AIDE

COLLABORATORS			
	TITLE : AIDE		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 17, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AIDE

1.1 AIDE v1.1

```
                       +-----------+
          | AIDE v1.1 |
          +-----------+




                  Introduction

                  Setting Up

                  Using AIDE

                  AIDE's Make Process

                  Final Comments
```

1.2 Introduction

```
              Introduction
------------
```
AIDE (ACE Integrated Development Environment) is a freely distributable
graphical front-end for the development of ACE programs.

It is written in ACE and currently requires AmigaOS Release 2.04 or higher
to run.

AIDE uses the PIPE: device, so either your startup-sequence script must
contain the line "mount PIPE:" or you must type this from a shell before
running AIDE.

I will try to get a future version of AIDE running correctly under 1.3
when I get some time, but for anyone wishing to have a go at doing this I
have included the source code (which is also freely distributable but I
retain the copyright to it). The problems under 1.3 seem to be related to

the way in which AIDE uses a NewShell and PIPE. In any case, at present if
you start the program under 1.3, a requester will appear telling you that
you need Wb 2.04 or higher to run AIDE.

** Please note that this restriction only applies to AIDE not ACE!! **

Anyway, if you do have Release 2.04 or higher, you can start AIDE by
double-clicking its icon from the Workbench. With AIDE, you never need to
start from a shell again when developing ACE programs, although you may
spawn a shell by selecting an item from AIDE's
                Project Menu
                  (see below).

Before starting AIDE for the first time however, you must examine and possibly
edit one file. This configuration file is discussed next.

1.3 Setting Up

```
Setting Up
----------
AIDE requires a file called AIDE.config to be present in the same directory
as itself. Without this, the program won't run.

A sample AIDE.config is included in the AIDE archive. It looks like this:

---------------
#
# AIDE configuration file.
#
# For each AIDE configuration variable,
# replace its value with the full path
# of the tool or directory of your choice.
#
# All definitions are optional. Place a '#'
# before a name if you want to comment out
# the corresponding definition.
#


##
## TOOLS
##

# Editor (default=c:Ed).
#
EDITOR=sys:tools/memacs

# File viewer (default=ACE:bin/muchmore).
#
VIEWER=ACE:bin/muchmore

# Calculator tool (no default).
#
CALCTOOL=ACE:utils/ACECalc/ACEcalc
```

```
# AmigaGuide tool. If this entry is set, AIDE uses ref.guide, ace.guide
# and aide.guide instead of the equivalent .doc files.
#
# Example.Guide also relies upon this tool.
#
# The tool itself is likely to be either AmigaGuide or MultiView.
#
AGDTOOL=sys:utilities/AmigaGuide


##
## DIRECTORIES
##

# Temporary directory for preprocessor,compiler,assembler & linker
# (default=T:).
#
TMPDIR=T:

# Directory in which to (initially) look for ACE source files.
#
# If not specified, defaults to directory in which AIDE is started from.
#
SRCDIR=ACE:prgs

# Directory in which to put built executables and icons.
#
# If not specified, this defaults to the directory in which
# the source file which is being compiled resides.
#
#BLTDIR=ACE:Run

# Directory in which docs for the compiler, assembler & linker reside
# (default=ACE:docs).
#
DOCDIR=ACE:docs
---------------
```

In the example above, it is assumed that ACE: is a valid ASSIGN (see the
"Installation" file which comes with
the ACE distribution). Notice that lines commencing with a "#" are taken to
be comments.

AIDE has no built-in editor so you need to specify your favourite. I may
eventually write an editor, but my current feeling is that I can spend
my time more productively in other ways, given the plethora of text editors
available for the Amiga. The same is true for file viewers.

If anyone wishes to write a text editor for AIDE, feel free to let me know
about it. Likewise, if you think I should supply a particular FreeWare
editor with AIDE, I'll be glad to consider any suggestions you have.

Note that the editor you specify in AIDE.config must be capable of taking
command-line arguments (a file name). The same applies to the file viewer.

Once you have set the entries in AIDE.config to your satisfaction, you can

start AIDE from the Workbench. Note that in order to speed up the load time
of AIDE.config you may wish to copy it to say, AIDE.config.full and remove
all lines starting with '#' from AIDE.config.

1.4   Using AIDE

```
               Using AIDE
----------
```
There are six pull-down menus in AIDE: Project, Program, Compiler, Linker,
Utilities and Help. The Project menu is for general house-keeping tasks; the
Program menu is for editing, compiling and running ACE programs; the Compiler
menu is for setting compiler options; the Linker menu allows you to specify a
number of object files or scanned libraries to link with your main ACE program;
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

                  Program

                  Compiler

                   Linker

                  Utilities

                    Help
```

1.5   Project Menu

```
Project Menu
============
  New   - Invokes the editor with a blank slate.

  Open...   - Invokes the editor with a file name
        which you must specify via a file
        requester.

  View...   - Lets you view a file which is selected
        via a file requester.

  Delete... - Lets you delete a file specified via a
        file requester.

  Print...  - Lets you choose a file to print via a
        file requester.

  Execute...  - Allows an AmigaDOS command to be executed.
        The command and its arguments are entered
```

into an ACE input requester, a console
window is opened for any I/O associated
with the command and the latter is executed
if possible.

Spawn Shell - A NewShell process is started for more
comprehensive house-keeping than the
"Execute..." option is intended for.

About   - Brings up a message box telling you briefly
about AIDE.

Quit AIDE - Exits from AIDE.

1.6 Program

```
Program
=======
```
  Set Source... - Lets you specify the ACE source file
to work with via a file requester. No
other Program menu items are accessible
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 this option was
invoked (see also SRCDIR entry in AIDE.config).

If the file selected does not exist,
ACE will invoke the editor.

  Edit Source - Invokes the editor with the current
source file as its argument.

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

  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 requester.
Supply no arguments if none are required.

  Compile - Compiles the current program, putting
the target assembly source file into the
temporary working directory specified in
AIDE.config. No executable is produced.
This is useful if you just want to see
whether a program compiles and/or want
to inspect the assembly source file.

  Make Executable - "Makes" the current program, putting
all intermediate files and the final
executable program into the temporary

```
        working directory specified in AIDE.config.

     Build Application - Does the same as "Make Executable" but
        also copies the final executable from
        temporary storage to the same directory
        as the original source file or to the
        directory specified by the BLTDIR config
        variable. If an icon is requested, this
        is also copied.

Create Linkable Module  - Creates an object module which can later
        be linked with an 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.

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

    View Preprocessed
   Source  - Allows you to view the APP preprocessed
        source file.

   View Compiler
   Errors  - Lets you see the errors (if any) generated
        by the last compilation. This pertains to
        ACE errors only at this stage.
```

1.7 Compiler

```
            Compiler
========
  There are seven compiler options in this menu: Preprocess Source,
  Break Trapping, Assembly Comments, Create Icon, List Source Lines,
  Optimise Assembly and Window Trapping. The latter six correspond to
the compiler's command-line switches: b,c,i,l,O and w respectively.
        See Ace.Guide's "Compiler options" section for details of each
        of these.

  Since AIDE always uses ACE's -E switch to create an error file,
  this option is not available in this menu. Compilation errors are
  however, viewable from the Program menu's "
            View Compiler Errors
            "
  item.

  Items in this menu which have a check mark to their left are set.


  The "Optimise Assembly" option is set by default when AIDE starts.
```

"Preprocess Source" is also checked at startup. To prevent ACE
source code from being preprocessed by APP, uncheck this item.

Note that if you want AIDE to create an icon for an executable,
the file ACE:icons/exe.info MUST exist. This is required by ACE
as well (see Ace.Guide).

To save the current compiler options, select the last item in
the menu: "Save Options". Next time AIDE is run, the Compiler menu
will be restored accordingly.

1.8 Linker

```
Linker
======
```
  In addition to the normal assembled program file and the scanned
  libraries incorporated by the linker into the final executable,
  additional object files to be linked can be specified via this
  menu.

  Unless you are using external functions from other scanned
        libraries or modules (written in ACE, C or assembly etc), you may
        never have cause to use this facility.

  The list of extra modules to be linked is displayed as part of
  the menu. These new items cannot be selected and are for display
  purposes only.

  Add Module...   -   Adds an object file or scanned
        library to the list of modules to
        be linked.

  Remove Module... -   Removes an object file or scanned
        library from the list of modules to
        be linked.

  Remove All Modules - Removes all modules from the list
        of extra files to be linked.

1.9 Utilities

```
Utilities
=========
```
  Calculator   - Invokes the calculator of your
        choice as specified by CALCTOOL
        in AIDE.config.

  Create BMAP file * - Converts the FD file 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.

  AmigaBASIC to ASCII * - Converts a compressed AmigaBASIC
          file into an ASCII text file for
          use by ACE.

  * AIDE expects the utility to be in ACE:utils.

1.10 Help

Help
====
  This menu provides access to the documentation for the compiler,
  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.

1.11 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 re-compile 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.

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.

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 modify the modules list in the Linker menu, AIDE will recompile
everything. This is overkill really and a future version will limit this
to the link phase.

Making a change to any of the options in the Compiler menu will cause
previous makes to be ignored also.

In order to preprocess, compile, assemble and link a program, AIDE first

creates a shell script and then executes it, opening a shell in order to
do so.

Meanwhile, AIDE awaits a final message from the shell process via a PIPE
before continuing. In addition to some status messages which AIDE will
display, one of two final messages will be received.

```
   MakeDone
     or
   MakeAborted
```

If an error or user break occurs during preprocessing or compilation, the
shell process returns a "MakeAborted" message and AIDE acts accordingly,
by, for example, not running the program.

If an error or user break occurs during assembly or linking however, you
will be left staring at a shell prompt. The simplest way out of this is
to issue the following two commands at the shell prompt:

```
  QuitMake
  EndCli
```

one at a time. These will: (i) send a message back to AIDE telling it that
something has gone awry during assembly or linking; (ii) close the shell.
In case you are interested, "QuitMake" is an alias (for "MakeAborted") set
up by AIDE.

This is one area in which AIDE could be improved, but at least this solution
is workable.

1.12   Final Comments

```
Final Comments
--------------
```
As you can see, there is certainly scope for improvement and it is my hope
that AIDE will continue to get better over time.

Feel free to make any suggestions about AIDE and please report any bugs
to me. See Ace.Guide for details of how to contact me by e-mail, snail-mail
or telephone.

If you make any substantial changes to AIDE I'd appreciate knowing about
them, since I may be able to incorporate such modifications into the next
release. Naturally any contributions will be acknowledged.

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

programs.

Regards,
David Benn, Tasmania

12th October 1994