AIDE Documentation 2.11

COLLABORATORS

	TITLE : AIDE Documentation 2.11		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AIDE Documentation 2.11

1.1 AIDE - AmigaGuide® Documentation

```
                 A I D E  v 2.12
                Table of Contents
                 Preface

 My name is
                Daniel Seifert
                . Perhaps some of you remember me from NAP,
 a preprocessor I wrote for the ACE language.

                Herbert Breuer
                , who developed the 2.x versions of AIDE until v 2.11,
 has asked me to continue his work on AIDE. Of course I couldn’t have
 said "no" ;-) as this is an interesting experience for me.
 Although  Herbert gave  AIDE  to me, he is still having a look at it
 for me  not making  any bad mistakes :-)  We work close together and
 you  might still
                address him
                  if you have any questions concerning
 AIDE.

 I hope to be able to continue Herberts wonderful work in a good way.
 Thanks in advance for your help and support.
                                                      Daniel Seifert
```

1.2 Table of Contents

```
                 Introduction

 AIDE stands for  "ACE Integrated Development Environment".  It  is a

                freely  distributable
                  graphical  front-end  for the  development of
 ACE programs.  By using AIDE you will never need again to start from
 a shell when developing ACE programs (although you may
                spawn a shell
```

by selecting an item from AIDE's Project Menu.

Table of Contents

1.3 System Requirements

System Requirements

AIDE requires at least AmigaOS Release 2.04. There is no chance that it will run on any lower OS release.
AIDE will be comfortable with 0.5 megabyte RAM. (But ACE might get trouble then) AIDE does use only 4096 bytes of stack and does set up a stack size of 40000 bytes for any process run by it (esp. ACE will be very grateful ;-)

1.4 Setting up

Setting up

Before starting AIDE for the first time however, you must have ACE installed correctly with the following corresponding directories:

```
ACE:
ACE:AIDE        (put at least the AIDE icon, e.g. AIDE.info,
                 here)
ACE:bin
ACElib:         (ACE:lib)
ACEinclude:     (ACE:include)
```

```
       ACEbmaps:        (ACE:bmaps)

       (see "ACE.doc"/ACE.guide for the details)

If these dirs aren’t set up correctly, AIDE will abort with an error
message.  Same  if  one  of the following files are not in the right
directory:

       mrt.library      (in LIBS:)
       thinpaz.font     (in FONTS:)      (These files are included
                                          within this package)

I recommend to have the followings directories as well:

       ACE:Docs         where all the docs reside
       ACE:Icons        where the necessary icons for ACE should be
       ACE:Mods         where your compiled SubMods reside

                        ATTENTION:
                        Subdirectories are not allowed!  Please  copy
                        all  your modules into  ACE:Mods/,  otherwise
                        the modules will not be available.

       ACE:prgs         where your source code resides
       ACE:run          where your executables are going
       ACE:temp         where your temporary files during compilation
                        are going

                        ATTENTION:
                        "ACE:temp/"  is  the  DEFAULT  directory  for
                        temporary files during compilation.  To  save
                        memory  they are not any longer stored in  T:
                        (if the default would be T: and T: resides in
                        memory  the source file would be two times in
                        memory) As memory’s always short, it’s better
                        to have the temporary dir on HD  and  not  in
                        RAM:. Change this to your personal taste.

       ACE:utils        where all the ACE related utilities reside
                        --> ACE:Utils/ACEcalc/Acecalc
                            ACE:Utils/ReqEd/ReqEd
                            ACE:Utils/FD2BMAP/FD2BMAP
                            ACE:Utils/AB2ASCII/AB2ASCII
                            ACE:Utils/UppercACEr/UppercACEr

If those dirs do not exist, AIDE’s internal
               configuration
                part gives
you the possibility to do the
               setup
                to your personal need.
```

1.5 configuration

Configuration

AIDE will create a new file called AIDE2.config to run properly. If you've old config files from AIDE 1.5 and before, your settings will be copied to the new file. You can later delete the old files, if you wish. You can not edit the new file anymore, just use the integrated setup part in AIDE.

No AIDE2.config is included in the AIDE archive. If there does not exist one, AIDE will create one with the default settings.

If all the directories exist, as mentioned above , AIDE will create the setup in this way that you do not need to setup these settings manually. Only for options like PubScreen, Requester, ... you need to open the setup window , anything else is already done.

As AIDE has no built-in editor you have to specify your favourite one (cmp. setup , otherwise "C:Ed" is used ...)
Note, that the specified editor MUST be capable of taking command-line arguments (e.g. a file name). Same applies to the viewer (default: AmigaGuide/Multiview).

1.6 AIDE's menus

AIDE's menus

There exist three pull-down menus in AIDE: Project, Utilities and Help. The Project menu is for general house-keeping tasks, the Utilities menu provides access to a number of useful programs and the Help menu gives you access to documentation for the compiler, assembler, linker and the ACE language.

Project menu

- Open ...
- View ...
- Rename ...
- Copy ...
- Delete ...
- Print ...
- Execute ...

.

Spawn Shell

.

AIDE setup

.

Load Config File

.

Save Config File

.

About

.

Iconify

.

Quit

Utilities menu

Here you find some useful utilities (not within the AIDE package). The corresponding program must be in the right directory as mentioned in the
installing section
, or there must be an entry in
AIDE's
setup window
.

Please note, that AB2ASCII is a little bit buggy. If you want to improve it, well, do it!

```
  Calculator          -   Invokes the calculator
  ReqEd               -   Invokes ReqEd
  Create BMAP file(s) -   Converts  FD file(s) for a library into a
                          .bmap  file  used  by  ACE and AmigaBASIC
                          when  calling  shared  library functions.
                          The .bmap file is stored in the ACEbmaps:
                          directory.
                          A file requester for selection is invoked
                          and you can select only one, all,  or the
                          specific files you want to convert.
  AmigaBASIC to ASCII -   Converts a compressed AmigaBASIC file in-
                          to an ASCII text file for use by ACE. In-
                          vokes a  file requester  for selection of
                          the input and output dir and file name.
  UppercACEr          -   Invokes UppercACEr

                          Important:
                              UppercACEr.Reserved must reside in S:

                          In any case you don't have to worry about
                          the filenames. AIDE handles that for you.
                          The converted file will have an _u exten-
                          sion to the filename.
  Utility 0 to 3      -   Here you have the possibility  to specify
                          any utility of your choice.  A correspon-
                          ding entry in AIDE2.config must exist!

                          You can only use tools,  which don't need
                          an argument  to run  (this may be changed
                          in a future release).
```

Help menu

This menu provides access to the documentation for the compiler, SuperOptimizer, assembler and linker from within AIDE.
In addition, access to the ACE Language Reference manual ACE examples, ACE history and this document can be gained via the Help menu.

The Help function'll do nothing if no viewer at all is specified.

All the supported doc files must be in your specified "Doc Dir".

The supported docs are:

```
AIDE.doc/AIDE.guide
ACE.doc/ACE.guide
SuperOptimizer.guide
A68K.doc
PhxAss.guide
BLink.doc
PhxLnk.guide
ref.guide
ace-rwords
example.guide
history
ACEcalc.doc (/ACEcalc.guide if available)
ReqEd.doc   (/ReqEd.guide if available)
```

Note:

The docs must exist under the names described above. AIDE relies on them. If you have other docs, you can view them with the function
View
from the "Project" menu in any case.

1.7 Open ... (Project menu)

Open ...

Invokes the editor with a file name which you must specify via a file requester. If no file name is given, AIDE will inform you with a message.
This option is included to edit any file of your choice. For editing the source file see below. The current used dir will be remembered, so you don't have to switch to all the directories again.

1.8 View ... (Project menu)

View ...

Lets you view a file which is selected via a file requester. Remembers the current dir.

1.9 Rename ... (Project menu)

Rename ...

Lets you rename a file specified via a file requester. Remembers the current dir.
First a file requester will be opened where you have to select a file, which you want to rename. Now a second file requester appears where you have to select the new name. You can change the directory name, too, so a rename has the same effect as "move a file". This is only possible for the same device.

1.10 Copy ... (Project menu)

Copy ...

Lets you copy a file specified via a file requester. Remembers the current dir.
There will be also two file requester: the first one to choose the source file, the second one to choose the destination dir and file name. If you change the name for the destination file, it will be copied with the new name.

1.11 Delete ... (Project menu)

Delete ...

Lets you delete a file specified via a file requester. Remembers the current dir.

Note:
No second requester will appear, asking, whether you really want to delete this file. You can’t double click the filename. You have to choose "OK" to perform the action.

1.12 Print ... (Project menu)

Print ...

Lets you choose a file to print via a file requester. Remembers the current dir.

Note:
This function is very simple. It uses your printer preferences, as setup via the prefs tool from your workbench disk. No page checking, no formatting, nothing else is done.
The function prints files, no matter which size, when they can be loaded in memory.
This may be more comfortable in a future release. ;-)

1.13 Execute ... (Project menu)

Execute ...

Allows an AmigaDOS command to be executed. The command and its arguments are entered into an input requester, a console window is opened for any I/O associated with the command and the latter is executed if possible.

1.14 Spawn Shell (Project menu)

Spawn Shell

A NewShell process is started for more comprehensive house-keeping tasks than the Execute ... option is intended for. Your default settings as defined in S:Shell-Startup are used.

1.15 Load and Save Config File (Project Menu)

Load Config File & Save Config File

You can create now multiple config files. With these two functions you can load/save now the default or any other config file.

default:

- will try to load/save a file named ACE:AIDE/AIDE2.config

other:

- will bring up a file requester where you can specify any directory and filename of your choice.

1.16 About (Project menu)

About

Brings up a message box telling you briefly about AIDE.

1.17 Iconify (Project menu)

Iconifiy

AIDE does now allow you to close the main window and to have an app-icon on the workbench. To wake AIDE up you just have to double-click on the appicon.

Note: To work properly, the AIDE icon (AIDE.info) must reside in the directory ACE:AIDE/. If this isn't the case, AIDE will use the default program icon (as can be specified via IconEdit) for the appicon.

1.18 Quit AIDE (Project menu)

Quit AIDE

Exits from AIDE. (You can also click the close gadget of the main window.) Depending on the settings (see SetUp) you will be asked whether you want to quit AIDE.

1.19 Setup

Setup

This is completely new, so you don't have to edit a config file outside of AIDE. All the settings you change there will be active immediately (despite of one).

Just one word to the default font setting:

"Thinpaz.font" is used so all the gadgets will fit in a window which can be displayed on a standard NTSC screen. For those who use a higher resolution of their Workbench screen, the option "Public Screen" can be used.

Directory Setup:

Well, when you choose AIDE setup the main window will be closed and the setup window will be opened. On the left side there are all the string gadgets for the directory setup. If you want to change the names manually you can do it there. If you type an error the old entry will be displayed again and the string gadget will be set to active state.
When you select the "R" button ("R" means requester), right beside the string gadget, an ASL file requester will pop up where you can choose a directory or file name of your choice.

Note:
If only a directory is necessary for the setup a selected file name will be ignored.
If no file name is selected where one is absolutely necessary, your choice will not be accepted.

```
Other:

  Public Screen:
       Here you have the choice to define,  if AIDE should run on an
       own public screen.  This screen'll have "Shanghai" abilities.
       That means that all windows which would normally be opened on
       the Workbench screen,  will now be opened on the AIDE screen.
       The AIDE screen will close automatically  if the  AIDE window
       and all the visitor windows are closed.

       This is the only setting  which will  have effect,  when  you
       start  AIDE  again  after saving your config file  as default
       (AIDE2.config).

       This may be changed in a future release.

  Requester:
       Here you have the choice to define:
               All:    all requester will be displayed
               Error:  only error requester will be displayed
               Min:    only fatal error requester will be displayed!

       Just try it out and set this flag to your personal taste.

  Clean up:
       Here you have the choice to define:
               Yes:  This is the default setting and that means that
                     all the  temporary files  will be deleted  when
                     you quit AIDE.
               No:   If  you  select  this,  then no temporary files
                     will be deleted when you quit AIDE. That may be
                     useful  if  you  want  to have a look  at those
                     files later. Older versions of AIDE will always
                     delete those files.
                     Still, all temporary files will be deleted when
                     you perform a new compiler run.

                     Note:

                     ACE.err is now found in ACE:temp  and no longer
                     in ACE:AIDE.


  Menu Pen:
       Here  you can select the color of the menu pen  from  0 to 2.
       This may be useful  if you do not like the default setting of
       menu pen 0 (background color).
```

1.20 Icon Tooltypes

```
Icon Tooltypes

You can configure AIDE now  via the tooltypes entry  of  the  source
file icon and the AIDE program icon. The defined tooltypes are:
   FILETYPE    =   ACESource

                   If  you are  using  icons  for your source files,
                   this tooltype must be set  if you wish the source
                   file to be set within AIDE.

                   Note:
                   This entry is ignored  if set in the AIDE program
                   icon.


   CONFIGFILE  =   AIDE2.config or any other config file name you've
                   specified for this source file.


   PRECO       =   app or any other precompiler you want to use with
                   this source file (preferable NAP ;-).

                   Note:
                   An example source icon is in the AIDE archive.
```

1.21 Main Window

```
                Main Window

The main window holds now the functions  and many more,  which  were
found before in the pull down menu.

All these functions are necessary  to compile an  ACE program  under
different conditions in a comfortable way.


.
               The Source Section
                .
               The Program Section
                .
               The Make Section
                .
               The Precompiler section
                .
               The ACE Options Section
                .
               The SuperOptimizer Section
                .
               The View Section
                .
               The Assembler Section
                .
               The Linker Lib Section
                .
               The Linker Section
                .
               The Module section
```

1.22 The Source Section

```
 Source

    Set          - Lets you specify the ACE source file  to work with,
                   via a file requester.  Some gadgets are not access-
                   ible until this has been done for the first time.
                   The file requester for "Set Source..." starts up in
                   the directory  which  was  referenced the last time
                   option was invoked.
                   If the file selected does not exist, AIDE'll invoke
                   the editor.

    Reset        - Resets the current  source file  for a new compiler
                   run. Useful if you did invoke the editor outside of
                   AIDE.

    Edit         - Invokes the editor with the current source file  as
                   its  argument.  If  no  editor  is  specified, this
                   option will be disabled
```

1.23 The Program Section

```
                Program

    Precompile  - This will invoke the specified precompiler with the
                  current source file as its argument.  No check,  if
                  the source file was already precompiled is done.

    Compile     - This will invoke ACE and the SuperOptimizer (if set
                  to active). No check if the source file was already
                  compiled is done.

    Assemble    - This will invoke the selected  assembler.  No check
                  if the source file was already assembled is done.

    Link        - This will invoke the selected linker.  No check  if
                  the source file was already linked is done.

    Run         - Runs the current program,  compiling  it  first  if
                  necessary.

                  Important:

                  This  functions  can now be used for  CLI programs,
                  too, but no input box will appear.  If you need  to
                  specify a command line argument  use "Run in Shell"
                  instead.  But if you do  not need  to enter command
                  line arguments,  this function will do now the same
                  task.

    Run in Shell- Runs the current program in a shell,  compiling  it
                  first if necessary. Before the program is run, AIDE
                  asks for command line arguments via an input box.
```

Supply no arguments if none are required, or use "Run" instead.

Important:

The buttons in the "Program" section will not change their state if you change any names/options for
precompiler
,
ACE
,
assembler
,

linker lib
or
linker
and the internal state of the
make process
won't be affected either.

If you did change the options just select the corresponding buttons in the "Program" section to perform the missing steps of your compilation process.
This habit differs completely from older versions of AIDE. It was done this way, so that the program is not recompiled completly, because only the assembler or linker option had been changed.

If you want to perform a new compiler run, just select "Reset" from the
source section
and "Make Executable" or whatever AIDE should do for you.

1.24 The Make Section

Make

Executable -
Makes
the current program, putting all intermediate files and the final executable program into the temporary working directory.

Application - Does the same as "Make Executable" but also copies the final executable from temporary storage to ACE:Run. This is the default.
If the default is overwritten with an empty entry, the executable goes to the root directory where AIDE resides.
An icon is also copied. That is the default for "Make Application". You don't need to specify "Create Icon" for the
ACE options
.

Module - Creates an object module which can later be linked with any ACE program. The only difference between this option and Make Executable (above) is that no linking occurs and ACE’s -m switch is used. See also "Creating & using ACE subprogram modules" in ace.doc.
The object module goes to ACE:Mods (default), or where the entry in AIDE2.config points to. If this is empty, the module goes to the root directory, where AIDE resides.

The displayed module list will be updated if the entry point in your configuration is setup correctly.

1.25 The Precompiler Section

Precompiler

The mutual exclusive buttons gives you the possibility to choose APP, ACPP or any other precompiler to be used to precompile your source file.

If you select "other", an input box will pop up where you can enter the name of your desired precompiler.

If you choose DCPP then the options will be set up correctly to be used for ACE programs. DCPP is much faster as ACPP and does the same task.

I recommend to use NAP (guess why ;-) as it’s almost as fast as ACPP (and I hope it will be faster the next versions) and does the same task as ACPP and APP together (processing defines, C comments AND ACE comments!)

If no precompiler name is entered, the button of your last selected one will be activated again and an existing precompiler name will be deleted.

Clicking the close gadget of the window will ignore all what you may have entered in the string gadget. If your settings should take effect press <RETURN>.

1.26 The ACE Options Section

ACE Options

Break Trapping
Assem. Comments
Create Icon
Optimise

Window Trapping

These five correspond to the compiler’s command-line switches: b, c, i, O and w respectively. See ace.doc’s "Compiler options" section for details of each of these.

"List Source Lines" is not available via checkbox gadget, if you need this, just enter "l" in the input box after having selected "other"!

Since AIDE always uses ACE’s -E switch to create an error file, this option is not available. Compilation errors are, however, viewable via "View Compiler Errors"!

other - If you select this button an input box will pop up and give you the possibility to enter any kind of ACE option which may be defined new for the future. Use this as well if you want to specifiy "list source line".

You have to enter the options as defined in ACE.doc. If no "space" is required, then don’t enter it.

If you enter an empty string here all options which may be defined before will be deleted.

Clicking the close gadget of the window will ignore all what you may have entered in the string gadget. If your settings should take effect press <RETURN>.

1.27 The SuperOptimizer Section

SuperOptimizer

Here you can choose if you want to use the SuperOptimzer from Manuel Andre. With "Set Level" you can set the the superopt level. A input box will pop up, where you can enter the number. Only numbers from 1 to 12 are accepted. The default value is set to 12.

Note:
The SuperOptimizer is used in combination with the make process of ACE. If you did change the level you have to recompile the source again with ACE using the "Compile" button.

If you did switch off the Superoptimizer, you don’t have to recompile again, because the unoptimized file <name>.s still exists and AIDE will use this one instead of <name>_opt.s.

In a future release it might be possible to select the SuperOptimizer separately.

PS: Unfortunately Manuel Andre changed the way to specify the optimizing depth from numbers to letters. Until now it has not been possible to make AIDE working with that. Therefore, you’ll have to use older versions of SuperOptimizer which do still support the numbers.

I hope to be able to find a way to support the newer version in the next versions of AIDE.

1.28 The View Section

View

Precompiled Source
- Allows you to view the preprocessed source file.

Assembly Source
- Allows you to view the target assembly source file which results from compiling a program.

Compiler Errors
- Lets you see the errors (if any) generated by the last compilation with ACE.

Note, that these options are only available if a viewer has been specified.

1.29 The Assembler Section

Assembler

Here you can choose which assembler you like to use. A68K and PhxAss (see the corresponding doc files for more details) are fully supported from AIDE.

Just select the desired options, or enter new ones with the "Set Options" buttons. All what you enter has to be done the same way as if you would type them at the Shell prompt. For A68K and PhxAss the options are set in the right place of the command string. PhxAss ist set to NULL for optimizing the assembly output. It does depend on your program which optimizing options you can use. Read the docs carefully and try them. Optimizing the assembly output results in shorter/faster executables.

When you specify another assembler via the "other" button then the options will be added after the name of the file which has to be assembled.

Example:

```
 superassembler test.s ......... this is the default used name build
                                 out of the assembler name  and  the
                                 compiled source name.
```

Then all what you did enter via the "Set Option" button will be added! You have to specify an object file name "test.o" if the assembler needs it to create a linkable module.

But be aware:

If you did define options for A68K and you switch to PhxAss then you have to delete those options first, otherwise they will be used also for PhxAss. This will result in an error message from PhxAss.

Small Code/Small Data model:

Please read the corresponding docs for the assemblers to this point very very carefully. Not all assemblers does the same task!!!
Don't choose these options if you are not sure what they are doing.

Debug Info:

If you select this, A68K or PhxAss will create a symbol table for all the labels and variable names which exists in the .s file. This is especially helpful if also "Assembly Comment" is selected as ACE option.
The compiled program can then be watched with a symbol debugger like MonAm. And you can compare it with your assembly source file.

1.30 The Linker Lib Section

Linker Lib

Here you can choose which linker lib despite of the defaults db.lib and startup.lib, which are linked always, should be used. For PhxLnk you can use the standard Amiga.lib instead of Ami.lib.

Important:

Blink will crash if you try to link Amiga.lib!!!!!!

1.31 The Linker Section

Linker

Here you choose which linker you like to use. BLink and PhxLnk are fully supported by AIDE.

What I did say about the handling of the "Set Options" button above in the
assembler
chapter is also valid for the linker part. So read the linker docs carefully if you want to use other options as predefined in AIDE.

SmallCode:

If you choose this option the supported linkers will link all your program code into one piece. This results in shorter executable but can't be "scattered loaded". See, in any case, the corresponding linker docs files for the details.

Small Data:

This has the same effect for the data section of your program as described above for the program section.

Important:

PhxLnk links all data sections into one piece, no matter if for some special ones the Chip mem bit is set! This can result in the effect that some machines are reporting "not enough memory" if they try to load a program wich was linked with PhxLnk and this option set. That happens, because ACE creates large data sections (depending on the used amount of variables) which has to be loaded to Chip mem now. Machines with only 512k and sometimes those with 1Mega byte of chip mem, will report "not enough memory".

PhxLnk has options to force Chip mem data sections to Chip mem and all the other data sections to Fast mem, but that has to be done inside of the assembler source file. :-(This is not possible for normal :-) ACE programmers. We only can hope that Klaus will be so kind to add an option to PhxLnk, so this'll be done automatically.

Blink separates Chip mem data sections from the others, so this'll not happen with BLink.

In any case you should be aware, that this option only makes sense, if also the assembler option was set to SmallCode/Small Data. If it was not (and ACE does not allow SmallCode/Small Data for all your programs) and SmallCode/Small Data option set for the linker only, results in programs which can have loading problems on machines with less memory and high memory fragmentation.

No Debug Info:

If you select this option the program will be linked ignoring all defined symbol table entries, if they exist.

1.32 The Module section

Module

All SUBMods which resides in the specified "Module Dir" are displayed in the main window. Just click on the name to select the one you want to link and the choosen one will be highlighted. Clicking on a selected entry results in unselecting it. If you want to remove all modules just click on the button "Remove all Modules"

There is no limit for the amount of Modules defined which can be linked.

The amount is only limited by the length of the command line the linker accepts for input. My testing did result in about 20 modules as maximum (depending on the length of the file name). The buffer of AIDE allows up to 120 modules until an error message is displayed.

Note:

No checking of the filename is done, only directory names are ignored. Therefore you can place any type of linkable file into your "Module Dir".
If you want to link other linker libs as predefined just place them here and select them as described above.

Note:

If you want to save your option settings from the main window, use

Save Configuration from the Project menu. All your settings will be saved in AIDE2.config (default) or in another file if you wish to do so.

1.33 AIDE's Make Process

AIDE's "Make" Process

As you may have gleaned from the above, AIDE has a simple inbuilt "make" capability.

This means that before it builds or makes an exectuable and prior to compiling or running a program, AIDE checks to see whether compilation needs to take place at all.

If you have created an executable and then decide to run it, AIDE will not recompile the source first. Likewise, if you want to build an executable and the program has already been compiled, the executable will be copied from AIDE's temporary workspace only.

In like manner, if you have compiled a program to produce an assembly source file, so long as you don't edit the ACE source file, AIDE will not recompile the program before assembling and linking it.

If you want to recompile select Source Reset or do it step by step with the buttons in the Program section.

On the other hand if you edit a source file, next time you go to run it, AIDE will "bring the program up to date" by compiling, assembling and linking it first.

If you forgot to link some modules just select them and select "Link" from the
Program
section.

In order to preprocess, compile, assemble and link a program, AIDE now uses the DOS function SystemTagList(). An intuition window will be opened to watch the progress of your compilation run. If you select "Stop" there, the next step will not be taken. If you want to stop immediately you have to press Ctrl-C in the CON: window, which was also opened to display the current output from ACE and the other programs. But not all programs, used to make an executable, accept Ctrl-C, therefore the "Stop" button was added.

Note:

Only the next step can be aborted via the "Stop" button, because SystemTagList() didn't return before and did give the control back to AIDE.

PIPE: is not used anymore!

If an error or user break occurs during preprocessing or compilation, AIDE will stop and give the control back to the user.

1.34 credits

Credits:

I have to thank the following persons for their help, and for their patience to test all my bad and buggy :-) beta versions.

David Benn (ACE),
Frank Wille (PhxAsss, PhxLnk),
Klaus Gessner

1.35 history

Well, Herbert never started a history so a long period is missing. Myself is also not that successful in keeping a history but I'll try my best :-)

```
Date       Changed
-----------------------------------------------------------
??/??/??   Since the last release 2.07 and this one (v2.12) were
           some other versions, still by Herbert, which have not
           been published. (Some little bugs were removed.)
```

```
>------------------------------------------------------------------<
04/08/97   got source code of AIDE, started to go through to un-
           derstand what happens where and why and how and.. ;-)
04/17/97   started to include iconization ability
04/27/97   rewrote manual (AmigaGuide format)
05/11/97   When loading a new configuration file via the menu it
           will now be checked whether the current configuration
           has changed.
```

1.36 Final Comments

Final Comments

AIDE is completely written in assembler. Nothing is perfect and AIDE still is not! So all your comments, hints and bug reports (huh? ;-) are welcome and wanted! Feel free to make any suggestions about AIDE and please report _any_ bug you've found to either Herbert or me:

Herbert Breuer
A.A. 5469
Bogota/Columbia
South America

Tel.: (Columbia)91/8261959

hbreuer@anditel.andinet.lat.co

Daniel Seifert
Elsenborner Weg 25
12621 Berlin
GERMANY

Tel.: (Germany)30/5669471

dseifert@hell1og.be.schule.de

In any case, I hope you find AIDE to be a pleasant environment in which to develop ACE programs.

One last word: AIDE is freeware. This means, you are allowed to copy it freely as long as all files stay unaltered and no file is missing in the copy. As the author does not want any money for AIDE, you are not allowed either to demand money for copying AIDE.

Regards,

Daniel Seifert
Berlin, Germany
19th May 1997