
NOCASE: Case insensitive: 'a' is the same as 'A'.
WORDS: Look for whole words: 'testing' is not good if you are looking for 'test'.
NOWORDS: A part of a word will do as well.
ASK: Show the 'Find...' window.

See also:

Replace

1.81 Internal command: Replace

Replace

Function: Replace the string you found with 'Find'.
Syntax: REPLACE/K,NEXT/S,PREV/S,ALL/S,ASK/S
Result: 0 - Ok.
10 - No editor window opened / selected.

Options: REPLACE: The string you want to be placed on the place of the found string.
NEXT: Find the next occurrence of the string (defined with 'Find') after this one has been replaced.
PREV: Find the previous occurrence of the string after this one has been replaced.
ALL: Replace all found occurrences of the string. (Starting from the beginning of the file).
ASK: Show the 'Find...' window (where you can also find the replace options).

See also:

Find

1.82 Internal command: FindFunctions

FindFunctions

Function: Show the 'Find functions...' window.
Syntax: ASK/S
Result: 0 - Ok.

Options: ASK: Show the 'Find functions...' window.

1.83 Internal command: Block

Block

Function: Select / unselect a block, save it, remove it, etc.

Syntax: START/S,END/S,WORD/S,LINE/S,ALL/S,FROMCURS/S,UNSELECT/S,
SAVEAS/K,SAVEREQ/S,ERASE/S,WHOLELINES/S

Result: 0 - Ok.
10 - No editor window opened / selected.
20 - WORD -> no word under cursor.
- WHOLELINES -> no block selected.

Options: START: Set the start position of the block on the current
cursor position.
END: Set the end position of the block on the current
cursor position.
WORD: Select the word under the cursor.
LINE: Select the line the cursor is on.
ALL: Select the whole text.
FROMCURS: Select all text beginning at the cursor
position. (Same as 'Block ALL', followed by
'Block START')

UNSELECT: Unselect the block.

SAVEAS: Save the block in the file with the name
given after this keyword.

SAVEREQ: Save the block using a filerequester.

ERASE: Erase the block from the text (not copied to
the clipboard!).

WHOLELINES: Enlarge the selected area in such a way that
all lines in the block are completely
selected.

See also:

CutBlock
,
CopyBlock
,
PasteBlock

1.84 Internal command: BlockLayout

BlockLayout

Function: This command can add or remove spaces at the beginning of
all lines in a block in such a way that the text will be
centered, left aligned or right aligned.

Syntax: CENTER/S,LEFT/S,RIGHT/S,JUSTIFY/S

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: CENTER: This will center the text on the lines in the block.
LEFT: This will left align the text.
RIGHT: This will right align the text.

JUSTIFY:This will outline the text.

1.85 Internal command: BlockMode

BlockMode

Function: With this command you can switch between normal (horizontal) blocks and vertical blocks. The best way to see the difference is try it. I don't really know how to explain this.

Syntax: HORIZONTAL/S,VERTICAL/S

Result: 0 - Ok.
10 - No editor window opened.

Options: HORIZONTAL: Switch to horizontal blocks.
VERTICAL: Swicth to vertical blocks.

1.86 Internal command: CutBlock

CutBlock

Function: Remove the selected block from the text and copy it to the clipboard.

Syntax: CLIP/N

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: CLIP: The number of the clip you want to copy the block to.

See also:

- Block
- ,
- CopyBlock
- ,
- PasteBlock

1.87 Internal command: CopyBlock

CopyBlock

Function: Copy the selected block to the clipboard.

Syntax: CLIP/N

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: CLIP: The number of the clip you want to copy the block to.

See also:

```
Block
,
CutBlock
,
PasteBlock
```

1.88 Internal command: PasteBlock

PasteBlock

Function: Insert text from the clipboard in the text.
Syntax: CLIP/N
Result: 0 - Ok.
10 - No editor window opened / selected.

Options: CLIP: The number of the clip you want to copy the text
from.

See also:

```
Block
,
CutBlock
,
CopyBlock
```

1.89 Internal command: Undo

Undo

Function: Undo the last action.
Syntax: SHOWLIST/S
Result: 0 - Ok.
10 - No editor window opened / selected.

Options: SHOWLIST: Show a list with all undo's in memory.
(No option: normal undo)

1.90 Internal command: Redo

Redo

Function: Redo the action you undid with 'Undo'.
Syntax: -
Result: 0 - Ok.
10 - No editor window opened / selected.

Options: -

1.91 Internal command: Macro

Macro

Function: Record or play a macro.
Syntax: NAME/K,PLAY/S,RECORD/S,STOPREC/S,ASK/S
Result: 0 - Ok.

Options: NAME: Here you can give the name of the macro that should be played / recorded.
PLAY: Play the selected macro.
RECORD: Record a new macro (with name NAME or if not given: 'New Macro').
STOPREC: Stop recording.
ASK: Show the 'Macro...' window.

1.92 Internal command: Select Window

SelectWindow

Function: Select an editor window.
Syntax: NAME/K,NEXT/S,PREV/S,NUMBER/N
Result: 0 - Window selected.
5 - No window selected.
10 - There are no windows at all.

Options: NAME: The window with this name will be selected (if there is no window with the given name the function will return 5 as result and no window will be selected).
NEXT: The next window will be selected.
PREV: The previous window will be selected.
NUMBER: The window with this number will be selected (if there is no window with that number the function will return 5 as result and no window will be selected).

See also:

Window
,
ArrangeWindows

1.93 Internal command: Window

Window

Function: Change window size and position, lock or unlock a window or hide or reveal a window.
Syntax: X/N,Y/N,WIDTH/N,HEIGHT/N,LOCK/S,UNLOCK/S,HIDE/S,REVEAL/S
Result: 0 - Ok.
10 - No editor window opened / selected.

Options: X: Set the x position of the window.
 Y: Set the y position of the window.
 WIDTH: Set the width of the window.
 HEIGHT: Set the height of the window.
 LOCK: Lock the window from user input (and busy pointer).
 UNLOCK: Make user input possible again.
 HIDE: Hide the current window.
 REVEAL: Make a window visible again.

See also:

SelectWindow
 ,
 ArrangeWindows

1.94 Internal command: ArrangeWindows

ArrangeWindows

Function: Change the position and size of the windows in an orderly way.

Syntax: HORIZONTAL/S, VERTICAL/S, STACKED/S, DIAGONAL/S, FULLSIZE/S

Result: 0 - Ok.
 10 - No editor window opened / selected.

Options: HORIZONTAL: All windows over the full width, equally divided over the height of the screen.
 VERTICAL: All windows over the full height, equally divided over the width of the screen.
 STACKED: The first window fullsize, the next windows every time a little shorter. All title bars will be visible.
 DIAGONAL: Same as stacked, but now also every time a little smaller. Now the depth gadget of every window will always be visible.
 FULLSIZE: All windows over the full screen size.

See also:

SelectWindow
 ,
 Window

1.95 Internal command: Compile

Compile

Function: Compile the file (it has to be a C source code and thus the file name should end with .c, .cpp, .cxx, .C, .CPP or .CXX).

Syntax: -

Result: 0 - Ok.
 10 - No editor window opened / selected.

Options: -

See also:

```
Assemble
/
Link
/
Debug
/
Run
/
ParserError
```

1.96 Internal command: Assemble

Assemble

Function: Assemble the file (it has to be a ASM source code and thus the file name should end with .a, .s, .asm, .A, .S or .ASM).

Syntax: -

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: -

See also:

```
Compile
/
Link
/
Debug
/
Run
/
ParserError
```

1.97 Internal command: Link

Link

Function: Link the file. It has to be a source code file. If so the file name without the extension (.a or .c, etc.) will be inserted in the link command string (see program settings).

Syntax: -

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: -

See also:

```
Compile
/
Assemble
/
Debug
/
Run
/
ParserError
```

1.98 Internal command: Debug

Debug

Function: Debug the file. It has to be a source code file. If so the file name without the extension (.a or .c, etc.) will be inserted in the debug command string (see program settings). Which debug command string (Asm or C) depends on the file extension.

Syntax: -

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: -

See also:

```
Compile
/
Assemble
/
Link
/
Run
/
ParserError
```

1.99 Internal command: Run

Run

Function: Run the file. It has to be a source code file. If so the file name without the extension (.a or .c, etc.) will be send to the system as a dos command.

Syntax: -

Result: 0 - Ok.
10 - No editor window opened / selected.

Options: -

See also:

```
Compile
,
Assemble
,
Link
,
Debug
,
ParserError
```

1.100 Internal command: ParserError

ParserError

Function: Go to a line where an error was reported by the compiler or assembler.

Syntax: NEXT/S, PREVS/, FIRST/S, LAST/S

Result: 0 - Ok.

Options: NEXT: Go to the line with the next error.
PREV: Go to the line with the previous error.
FIRST: Go to the line with the first error.
LAST: Go to the line with the last error.

See also:

```
Compile
,
Assemble
,
Link
,
Debug
,
Run
```

1.101 Internal command: Project

Project

Function: Open or save a project, start a new one, close the active project or show the 'Project Manager' window.

Syntax: NAME/K, NEW/S, OPEN/S, SAVE/S, CLOSE/S, MANAGER/S

Result: 0 - Ok.

Options: NAME: The file name of the project you want to open, create or the file name were you want to save the selected project to. ('.prj' will be added).
NEW: Start a new project with name NAME, if not given you will get a filerequester to enter a name.

OPEN: Open the project with file name NAME, if not given a filerequester will be used.
SAVE: Save the selected project in file NAME, if not given a filerequester will be used.
CLOSE: Close the selected project.
MANAGER: Show the 'Project Manager' window.

1.102 Internal command: ExecuteCommand

ExecuteCommand

Function: Execute a DOS command.
Syntax: COMMAND/K,ASK/S
Result: 0 - Ok.
Options: COMMAND: The DOS command you want to execute
(Example: "Dir C:").
ASK: Show the 'Execute command...' window.

1.103 Internal command: InternalCommand

InternalCommand

Function: Open the 'Internal command...' window.
Syntax: ASK/S
Result: 0 - Ok.
Options: ASK: Open the 'Internal command...' window.

1.104 Internal command: Shell

Shell

Function: Open a new shell window (on the AmisED screen).
Syntax: -
Result: 0 - Ok.
Options: -

1.105 Internal command: REXX

REXX

Function: Execute an AREXX script.
Syntax: SCRIPT/A, WAIT/S, FAST/S
Result: 0 - Ok.

SAVE: Save the settings with name NAME, if not given you will get a filerequester.
ASK: Show the 'Toolbar...' window.

1.110 Internal command: EditorSettings

EditorSettings

Function: Load or save the editor settings and/or show the 'Editor...' window.
Syntax: NAME/K,LOAD/S,SAVE/S,ASK/S
Result: 0 - Ok.

Options:

- NAME: The name of the preference file to load or save.
- LOAD: Load the settings with name NAME, if not given you will get a filerequester.
- SAVE: Save the settings with name NAME, if not given you will get a filerequester.
- ASK: Show the 'Editor...' window.

1.111 Internal command: KeyboardSettings

KeyboardSettings

Function: Load or save the keyboard settings and/or show the 'Keyboard...' window.
Syntax: NAME/K,LOAD/S,SAVE/S,ASK/S
Result: 0 - Ok.

Options:

- NAME: The name of the preference file to load or save.
- LOAD: Load the settings with name NAME, if not given you will get a filerequester.
- SAVE: Save the settings with name NAME, if not given you will get a filerequester.
- ASK: Show the 'Keyboard...' window.

1.112 Internal command: MenuSettings

MenuSettings

Function: Load or save the menu settings and/or show the 'Menu...' window.
Syntax: NAME/K,LOAD/S,SAVE/S,ASK/S,APPEND/S
Result: 0 - Ok.

Options:

- NAME: The name of the preference file to load or save.
- LOAD: Load the settings with name NAME, if not given you will get a filerequester.
- SAVE: Save the settings with name NAME, if not given you will get a filerequester.

ASK: Show the 'Menu...' window.
APPEND: When this option is given the menu will be added to the menu already in memory.

1.113 Internal command: FontSettings

FontSettings

Function: Load or save the font settings and/or show the ASL Font requester.
Syntax: NAME/K,LOAD/S,SAVE/S,ASK/S
Result: 0 - Ok.

Options:

- NAME: The name of the preference file to load or save.
- LOAD: Load the settings with name NAME, if not given you will get a filerequester.
- SAVE: Save the settings with name NAME, if not given you will get a filerequester.
- ASK: Show the ASL Font requester.

1.114 Internal command: ScreenSettings

ScreenSettings

Function: Load or save the screen settings and/or show the ASL Screen requester.
Syntax: NAME/K,LOAD/S,SAVE/S,ASK/S
Result: 0 - Ok.

Options:

- NAME: The name of the preference file to load or save.
- LOAD: Load the settings with name NAME, if not given you will get a filerequester.
- SAVE: Save the settings with name NAME, if not given you will get a filerequester.
- ASK: Show the ASL Screen requester.

1.115 Internal command: ProgramSettings

ProgramSettings

Function: Load or save the programming settings and/or show the 'Programming...' window.
Syntax: NAME/K,LOAD/S,SAVE/S,ASK/S
Result: 0 - Ok.

Options:

- NAME: The name of the preference file to load or save.
- LOAD: Load the settings with name NAME, if not given you will get a filerequester.
- SAVE: Save the settings with name NAME, if not given you will get a filerequester.

ASK: Show the 'Programming...' window.

1.116 Internal command: SmartIndentSettings

SmartIndentSettings

Function: Load or save the smart indent settings and/or show the 'Smart Indent...' window.
Syntax: NAME/K,LOAD/S,SAVE/S,ASK/S
Result: 0 - Ok.

Options: NAME: The name of the preference file to load or save.
LOAD: Load the settings with name NAME, if not given you will get a filerequester.
SAVE: Save the settings with name NAME, if not given you will get a filerequester.
ASK: Show the 'Smart Indent...' window.

1.117 Internal command: MiscSettings

MiscSettings

Function: Load or save the programming settings and/or show the 'Miscellaneous...' window.
Syntax: NAME/K,LOAD/S,SAVE/S,ASK/S
Result: 0 - Ok.

Options: NAME: The name of the preference file to load or save.
LOAD: Load the settings with name NAME, if not given you will get a filerequester.
SAVE: Save the settings with name NAME, if not given you will get a filerequester.
ASK: Show the 'Miscellaneous...' window.

1.118 Internal command: Settings

Settings

Function: Load or save all settings.
Syntax: NAME/K,LOAD/S,SAVE/S
Result: 0 - Ok.

Options: NAME: The name of the preference file to load or save.
LOAD: Load the settings with name NAME, if not given you will get a filerequester.
SAVE: Save the settings with name NAME, if not given you will get a filerequester.

1.119 Internal command: Iconify

Iconify

Function: This function will iconify AmisED, all windows and the AmisED screen will be closed. You can reopen AmisED with a double click on the AppIcon on the workbench. (Warning: If you use too much memory when AmisED is iconified you might not be able to reopen the AmisED environment!).

Syntax: -

Result: 0 - Ok.
10 - Failed.

Options: -

1.120 Internal command: Quit

Quit

Function: Quit AmisED.

Syntax: FORCE/S

Result: 0 - Ok.

Options: FORCE: Quit without the usual quit requester.

1.121 Internal command: ResultToVar

ResultToVar

Function: This function allows you to use the result of a previous internal command as a variable for the next internal command.

Syntax: VAR/N

Result: Always the same as the previous command, this has no special meaning.

Options: VAR: Here you must give the number of the variable of the next command where the result should be used.

Example: 'Request BODY="Enter a number" LONG'
The result will be the number the user entered.
'ResultToVar VAR=3'
3 Because 'ASCII=...' is the 3th option of the 'Insert' command.
'Insert'
This will insert a character with the ASCII code the user entered in the requester!

1.122 Internal command: Help

Help

Function: Show the on-line help guide (this file).
Syntax: -
Result: 0 - Ok.
Options: -

1.123 Internal command: Graphics

Graphics

Function: Switch off graphical routines. This is very useful for macro's, they will be executed much faster because not all steps are displayed. You MUST turn it back on when you are done, otherwise the user won't see any changes.
Syntax: ON/S,OFF/S
Result: 0 - Ok.
Options: ON: Switch graphical routines on.
OFF: Switch graphical routines off.

1.124 Internal Command: Fold

Fold

Function: This function is used to 'fold' a part of a text. This means that some lines are made invisible to make the structure of the text more clear. For example in AmigaGuide mode this function will hide all lines of a node (page), it only leaves the title visible.
Syntax: TOGGLE/S,ALL/S
Result: 0 - Ok.
10 - Viewer doesn't support folding.
Options: TOGGLE: Toggle between folded and unfolded text.
ALL: Fold (or toggle) all parts of the text that can be folded.

1.125 Internal Command: UnFold

UnFold

Function: This function is used to 'unfold' a part of a text. For more information about folding see Fold
.
Syntax: ALL/S
Result: 0 - Ok.

10 - Viewer doesn't support folding.

Options: ALL: UnFold all parts of the text that are currently folded.

1.126 Internal Command: Print

Print

Function: Print the current tekst.
Syntax: -
Results: 0 - Ok.

1.127 AmigaDOS Arguments

All internal commands work with the standard AmigaDOS template functions, this means that arguments should be given in the same way as for normal DOS commands.

There are several argument types. In the template string (syntax string) the type of the argument is indicated with these characters:

\N This is a number.
\K This is a text.
\S This is a switch.

\A This argument must always be given.

For strings there are some extra things you should know. If you want to insert spaces in a string you should quote the string with " characters. For example: TEXT="A test" (A test). If you want to use " characters in the string you should give *" instead. For example: TEXT="*"A test*"" ("A test"). If you want to use a * character in the string you should give ** instead. For example: TEXT="** A test **" (* A test *).

1.128 Index

AmigaDos arguments

AmisED manual

ARexx

Credits

Disclaimer

Getting~started

Known bugs

Registration

Requirements

Save your files after a failure

Startup options

The Amis project

The editor

What's new

WYSIWYG

Internal~Commands

Internal Commands (Alphabetical)

Internal Commands (by subject)

Internal command: ArrangeWindows

Internal command: Assemble

Internal command: AutoIndent

Internal command: AutoSave

Internal command: Backspace

Internal command: BackUps

Internal command: Block

Internal command: Codebar

Internal command: Compile

Internal command: CopyBlock

Internal command: CreateIcons

Internal command: CursDown

Internal command: CursLeft

Internal command: CursMode

Internal command: CursorWrap

Internal command: CursRight

Internal command: CursUp

Internal command: CutBlock

Internal command: Debug

Internal command: Delete

Internal command: EditorSettings

Internal command: EOLMode

Internal command: ExecuteCommand

Internal command: FastMode

Internal command: File

Internal command: Find

Internal command: FindFunctions

 Internal command: Fold

Internal command: FontSettings

Internal command: GetASCII

Internal command: GetLine

Internal command: GetWord

Internal command: Goto

Internal command: Graphics

Internal command: Help

Internal command: Iconify

Internal command: Insert

Internal command: InternalCommand

Internal command: KeyboardSettings

Internal command: Line

Internal command: Link

Internal command: Macro

Internal command: MenuSettings

Internal command: Message

Internal command: MiscSettings

Internal command: OverWrite

Internal command: PasteBlock

Internal command: Print

Internal command: Project

Internal command: ProgramSettings

Internal command: Query

Internal command: Quit

Internal command: Redo

Internal command: Replace

Internal command: Request

Internal command: ResultToVar

Internal command: REXX

Internal command: Run

Internal command: ScreenSettings

Internal command: SelectWindow

Internal command: SetBookmark

Internal command: SetTab

Internal command: Settings

Internal command: Shell

Internal command: ShowAbout

Internal command: ShowStatistics

Internal command: ShowWindowManager

Internal command: SmartIndentSettings

Internal command: Statusbar

Internal command: Toolbar

Internal command: ToolbarSettings

Internal command: Undo

Internal command: UnFold

Internal command: Viewer

Internal command: Viewerbar

Internal command: Window

Internal command: WriteProtect

Menus

Windows

Window: 'About'

Window: 'Current window settings...'

Window: 'Default fonts...'

Window: 'Edit commands...'

Window: 'Editor...'

Window: 'Execute Command...'

Window: 'Find...'

Window: 'Find functions...'

Window: 'Information...'

Window: 'Internal command...'

Window: 'Keyboard...'

Window: 'Macro...'

Window: 'Menu...'

 Window: 'Miscellaneous...'

Window: 'Palette...'

Window: 'Programming...'

Window: 'Project Manager'

Window: 'Screen...'

Window: 'Smart Indent...'

Window: 'Toolbar...'

Window: 'Undo list...'

Window: 'Window Manager'
