

# **AMIS**

Willem Mestrom

**COLLABORATORS**

	<i>TITLE :</i> AMIS		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Willem Mestrom	January 18, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>AMIS</b>	<b>1</b>
1.1	AMIS Help guide . . . . .	1
1.2	legal stuff . . . . .	2
1.3	What is AMIS . . . . .	2
1.4	Startup options . . . . .	3
1.5	The editor . . . . .	4
1.6	Windows . . . . .	4
1.7	'Find...' window . . . . .	5
1.8	'Find functions...' window . . . . .	6
1.9	'Undo list...' window . . . . .	6
1.10	'Macro...' window . . . . .	6
1.11	'Statistics...' window . . . . .	7
1.12	'About' window . . . . .	7
1.13	'Project Manager' window . . . . .	7
1.14	'Window Manager' window . . . . .	8
1.15	'Toolbar...' window . . . . .	8
1.16	'Editor...' window . . . . .	9
1.17	'Keyboard...' window . . . . .	10
1.18	'Menu...' window . . . . .	10
1.19	'Programming' window . . . . .	11
1.20	'Internal command...' window . . . . .	12
1.21	'Execute Command...' window . . . . .	12
1.22	'Edit commands...' window . . . . .	12
1.23	Requirements . . . . .	13
1.24	Getting started . . . . .	13
1.25	Registration . . . . .	13
1.26	Save your files after a failure . . . . .	16
1.27	Credits . . . . .	16
1.28	Known bugs . . . . .	17
1.29	Improvements . . . . .	17

---

---

1.30	ARexx . . . . .	17
1.31	Internal Commands . . . . .	18
1.32	Alphabetical list of the Internal Commands . . . . .	19
1.33	Available Internal Commands . . . . .	22
1.34	Internal command: CursUp . . . . .	24
1.35	Internal command: CursDown . . . . .	25
1.36	Internal command: CursLeft . . . . .	25
1.37	Internal command: CursRight . . . . .	25
1.38	Internal command: Goto . . . . .	26
1.39	Internal command: SetBookmark . . . . .	26
1.40	Internal command: Backspace . . . . .	26
1.41	Internal command: Delete . . . . .	26
1.42	Internal command: Line . . . . .	27
1.43	Internal command: Insert . . . . .	27
1.44	Internal command: File . . . . .	27
1.45	Internal command: EOLMode . . . . .	28
1.46	Internal command: CursorWrap . . . . .	28
1.47	Internal command: WriteProtect . . . . .	29
1.48	Internal command: SetTab . . . . .	29
1.49	Internal command: BackUps . . . . .	29
1.50	Internal command: AutoSave . . . . .	30
1.51	Internal command: AutoIndent . . . . .	30
1.52	Internal command: CreateIcons . . . . .	30
1.53	Internal command: FastMode . . . . .	30
1.54	Internal command: OverWrite . . . . .	31
1.55	Internal command: Query . . . . .	31
1.56	Internal command: GetWord . . . . .	32
1.57	Internal command: GetLine . . . . .	32
1.58	Internal command: GetASCII . . . . .	32
1.59	Internal command: Toolbar . . . . .	32
1.60	Internal command: Codebar . . . . .	33
1.61	Internal command: Statusbar . . . . .	33
1.62	Internal command: Viewerbar . . . . .	33
1.63	Internal command: Viewer . . . . .	33
1.64	Internal command: Request . . . . .	34
1.65	Internal command: Message . . . . .	34
1.66	Internal command: Find . . . . .	35
1.67	Internal command: Replace . . . . .	35
1.68	Internal command: FindFunctions . . . . .	36

---

---

1.69 Internal command: Block . . . . .	36
1.70 Internal command: CutBlock . . . . .	37
1.71 Internal command: CopyBlock . . . . .	37
1.72 Internal command: PasteBlock . . . . .	37
1.73 Internal command: Undo . . . . .	38
1.74 Internal command: Redo . . . . .	38
1.75 Internal command: Macro . . . . .	38
1.76 Internal command: Select Window . . . . .	39
1.77 Internal command: Window . . . . .	39
1.78 Internal command: ArrangeWindows . . . . .	40
1.79 Internal command: Compile . . . . .	40
1.80 Internal command: Assemble . . . . .	41
1.81 Internal command: Link . . . . .	41
1.82 Internal command: Debug . . . . .	42
1.83 Internal command: Run . . . . .	42
1.84 Internal command: Project . . . . .	42
1.85 Internal command: ExecuteCommand . . . . .	43
1.86 Internal command: InternalCommand . . . . .	43
1.87 Internal command: Shell . . . . .	43
1.88 Internal command: Rexx . . . . .	44
1.89 Internal command: ShowStatistics . . . . .	44
1.90 Internal command: ShowWindowManager . . . . .	44
1.91 Internal command: ShowAbout . . . . .	44
1.92 Internal command: ToolbarSettings . . . . .	44
1.93 Internal command: EditorSettings . . . . .	45
1.94 Internal command: KeyboardSettings . . . . .	45
1.95 Internal command: MenuSettings . . . . .	45
1.96 Internal command: FontSettings . . . . .	46
1.97 Internal command: ScreenSettings . . . . .	46
1.98 Internal command: ProgramSettings . . . . .	46
1.99 Internal command: Settings . . . . .	47
1.100 Internal command: Iconify . . . . .	47
1.101 Internal command: Quit . . . . .	47
1.102 Internal command: ResultToVar . . . . .	47
1.103 Internal command: Help . . . . .	48

---

# Chapter 1

# AMIS

## 1.1 AMIS Help guide

```

AMIS V0.95
Copyright © 1993-1996 Willem Mestrom

What~is~AMIS~~~~  ~~
    Purpose and philosophy...

Getting~started~  ~~
    If you don't no what to do...

Requirements
    What do I need for all this...

Startup options
    CLI and Workbench startup options...

The editor
    What can you do with the editor...

Windows          ~~
    What are all windows for...

Internal~Commands  ~
    All about the internal commands...

ARexx
    The ARexx facilities of AMIS...

Save your files
    Very important after a failure...

Registration
    How do I get the full version...
```

---

```
Legal stuff
    For my security...

Credits~~~~~      ~~~~~
    Who made this...

Bugs~& fixes~    ~~~~~
    Are there any bugs (fixed)...

Improvements
```

## 1.2 legal stuff

--- SOME VERY IMPORTANT INFORMATION FOLLOWS! ---

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!

## 1.3 What is AMIS

Philosophy:

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

AMIS is an editor designed for only one purpose: Make Life Easier. This means that I tried to make it as user-friendly as I could, but still give you all the special features. Things that never were can always come...

What I find to be most difficult in most editors is trying to create a nice ANSI text, with all the colours and text styles. Therefore AMIS has a special ANSI editing mode. This gives you a 'bar' with all functions you need to change colours and text styles. Of course you can also see the colours and styles while you are editing.

For AmigaGuide databases AMIS has got something much the same as for ANSI texts. Again there is a special editing mode which allows you to change colours and styles, create and change buttons and create new nodes. And it is all displayed in much the same way as multiview does. So no more 'try and error', just do it!

Here follows a list of the main features of AMIS, just to give you an impression of what is there:

- o All windows, menus etc. have the normal OS3.0 look and feel.
- o 3 Different viewers/editors for normal text, ANSI text and AmigaGuide database files. (So you can SEE the colours and buttons you create!)

- o A 'Viewerbar' for the ANSI viewer/editor. This bar allows you to change the foreground colour, background colour and text style with just one mouse-click.
- o A 'Viewerbar' for the AmigaGuide viewer/editor. This bar allows you to change the foreground colour, background colour, text style, add a button, add a node and change a link with just one mouse-click.
- o A 'Project Manager' which allows you to make a list of text files which in some way 'belong together'. Very useful for the different source files of a program for example.
- o A 'Toolbar' (which can be changed according to your wishes) very much like the one we now from Wordworth.
- o A very powerful set of internal commands (more than 65 commands!) which are at the same time the ARexx commands.
- o All keys are 'user defined'. All keys can be given functions by editing the internal commands which should be executed when the key is pressed.
- o User settings for programming functions like Compile, Assemble, Link, etc.
- o Settings windows for the Toolbar, Editor, Keyboard, Menu, Screen, Font and Programming settings.
- o Macro support. You can record, play, load, save and edit macro's. Shortcuts and macro names are also supported.
- o All the usual editor functions like Find (& Replace), block commands (with clipboard support), statistics, etc.
- o Although it is only version 1.00 AMIS is very stable (with thanks to my beta testers Vincent Groenewold, Robert Nienkemper and Jos van Oijen).
- o And a lot more!!

## 1.4 Startup options

AMIS can be started from the Workbench or from the Shell. If you start AMIS 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:

```

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

From the Shell you can also give some startup options. AMIS 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,STARTNEW/S

All options are the same as the tooltypes, where FILE is the file you want to open on startup. STARTNEW will make sure AMIS will always start a new copy, no matter whether there is already an AMIS running 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 AMIS running, the new AMIS 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 AMIS running.

## 1.5 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 find the most complex viewerbar with functions to create and modify buttons, text color and text styles.

The statusbar is a bar under the viewerbar. In this bar some information is displayed like the line (L:) and column (C:) position of the cursor. Here you can also see in which viewing mode you are.

The codebar is a bar at the bottom of the window. In this bar you can see the same line as the one were 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'. For example commands in an AmigaGuide document.

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 using the item 'Menu...' in the 'Settings' menu (if not changed by the user ;- ) ).

There are also a lot of information and configuration windows available, for a complete list please see

Windows

.

## 1.6 Windows

Editing:  
Find...  
Find functions...  
Undo list...  
Macro...  
Information:  
Statistics...  
About  
Managers:  
Project Manager  
Window Manager  
Settings:  
Toolbar...  
Editor...  
Keyboard...  
Menu...  
Programming...  
Other:  
Internal command...  
Execute command...  
Edit commands...

## 1.7 '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.
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.

## 1.8 '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.

## 1.9 'Undo list...' window

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

The gadgets:

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

## 1.10 '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.

## 1.11 'Statistics...' window

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

Gadget	Function
Tab size	Change the tab size for this text (the global tab size is defined in the ' Editor... ' window).
Ok	Close this window.
Next	Some information about the next text file in memory.
Cancel	Close this window and reset the tab size to the size it was before this window was opened.

## 1.12 'About' window

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

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

## 1.13 '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).

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').

## 1.14 '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').

## 1.15 '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 this 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).

## 1.16 '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. 'Wrap' will move the cursor to the next line. 'Ignore' will move the cursor further to the right (although there is no more text there).

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.

ANSI Colors    'ANSI Like' will show an ANSI file in the MagicWB colors in such a way that it best matches the original ANSI colors.

'DOS' will display an ANSI file in the colors as AmigaDOS would do. 'True ANSI' is the same as 'ANSI Like', only if you use a screen of 16 colors or more AMIS will display the file in the original ANSI colors.

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).

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.

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.

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).

Fast mode    Every thing will be displayed in black and gray, this will speed up the graphical routines like scrolling and redrawing.

Automatic linefeed    This will automatically jump the cursor and the word you are typing to the next line when the end of the line is reached.

Ok    Close this window.

Save    Save the new editor 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 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).

## 1.17 'Keyboard...' window

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

Gadget	Function
Keys	The lists of keys that have a special function.
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 0.95 -
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).

## 1.18 '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).

## 1.19 '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 AMIS should go to before executing the compile command.
Compiler comm.	The command AMIS should execute when a compile command was given.
Assembler draw.	The directory AMIS should go to before executing the assemble command.
Assembler comm.	The command AMIS should execute when a assemble command was given.
Linker draw.	The directory AMIS should go to before executing the link command.
Linker comm.	The command AMIS should execute when a link command was given.
C-Debug. draw.	The directory AMIS should go to before executing the C-debug command.
C-Debug. comm.	The command AMIS 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 AMIS should go to before executing the ASM-debug command.
A-Debug. comm.	The command AMIS 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.	Indicate that AMIS should go to the directory where the (5x) source was saved before executing the command displayed left from this gadget.
Use own proc.	Indicate that AMIS should detach this program (otherwise (5x) AMIS 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).

## 1.20 'Internal command...' window

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

Gadget	Function
Command string	The command you want AMIS 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.

## 1.21 'Execute Command...' window

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

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 AMIS while the command is being executed).
Cancel	Close this window.

## 1.22 'Edit commands...' window

The 'Edit commands...' window can be used to change a list of  $\leftrightarrow$  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.

## 1.23 Requirements

To make life easier (See What is AMIS) you need a computer that makes life easier. Which means in this case:

- o Kick 3.0, although AMIS will start on Kick 2.0, I don't think it will be very stable. I'm working on it, but for now Kick 3.0 is strongly recommended.
- o An 68020 or better.

## 1.24 Getting started

If you start AMIS as you find it after using the 'Install' script, AMIS will open with an empty editor window. Here you can type your text and edit it using the menus. For ANSI files or AmigaGuide databases you can switch to the appropriate viewer using the 'Viewer' options in the 'Window' menu.

You can get this help file using the 'Help' option in the 'Misc' menu or pressing 'Amiga-H'. You can customize AMIS 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.

If you want the editor to do something and you can't figure out how you can do this you can send me an E-mail. I'll try to explain how you can do this, or if it is not possible with the current version of AMIS I'll try to built it into the next version. (This might get you a free registration and update!)

Good luck!

## 1.25 Registration

Why should you register ?

There are a lot of reasons why you could & should register, here are just a

---

few of which I think that they may be important.

For me:

- o I can only make AMIS up-to-date and compatible with new hardware if I can at least test it on new machines. This is the main reason why this program is shareware, why should I ask money if I didn't need it ?
- o It will motivate me to make AMIS better than any other editor.

For you:

- o You can save files bigger than 5 Kb with the registered version.
- o You will get the latest version of AMIS.
- o It's a lot cheaper then commercial editors or shareware editors like GoldED.
- o It's illegal to copy a keyfile from someone else.
- o Your name will appear in the about window.

How can I register ?

First of all make sure this documentation is less then 1 month old, otherwise you might send the wrong amount of money to the wrong address!! After you checked this do the following:

Fill in the orderform as below (Also in file: Orderform) and sign it. Then send it (in an envelop) to this address:

Willem Mestrom  
 Anjerveld 27  
 4613 DC Bergen Op Zoom  
 The Netherlands

If you pay cash please make sure you can't see from the outside that there is any money in it, this will prevent 'getting lost'. You will get your disk with keyfile and the latest version of AMIS or an E-mail with your (uuencoded) keyfile (if you have an E-mail address you can probably also excess the Aminet, therefore sending you the latest version of AMIS would be a waist of time (and money)).

Prices:

Your country	Price (cash)		Price (account)
The Netherlands	F1 25,-	F1 25,-	(see below)
An other country in 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,-		

For more information please write a letter to me (address above) or send an E-mail to: supernov@zeelandnet.nl

If you want to order AMIS from within The Netherlands and pay by account:

Maak Fl 25,- over op giro rekening 4302419 van de POSTBANK. Vermeld bij de overschrijving je naam en adres (of E-mail adres) waar je je keyfile wilt ontvangen. Je hoeft geen 'Orderform' in te vullen.

If you want to order AMIS from outside The Netherlands and pay by account:

Transfer the money (see prices above) to the following account:

account nr: 4302419  
Willem Mestrom  
Anjerveld 27  
4613 DC Bergen op Zoom  
The Netherlands  
Swift code: ING B NL 2-A

Please note your full name and address (or E-mail address), if you do so you don't have to fill in the 'Orderform'.

Note: If you pay by account you don't have to fill in the 'Orderform', but if you have special hardware you want to be supported or any you have some suggestions for further improvement of AMIS you can always fill in the 'Orderform' anyway or send me a letter or an E-mail or what ever.

Order form:

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

Name: \_\_\_\_\_

Address: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Country: \_\_\_\_\_

E-mail: \_\_\_\_\_

Hardware: \_\_\_\_\_

\_\_\_\_\_

Suggestions: \_\_\_\_\_  
(for improvements)

\_\_\_\_\_

\_\_\_\_\_

I am sending the shareware fee in cash, included with this letter,

- o from The Netherlands F1 25,-
- o from Europe DM 25 / £ 12 / US \$18 / F1 27,-
- o from outside Europe DM 28 / £ 13 / US \$20 / F1 32,-

Please send me the keyfile

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

---

(Date)

---

(Signature)

## 1.26 Save your files after a failure

Although AMIS 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, AMIS isn't responding any more, you can still save your file!!

This can be done with a very special feature of AMIS. When you start AMIS, it will look for an other AMIS that might already be running. If there is an other AMIS it will be signalled to show a requester asking the user wether or not he wants to start a new copy of AMIS. If the other AMIS 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 AMIS to the file list of the new AMIS. 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 excess the files by just 'revealing' them.

So all you have to do is start AMIS once again and you will get your files back! (if possible).

## 1.27 Credits

This program is designed and written by:

Willem Mestrom  
Anjerveld 27  
4613 DC Bergen Op Zoom  
The Netherlands

Advice and beta testing are coming from:

Vincent Groenewold (is supernov)  
Louis Armstrongarf 13  
4614 XV Bergen Op Zoom

---

The Netherlands  
E-mail: supernov@zeelandnet.nl

Robert Nienkemper  
The Netherlands

Jos van Oijen  
The Netherlands

Help for the listview programming came from:

Leon Makkink  
The Netherlands

If you have any hints, ideas, suggestions or something else to say: please don't hesitate to write to Willem Mestrom. And maybe next release your name will be in this list too.

## 1.28 Known bugs

Known bugs (hard errors & crashes):

- None.

Known bugs (small errors):

- There might be something wrong with the 'Insert File' function, but it will probably work correctly most of the time.
- The undo functions are not working very well yet, I'm still working on them.

Bug fixes in version 0.95:

- Clear command now works properly.
- Removed an enforcer hit.
- Quit requester is now programmed according to the official programming rules.

## 1.29 Improvements

Improvements in version 0.95:

- AMIS now has commodition support.

## 1.30 ARexx

AMIS 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 AMIS 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 AMIS. Since these are

that commands that are used for menu, macro and key definitions you can understand these are quite powerful.

One way to add new functions to AMIS 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!

## 1.31 Internal Commands

AMIS 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~~~~~  
Every thing the control cursor movement...

Editing~~~~~  
Commands to edit 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~

## 1.32 Alphabetical list of the Internal Commands

All Internal Commands in alphabetical order:

ArrangeWindows  
Assemble  
AutoIndent  
AutoSave  
Backspace  
BackUps  
Block  
Codebar  
Compile  
CopyBlock  
CreateIcons  
CursDown  
CursLeft

---

---

CursorWrap  
CursRight  
CursUp  
CutBlock  
Debug  
Delete  
EditorSettings  
EOLMode  
ExecuteCommand  
FastMode  
File  
Find  
FindFuncions  
FontSettings  
GetASCII  
GetLine  
GetWord  
Goto  
Help  
Iconify  
Insert  
InternalCommand  
KeyboardSettings  
Line  
Link  
Macro  
    Make  
Message  
MenuSettings

---

OverWrite  
PasteBlock  
Project  
ProgramSettings  
Query  
Quit  
ResultToVar  
Redo  
Replace  
Request  
Rexx  
Run  
ScreenSettings  
SelectWindow  
SetBookmark  
SetTab  
Settings  
Shell  
ShowAbout  
ShowStatistics  
ShowWindowManager  
Statusbar  
Toolbar  
ToolbarSettings  
Viewer  
Viewerbar  
Window  
WriteProtect

---

Undo

### 1.33 Available Internal Commands

All Internal Commands arranged by subject.

Cursor control:

CursUp

CursDown

CursLeft

CursRight

Goto

SetBookmark

Editing:

Backspace

Delete

Insert

Line

Macro

Blocks:

Block

CutBlock

CopyBlock

PasteBlock

Search:

Find

Replace

FindFunctions

Programming:

Compile

Assemble

Link

Make

---

---

Run

Debug

File & projects:

File

Project

Messages:

Request

Message

Info Windows:

ShowStatistics

ShowWindowManager

ShowAbout

Editor windows:

ArrangeWindows

SelectWindow

Window

Commands:

ExecuteCommand

InternalCommand

Shell

Rexx

Editor options:

EOLMode

CursorWrap

WriteProtect

SetTab

BackUps

AutoSave

AutoIndent

CreateIcons

FastMode

---

---

OverWrite  
Toolbar  
Viewerbar  
Statusbar  
Codebar  
Viewer  
Settings:  
ToolbarSettings  
EditorSettings  
KeyboardSettings  
MenuSettings  
ScreenSettings  
FontSettings  
ProgramSettings  
Settings  
Other:  
Query  
GetASCII  
GetWord  
GetLine  
ResultToVar  
Help  
Undo  
Redo  
Iconify  
Quit

### 1.34 Internal command: CursUp

CursUp

---

Function: Move the cursor upward.  
Syntax: LINES/N,PAGE/S,TOP/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.

### 1.35 Internal command: CursDown

CursDown

Function: Move the cursor downward.  
Syntax: LINES/N,PAGE/S,BOTTOM/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.

### 1.36 Internal command: CursLeft

CursLeft

Function: Move the cursor to the left.  
Syntax: CHARS/N,WORD/S,SOL/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.

### 1.37 Internal command: CursRight

CursRight

Function: Move the cursor to the right.  
Syntax: CHARS/N,WORD/S,EOL/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.

---

## 1.38 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

Result: 0 - Window selected.

10 - There are no windows at all.

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 0.93 -

BOOKMARK: Move the cursor to bookmark BOOKMARK.

INTOP: Make sure the line the cursor will be on is in top of the editor window.

## 1.39 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

## 1.40 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.

## 1.41 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.

## 1.42 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.

## 1.43 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').  
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.

## 1.44 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.

## 1.45 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.

## 1.46 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

## 1.47 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.

## 1.48 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.

## 1.49 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.

---

## 1.50 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 AMIS 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 AMIS will save your file.

ON: Switch AutoSave mode on.

OFF: Switch AutoSave mode off.

ASK: Let AMIS ask you wether or not you want to save your file.

## 1.51 Internal command: AutoIndent

AutoIndent

Function: Switch AutoIndent mode on or off. When switched on and you insert a new line (with return) AMIS 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.

## 1.52 Internal command: CreateIcons

CreateIcons

Function: Switch the creating of icons on or off. When switch on AMIS 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.

## 1.53 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.

## 1.54 Internal command: OverWrite

OverWrite

Function: Switch OverWrite on or off. When switched on AMIS 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.

## 1.55 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

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.

## 1.56 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: -

## 1.57 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: -

## 1.58 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: -

## 1.59 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.

---

## 1.60 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.

## 1.61 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.

## 1.62 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.

## 1.63 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.

## 1.64 Internal command: Request

### Request

Function: Display a requester with info and or a question for the user. The text 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='Ok|12').

Syntax: BODY,BUTTON/K, LONG/S, VALUE/N, NOCENTRE

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.

See also:

Message

## 1.65 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

## 1.66 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

## 1.67 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

## 1.68 Internal command: FindFunctions

FindFunctions

Function: Show the 'Find functions...' window.

Syntax: ASK/S

Result: 0 - Ok.

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

## 1.69 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
```

## 1.70 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
```

## 1.71 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
```

## 1.72 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

## 1.73 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)

## 1.74 Internal command: Redo

### Redo

Function: Redo the action you undid with 'Undo'.

Syntax: -

Result: 0 - Ok.

10 - No editor window opened / selected.

Options: -

## 1.75 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.

## 1.76 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

## 1.77 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
```

## 1.78 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
```

## 1.79 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
```

## 1.80 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
```

## 1.81 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
```

## 1.82 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

## 1.83 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

## 1.84 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.

## 1.85 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.

## 1.86 Internal command: InternalCommand

InternalCommand

Function: Open the 'Internal command...' window.

Syntax: ASK/S

Result: 0 - Ok.

Options: ASK: Open the 'Internal command...' window.

## 1.87 Internal command: Shell

Shell

Function: Open a new shell window (on the AMIS screen).

Syntax: -

Result: 0 - Ok.

Options: -

## 1.88 Internal command: Rexx

Rexx

Function: Execute an ARexx script.

Syntax: SCRIPT/A

Result: 0 - Ok.

Options: SCRIPT: The file name of the script you want to execute.

## 1.89 Internal command: ShowStatistics

ShowStatistics

Function: Show the 'Statistics...' window.

Syntax: -

Result: 0 - Ok.

Options: -

## 1.90 Internal command: ShowWindowManager

ShowWindowManager

Function: Show the 'Window Manager' window.

Syntax: -

Result: 0 - Ok.

Options: -

## 1.91 Internal command: ShowAbout

ShowAbout

Function: Show the 'About' windows.

Syntax: -

Result: 0 - Ok.

Options: -

## 1.92 Internal command: ToolbarSettings

ToolbarSettings

Function: Load or save the toolbar settings and/or show the 'Toolbar...' 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 'Toolbar...' window.

### 1.93 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.

### 1.94 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.

### 1.95 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

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.

## 1.96 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.

## 1.97 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.

## 1.98 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.

## 1.99 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.

## 1.100 Internal command: Iconify

Iconify

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

Syntax: -

Result: 0 - Ok.

10 - Failed.

Options: -

## 1.101 Internal command: Quit

Quit

Function: Quit AMIS.

Syntax: FORCE/S

Result: 0 - Ok.

Options: FORCE: Quit without the usual quit requester.

## 1.102 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 were 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!

## 1.103 Internal command: Help

### Help

Function: Show the on-line help guide (this file).

Syntax: -

Result: 0 - Ok.

Options: -