

**AStripI**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> AStripI	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		June 24, 2022
<i>SIGNATURE</i>		

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>AStripI</b>	<b>1</b>
1.1	AStripI.guide . . . . .	1
1.2	AStripI/Introduction . . . . .	1
1.3	AStripI/Read Me First!! . . . . .	2
1.4	AStripI/How To Use AStripI . . . . .	2
1.5	AStripI/Contacting The Author . . . . .	5
1.6	AStripI/About The Source . . . . .	5
1.7	AStripI/Notes . . . . .	6

---

# Chapter 1

## AStripI

### 1.1 AStripI.guide

```
AStripI v2.0
(05.09.1997)

by Cliff Earl
(cee@voyager.co.nz)

© 1997 Antix Software

--README FIRST--

Introduction

How To Use AStripI

About The Source

Notes

Contacting Me
```

### 1.2 AStripI/Introduction

Yep, I finally have started to program the OS quite a lot. The problem I have is that the C= includes are dead slow to use on my ECS A2000 68000. AStripI's the answer :^)

What AStripI does is optimize assembly language include files. As another side benefit, AStripI reduces the size of the include files a whole lot. Oh yes, AStripI is ONLY for assembly includes!!

Enjoy,  
Cliff Earl.

---

### 1.3 AStripI/Read Me First!!

Before you go off running AStripI on your includes you should read this document right through and understand what consequences there are of being ignorant of what AStripI will do to them!! I take no responsibility for any damage caused to your includes or your system!!!

This program is Copyright © 1996-1997 by Antix Software. You are not allowed to make a profit by selling this program and/or it's associated files. It may be distributed for free only.

People wishing to include this program on a CoverDisk or CD then they should send

me

a free copy of the CD or Magazine which contains this program.

### 1.4 AStripI/How To Use AStripI

In version 1.0 AStripI had to be run seperatly on every file. This was great if you only wanted to strip a single include file but not very efficient when stripping multiple files.

As of version 2.0 AStripI goes through all files and subdirs in the specified directory and does it's magic on each of them. AStripI will not overwrite your original include files, instead it will write it's own include file back to the same directory as the original one. it will prepend "a\_" at the start of the name, so the file "execbase.i" would be written to a new file called "a\_execbase.i". All the references inside the new file will be to other a\_includes and not the original ones.

To run AStripI just type it's name from CLI or Shell. In the case where you supply no options then it will be the same as typing "AStripI -d INCLUDE: -r 100 -w 120".

The options...

The -q option is used to suppress output by AStripI. It's sometimes nice to see what a program is doing eh?

The -d option is used to specify the directory where the include files you want stripped are present. Defaults to INCLUDE: if not present.

The -r option is used to specify the read buffer size. Every file that gets loaded will get loaded into this buffer. If the file is bigger than the buffer you will be notified (unless you have specified -q). The size is in KiloBytes. Defaults to 100 if not supplied.

---

The `-w` option is used to specify the size of the strip/write buffer. All `a_include` files will be generated and saved to and from this buffer. If not present will default to 128. NOTE: There is no checking if this buffer gets overrun!! so make sure you specify at least 28kb more than your biggest include file.

Now I'll show you what an unstripped file looks like and what a stripped file looks like. Here's (example only) an unstripped include file..

```

IFND CRAPTEST_I
CRAPTEST_I SET 1
**
** $VER: craptest.i 0.01 (28.5.92)
** Includes Release 40.13
**
** A test of AStripI's ability to greatly decrease the size of
** include files and make them HEAPS faster...
**
** (C) Copyright 1985-1993 Commodore-Amiga, Inc.
** All Rights Reserved
**

IFND EXEC_NODES_I
INCLUDE "exec/nodes.i"
ENDC ; EXEC_NODES_I

*-----

*
* Full featured list header
*
STRUCTURE LH,0
APTR LH_HEAD
APTR LH_TAIL
APTR LH_TAILPRED
UBYTE LH_TYPE
UBYTE LH_pad
LABEL LH_SIZE ;word aligned

*
* Minimal List Header - no type checking (best for most applications)
*
STRUCTURE MLH,0
APTR MLH_HEAD
APTR MLH_TAIL
APTR MLH_TAILPRED
LABEL MLH_SIZE ;longword aligned

*-----

;Prepare a list header for use
NEWLIST MACRO ; list
MOVE.L \1,LH_TAILPRED(\1)
ADDQ.L #4,\1 ;Get address of LH_TAIL

```

```

CLR.L    (\1)    ;Clear LH_TAIL
MOVE.L   \1,-(\1) ;Address of LH_TAIL to LH_HEAD
ENDM

BITDEF   CACR,EnableI,0    ;Enable instruction cache
BITDEF   CACR,FreezeI,1    ;Freeze instruction cache
BITDEF   CACR,ClearI,3     ;Clear instruction cache
BITDEF   CACR,IBE,4       ;Instruction burst enable

ENDC    ; CRAPTEST_I

```

You might notice that the normal include files are bloated with comments and usually contain extraneous space chars, most likely because the C= guys were all "C" programmers. Did you know that all those words like STRUCT, BITDEF, APTR, etc are actually all MACRO's? This is what makes your poor old assembler choke so much, especially on a 68000 Amiga. Now let's see the file that AStripI would make out of this...

```

        IFND    CRAPTEST_I
CRAPTEST_I
        IFND    EXEC_NODES_I
        INCLUDE "exec/a_nodes.i"
        ENDC
        RSRESET
LH_HEAD      rs.l  1
LH_TAIL      rs.l  1
LH_TAILPRED  rs.l  1
LH_TYPE      rs.b  1
LH_pad       rs.b  1
LH_SIZE      rs.b  0
        RSRESET
MLH_HEAD     rs.l  1
MLH_TAIL     rs.l  1
MLH_TAILPRED rs.l  1
MLH_SIZE     rs.b  0
CACRB_EnableI = 0
CACRF_EnableI = 1<<0
CACRB_FreezeI = 1
CACRF_FreezeI = 1<<1
CACRB_ClearI  = 3
CACRF_ClearI  = 1<<3
CACRB_IBE     = 4
CACRF_IBE     = 1<<4
        ENDC

```

See how much smaller our new file is? It's only about a third of the original's size. The file could be made even smaller by not padding the lines out to 32 chars with tabs.

Now would be a good time to mention that if your assembler does not support rs.b, rs.w, and rs.l then you should delete AStripI right away, it's not for you..

Ok. You might be wondering where the NEWLIST macro went? Well AStripI does not like those macros so it deletes them. Some people might be aghast that the lovely C= macros are cast into the void but since the C= macros don't get used in any of the generated code they just get trashed. If you want to run AStripI on your own custom include files you better make sure you aren't going to need the macro's if they contain any. As a practise you should keep your macros in their own files like "dos.macros", or "exec.macros". So remember.. ALL MACROS WILL BE REMOVED!!!!

So you should be able to deduce from the example above if you want to use AStripI on your includes.

- AStripI will skip any files that do not end with .i. So AStripI would strip the file "mycoolinclude.i" but would ignore the file "mycoolsource.s".
- AStripI will skip any file starting with the longword \$f9faf9fa. That's the identifier of an ASM-One source file. so all those files will be safe :^)
- AStripI will skip any file that begins with the 4 byte character string ";ASM". This is an easy way for you to protect any macro files you might have.

Please note that I only use the term protect very loosely as AStripI does not actually overwrite any files that it strips.

## 1.5 AStripI/Contacting The Author

If you have suggestions, comments, flames, gifts, or just wanna chat about ASM programming then contact me..

E-Mail: [cee@voyager.co.nz](mailto:cee@voyager.co.nz)

FIDO:

Snail: Cliff Earl,  
P.O Box 159,  
Otaki,  
New Zealand.

Phone: +64 6 364 5667

## 1.6 AStripI/About The Source

The full source to AStripI is included in the archive. For it to work you will need to copy the files from the include drawer into your own include drawer.

---



The source will assemble in ASM-One and is not tested in DevPac or any other assembler.

The source is NOT commented and is a bit messy but any programmer worth his salt should easily be able to get a hold of what's going on and be able to modify the program with relative ease.

As mentioned, the source is messy. This was only done as a quick hack. If you improve the source you should let me know.

## 1.7 AStripI/Notes

Ok some quick notes now as they roll out of my head...

- Tested on the 3.1 includes and it works fine..
  - AStripI runs from CLI or SHELL ONLY!!!
  - the best way to use AStripI is to unarchive the original C=includes into RAM: or a TEMP: if you have only a little ram. Then run AStripI on the includes there. Remember, AStripI will not modify the original include file but will make another new one prepended with "a\_".
  - Can corrupt your dos/dos\_lib.i file. That include uses some corny "LIBENT" macro to do it's \_LVO's. It's the only one to my knowledge that does this and IMHO it should die :^) All lvo's should live in the lvo dir anyway.
-