

**ASM-One\_V1.3x**

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

	<i>TITLE :</i> ASM-One_V1.3x	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		July 10, 2022
		<i>SIGNATURE</i>

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

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ASM-One_V1.3x</b>	<b>1</b>
1.1	ASM-One V1.4x Manual . . . . .	1
1.2	Copyright notice . . . . .	2
1.3	About this Manual . . . . .	2
1.4	What is ASM-One . . . . .	3
1.5	Bugs and ideas . . . . .	4
1.6	Starting with ASM-One . . . . .	4
1.7	ASM-One's MENU Structure . . . . .	6
1.8	General MENU Structure . . . . .	6
1.9	Editor MENU Structure . . . . .	9
1.10	Debug MENU Structure . . . . .	9
1.11	Monitor MENU Structure . . . . .	10
1.12	Preference MENU Structure . . . . .	10
1.13	DLC Syntax . . . . .	10
1.14	DLCs . . . . .	11
1.15	The '! ' DLC . . . . .	14
1.16	The '!! ' DLC . . . . .	14
1.17	The '# ' DLC . . . . .	15
1.18	The '= ' DLC . . . . .	15
1.19	The '=C' DLC . . . . .	16
1.20	The '=A' DLC . . . . .	16
1.21	The '=F' DLC . . . . .	17
1.22	The '=M' DLC . . . . .	18
1.23	The '=P' DLC . . . . .	18
1.24	The '=R' DLC . . . . .	19
1.25	The '=S' DLC . . . . .	19
1.26	The '>' DLC . . . . .	20
1.27	The '?' DLC . . . . .	20
1.28	The '[' DLC . . . . .	21
1.29	The '@A' DLC . . . . .	22

---

1.30	The '@B' DLC	22
1.31	The '@D' DLC	23
1.32	The '@H' DLC	24
1.33	The '@N' DLC	24
1.34	The 'A' DLC	25
1.35	The 'AC' DLC	25
1.36	The 'AD' DLC	26
1.37	The 'AO' DLC	26
1.38	The 'AS' DLC	27
1.39	The 'B' DLC	28
1.40	The 'BS' DLC	28
1.41	The 'C' DLC	29
1.42	The 'CC' DLC	29
1.43	The 'CD' DLC	30
1.44	The 'CS' DLC	31
1.45	The 'D' DLC	31
1.46	The 'E' DLC	32
1.47	The 'EL' DLC	32
1.48	The 'F' DLC	33
1.49	The 'G' DLC	34
1.50	The 'H' DLC	34
1.51	The 'I' DLC	34
1.52	The 'IB' DLC	35
1.53	The 'ID' DLC	35
1.54	The 'IH' DLC	36
1.55	The 'IN' DLC	36
1.56	The 'IS' DLC	37
1.57	The 'J' DLC	37
1.58	The 'K' DLC	38
1.59	The 'L' DLC	38
1.60	The 'M' DLC	39
1.61	The 'N' DLC	39
1.62	The 'O' DLC	40
1.63	The 'P' DLC	40
1.64	The 'PS' DLC	40
1.65	The 'Q' DLC	41
1.66	The 'R' DLC	42
1.67	The 'RB' DLC	43
1.68	The 'RE' DLC	43

---

---

1.69	The 'RO' DLC	44
1.70	The 'RS' DLC	45
1.71	The 'RT' DLC	45
1.72	The 'S' DLC	46
1.73	The 'T' DLC	47
1.74	The 'U' DLC	47
1.75	The 'UA' DLC	48
1.76	The 'V' DLC	48
1.77	The 'W' DLC	49
1.78	The 'WB' DLC	50
1.79	The 'WE' DLC	50
1.80	The 'WL' DLC	51
1.81	The 'WN' DLC	51
1.82	The 'WO' DLC	52
1.83	The 'WP' DLC	52
1.84	The 'WS' DLC	53
1.85	The 'WT' DLC	53
1.86	The 'X' DLC	54
1.87	The 'Y' DLC	55
1.88	The 'ZA' DLC	55
1.89	The 'ZB' DLC	56
1.90	The 'ZF' DLC	56
1.91	The 'ZI' DLC	56
1.92	The 'ZS' DLC	57
1.93	Warnings & Errors	57
1.94	Errors	58
1.95	Errors	59
1.96	Errors	60
1.97	Errors	61
1.98	Warnings & Errors	62
1.99	Workspace Memory Full	62
1.100	Address Register Byte/Logic	63
1.101	Address Register Expected	63
1.102	Comma Expected	64
1.103	Data Register Expected	64
1.104	Double Symbol	64
1.105	Unexpected End of File	65
1.106	End of File	65
1.107	User made FAIL	65

---

---

1.108Illegal Command . . . . .	65
1.109Illegal Address Size . . . . .	66
1.110Illegal Operand . . . . .	66
1.111Illegal Operator . . . . .	66
1.112Illegal Operator in BSS Area . . . . .	67
1.113Illegal Order . . . . .	67
1.114Illegal Register Size . . . . .	67
1.115Illegal Section Type . . . . .	68
1.116Illegal Size . . . . .	68
1.117Illegal MACRO Definition . . . . .	68
1.118Immediate Operand Expected . . . . .	69
1.119Include Jam . . . . .	69
1.120MACRO Overflow . . . . .	69
1.121Invalid Addressing Mode . . . . .	69
1.122LOAD without ORG . . . . .	70
1.123Missing Quote . . . . .	70
1.124Conditional Overflow . . . . .	70
1.125No Operand Space Allowed . . . . .	71
1.126No Constant/Label . . . . .	71
1.127Not in MACRO . . . . .	71
1.128Out of Range 0/3/4/5/6/7/8/16/32 bit . . . . .	71
1.129Relative Mode Error . . . . .	72
1.130Reserved Word . . . . .	72
1.131Closing Parenthesis Expected . . . . .	72
1.132SECTION Overflow . . . . .	72
1.133String Expected . . . . .	73
1.134Undefined Symbol . . . . .	73
1.135Register Expected . . . . .	73
1.136Word at Odd Address . . . . .	74
1.137Not a Local Area . . . . .	74
1.138Code Moved During Pass 2.. . . . .	74
1.139Bcc.B Out of Range in MACRO . . . . .	74
1.140Out od Range (20 to 100) . . . . .	75
1.141Out od Range (60 to 132) . . . . .	75
1.142Include Overflow . . . . .	75
1.143Linker Limitation . . . . .	75
1.144Repeat Overflow . . . . .	76
1.145Not in Repeat Area . . . . .	76
1.146Double Definition . . . . .	76

---

---

1.147Relocation made to EMPTY Section . . . . .	76
1.148File Error . . . . .	77
1.149No Files . . . . .	77
1.150No Object . . . . .	77
1.151No File Space . . . . .	78
1.152Printer Device Missing . . . . .	78
1.153Not Done . . . . .	78
1.154Illegal Path . . . . .	78
1.155Illegal Device . . . . .	79
1.156Write Protected . . . . .	79
1.157No Disk in Drive . . . . .	79
1.158Illegal Option !! . . . . .	80
1.159REM without EREM . . . . .	80
1.160TEXT without ETEXT . . . . .	80
1.161Illegal Xn Scale Size . . . . .	80
1.162Field OFFSET Expected: {OFFSET:width} . . . . .	80
1.163Missing Brace . . . . .	81
1.164Colon Expected . . . . .	81
1.165Missing Bracket . . . . .	81
1.166FPU needed for Operation !! . . . . .	82
1.167To Many Watchpoints (maximal 8) . . . . .	82
1.168Illegal Source, not Activated !! . . . . .	82
1.169No Valid Memory Directory Present . . . . .	83
1.170Auto Command Overflow (more than 256 characters) . . . . .	83
1.171End should be Higher than Start ! . . . . .	83
1.172Illegal Source Number . . . . .	84
1.173Including Empty Source ? . . . . .	84
1.174Include Source Jam . . . . .	84
1.175Unknown Conversion Mode: should be RB or RN . . . . .	84
1.176Unknown CMAP Place: should be (B)EFORE,(A)FTER or (N)ONE . . . . .	85
1.177Unknown CMAP Mode: should be AGA or ECS . . . . .	85
1.178Trying to Include NON IFF File . . . . .	85
1.179IFF File is Not a ILBM File . . . . .	86
1.180Can't handle a BODY before BMHD Chunk . . . . .	86
1.181Illegal Floating Point Size: should be B, W, L, Q, S, D, X or P . . . . .	86
1.182Illegal Size for Data Register: should be B, W, L or S . . . . .	86
1.183Bcc.W Out of Range in MACRO . . . . .	87
1.184Floating Point Register Expected . . . . .	87
1.185This is not an ASM-One Project File . . . . .	87

---

1.186Bitfield can't be Out of 32 Bit Range . . . . .	87
1.187Illegal CMAP Mode Error: expecting 12, 24, CE or CA . . . . .	88
1.188Illegal Color Offset (Bank) Specified . . . . .	88
1.189IFF Color(s) Out of Range . . . . .	88
1.190To Many Errors . . . . .	88
1.191Invalid Function Code: only SFC, DFC, Dn or #<data> Allowed . . . . .	89
1.192Address Register Indirect Mode Expected, for Return Address Register . . . . .	89
1.193Address Register Direct Mode Expected, for Return Address Register . . . . .	89
1.194Level Out of 3 Bit Range . . . . .	90
1.195Mask Out of 3 Bit Range . . . . .	90
1.196Mask Out of 4 Bit Range . . . . .	90
1.197Function Code Out of 3 Bit Range . . . . .	91
1.198Function Code Out of 4 Bit Range . . . . .	91
1.199Insuficient Memory to change Source . . . . .	91
1.200MMU Register Expected as 1st Operand . . . . .	91
1.201MMU Register Expected as 2nd Operand . . . . .	92
1.202Invalid MMU Register as 1st Operand: CAL, VAL or SCC Expected . . . . .	92
1.203Invalid MMU Register as 2nd Operand: CAL, VAL or SCC Expected . . . . .	92
1.204Missing START address or Label !! . . . . .	92
1.205Incorrect Index Register Specified. Only An, Dn or SP allowed . . . . .	93
1.206PC/ZPC cann't be used as Index Register. Only An, Dn or SP allowed . . . . .	93
1.207We haven't come up with a nice text for this error Error !!!! . . . . .	93
1.208The use of ZPC is not allowed in this addressing mode !! . . . . .	93
1.209Field Width Expected: {offset:WIDTH} . . . . .	94
1.210Field Width can't be ZERO . . . . .	94
1.211Warnings . . . . .	94
1.212>> Warning <<, Value Signed Extended to Longword . . . . .	95
1.213>> Warning << 68010++ Command Used . . . . .	95
1.214>> Warning << 68020++ Command Used . . . . .	96
1.215>> Warning << 68030++ Command Used . . . . .	96
1.216>> Warning << 68040++ Command Used . . . . .	96
1.217>> Warning << 68010 Specific Command Used . . . . .	96
1.218>> Warning << 68020 Specific Command Used . . . . .	97
1.219>> Warning << 68030 Specific Command Used . . . . .	97
1.220>> Warning << 68040 Specific Command Used . . . . .	97
1.221>> Warning << 68060 Specific Command Used . . . . .	97
1.222>> Warning << 68020/030/040 Specific Command Used . . . . .	98
1.223 Warning << 68020/030 Specific Command Used . . . . .	98
1.224>> Warning << MOVEP not supported by the 68060 . . . . .	98



---

1.225>> Warning << 68030/68851 Command Used . . . . .	99
1.226>> Warning << 68851 Command Used . . . . .	99
1.227>> Warning << 68881/68882 Command Used . . . . .	99
1.228>> Warning << PFLUSHA Assembled with 68030/68851 . . . . .	99
1.229>> Warning << PFLUSHA Assembled with 68040 . . . . .	100
1.230 Register Names & Private Symbols . . . . .	100
1.231 Valid Register Tabel . . . . .	101
1.232 ASM-One.Pref File . . . . .	102
1.233 M68000 Family Instruction Set . . . . .	109
1.234 M68000 Family Instruction Set . . . . .	109
1.235 M68000 Family Instruction Set . . . . .	110
1.236 M68000 Family Instruction Set . . . . .	111
1.237 M68000 Family Instruction Set . . . . .	112
1.238 M68000 Family Mnemonic Description . . . . .	113
1.239 M68000 Family Mnemonic Description . . . . .	114
1.240 M68000 Family Mnemonic Description . . . . .	115
1.241 M68000 Family Mnemonic Description . . . . .	116
1.242 M68000 Family Mnemonic Description . . . . .	116
1.243 M68000 Family Mnemonic Description . . . . .	117
1.244 Contacting the Author . . . . .	118
1.245 Thanks to.... . . . .	118

---

# Chapter 1

## ASM-One\_V1.3x

### 1.1 ASM-One V1.4x Manual

The ASM-One V1.4x Manual

The COMPLETE guide to ASM-One

Copyright (C) 1999-2000 By Boussh of TFA

THIRD DRAFT

Copyright

About

What is ASM-One

Bugs and ideas

Starting with ASM-One

ASM-One's MENU Structure

DLCs

Directive

Register Names & Private Symbols

ASM-One.Pref File

Warnings & Errors

-- Advanced Techniques --

Conditional Branches with IF

Building and Using MACRO's

-- MC68000 Family Information --

---

M68000 Family Instruction Set

M68000 Family Mnemonic Description  
M68000 Family Addressing Modes

Contacting the Autor

Thanks to....

## 1.2 Copyright notice

Copyright  
\*\*\*\*\*

Copyright Notice  
-----

This document is Copyright (C) 1999 by Boushh of TFA.

Disclaimer  
-----

This document is offered "as is", and the author will not be held responsible for any damage resulting from mistakes contained within, though the best effort has been made to ensure that all the information is complete and correct.

Distribution  
-----

This is only Copyrighted in order to protect the integrity of this document and to ensure its free availability to all. You may not directly profit from the distribution of this document. You may not put it in a disk collection without prior permission from the author. You may freely and for free distribute it across any medium.

There is no shareware registration fee since this is "freeware".

About  
-----

This document was written on an Amiga 3000, with an 1,5 Gb hd and 12MB ram. This guide is created with CygnusEd Professional V4.0.

## 1.3 About this Manual

About this Manual  
=====

---

Since the first release of ASM-One (V1.01, programmed by Rune Gram-Madsen), TFA added many new instructions, commands, directives etc. etc.

V1.01 probably had a printed manual (German). V1.02 (released by Venture) had the printed manual as an ASCII file on disk, as well as a small english version.

Up until version V1.15 of ASM-One, those manuals did cover most of ASM-One's functions. But after V1.15, there were more and more differences after every new release of ASM-One. Those manuals were therefore left out.

Around version V1.20 I (Boushh) translated the original German manual to Dutch (mainly for my convenience). But this was on paper and the programs used to edit the text, were non-Amiga programs, so it was no good idea to include it into a new release.

Since then, many ideas have been sent to use. And we discovered that quite a lot of these ideas were already built into ASM-One. But nobody knew, because there were not documented. But also new users had absolutely no idea what ASM-One could do.

Therefore I decided to finally put all the features of ASM-One in one AmigaGuide. And you are looking at it B-)

I did my best to make it as complete as possible, but if you have any suggestions, they are welcome.

Boushh  
(remco.weg@endemol.nl)

P.S. ASM-One has been cloned quite much. Many of these clones (like Trash'm-One (Deftronic), AsmPro (Genetic), etc, etc, will have most of ASM-One's functionality. So, if you are a user of those clones, you can use the manual as well B-)

## 1.4 What is ASM-One

What is ASM-One ?  
=====

ASM-One is an integrated assembler developing environment. Where other assemblers (commercial or PD) offer some integration, or only CLI execution, ASM-One offers everything in one package.

ASM-One consists of:

- An Editor
- An Assembler
- A Debugger
- A Monitor

ASM-One gives you the possibility to develop your software without browsing through several programs. Everything you need is available right at your hands.

## 1.5 Bugs and ideas

### Bugs & Ideas

-----

We maintain webpages with the latest fixes.

It's located at:

<http://www.euronet.nl/users/jdm/>

Before mailing me a bug or an idea, READ this manual and check out the webpages !!!!

### Mailing Bugs & Ideas

-----

If you mail Ideas:

- Tell exactly as you can how it should work.
- If possible, include sourcecode of your idea (WB/KS 2.x compatible).

If you mail bugs:

- Be as detailed as possible.
- Write down what you did, AND what the result was.
- Include sample sourcecode, generating the error (if possible).
- Include your system configuration (get that with tools/Showconfig).
- Include your ASM-One.Pref file.

Boushh  
(remco.weg@endemol.nl)

## 1.6 Starting with ASM-One

### Starting ASM-One

-----

Here's is an example of a first session with ASM-One:

Starting  
-----

Start ASM-One from the CLI (a Shell) or by double clicking on it's icon.

---

## First Steps

---

After ASM-One is started, you will get the following line:

```
ALLOCATE Fast/Chip/Publ/Abs>
```

Here you can select in which kind of memory your source will be located.

```
F = FastMem
C = ChipMem
P = PublicMem (the largest available memory block)
A = Absolute
```

After you have selected where your source should be allocated, you will have to choose how large the allocated block of memory should be:

```
WORKSPACE (Max.7680) KB>
```

After Max. is the largest available block of memory available.

Now you are ready to rock & roll B-)

## Loading a source

---

Type R after the >, and a requester will show up.

Select the source you want to work with.

## Going to the Editor

---

And press the ESC key to go into the Editor

(If you don't have a source to work with, press the ESC key to go into the Editor and make your own source.)

## Assembling a source

---

Press ESC to go to the > again.

Type A (Assemble), and ASM-One will assemble your source.

If any errors occur, press ESC and go into the Editor to correct them.

## Writing an object

---

After you have assembled a source, you have an object (executable file). You can save the object by typing WO (Write Object) at the >.

If you assembled a source that's not executable, type WB (Write Binary) at the >.

---

If you have made changes to your source, type U (Update) at the >.

If you made a new source, type W (Write) at the >.

## 1.7 ASM-One's MENU Structure

### ASM-One's MENU Structure

-----

ASM-One has different menu's for every part of ASM-One:

```

General MENU Structure
Editor MENU Structure
Debug MENU Structure
Monitor MENU Structure
Preference MENU Structure

```

## 1.8 General MENU Structure

### General MENU Structure

-----

The 'General MENU Structure' is available when you are at the CLI of ASM-One. You are there when you can type commands after the >.

This menu is divided into three parts:

```

Project
  Assembler
  Commands
  Project Info  =P
    Assemble    »
  Editor        »
  Old           O
  Assemble     A
  Jump Top     T
  Read         »
  Optimize     O

```

```

Jump Bottom      B

Environment      RE

Check Only      AC

Search          S

Source          R

Object Info =
    Zap Line(s)  ZL

Binary          RB
-----
Print Line(s)   P

Object          RO
  Editor        E
Extend Labels   EL
Write           »   Debugger      D
Memory         »

Environment     WE
  Monitor      M
Edit          M

Source +Mark    W

Preferences     »

Assemble       D

Source -Mark    WN
  Environment   [
HEX Dump       H

Binary         WB
  Assembler    ]
ASCII         N

Object         WO
  AGA Guide    =   -----

Link          WL

DisassembleLine @D

Preferences    WP

Assemble      @A

Insert        I

HEX Line      @H

```

---



```
Update          »
ASCII Line      @N
Source          U
Binary Line     @B
Project         UA
-----
Zap Auto-Alloc ZA
Search Memory   S
Zap File        ZF
Fill Memory     F
Zap Include     ZI
Copy Memory     C
Zap Source      ZS
Compare Memory  Q
-----
Create Sinus    CS
Add WorkMem     =M
Insert          »
-----
Disassembly     ID
About           #
HEX Dump        IH
-----
ASCII Dump      IN
Quit/Restart    !
Binary Dump     IB
Quick Quit      !!
Create Sinus    IS
Assemble        »
Assemble        A
Memory          @A
Optimize        AO
Debug           AD
```

---

Symbols	=S
Parameters Set	PS
Monitor	»
Jump	J
Go	G
Step	K
Status	X
Zap BPS	ZB
Disk	»
Read Sector	RS
Read Track	RT
-----	
Write Sector	WS
Write Track	WT
-----	
Calc. Checksum	CC
-----	
Extern	E
Output	>
Calculate	?
Calculate Float	[
Custom Registers	=R

## 1.9 Editor MENU Structure

Editor MENU Structure

-----

The 'Editor MENU Structure' is available when you are in the Editor.

This menu is divided into three parts:

## 1.10 Debug MENU Structure

### Debug MENU Structure

---

The 'Debug MENU Structure' is available when you are in the Debugger.

This menu is divided into two parts:

## 1.11 Monitor MENU Structure

### Monitor MENU Structure

---

The 'Monitor MENU Structure' is available when you are in the Monitor.

This menu is divided into two parts:

## 1.12 Preference MENU Structure

### Preference MENU Structure

---

The 'Preference MENU Structure' is available when you are in one of the Preference windows.

This menu is divided into two parts:

## 1.13 DLC Syntax

### DLC Sytnax

---

#### NAME

<COMMAND> [short function description]

#### SYNTAX

- <command syntax>

[KEY-SHORTCUT]

#### FUNCTION

- <function description>

[EXAMPLE]

[INPUT]

[RESULT]

---

[REMARKS]

[SEE ALSO]

FROM VERSION  
- <version number>

[BUGS]

## 1.14 DLCs

DLCs

====

What means DLCs ?  
-----

DLCs is short for: "Direct Line Command(s)". You issue a DLC at the Command Line Interpreter (CLI) of ASM-One. This CLI is the place where you normally start. It begins with the ">".

This part will cover all the DLCs from ASM-One in alphabetical order. Using the following

SYNTAX

.

So, here we go B-)

!

B

P

WS

!!

C

PS

WT

#

CC

Q

X

=

---

CS

R

Y

=C

D

RB

ZA

=F

E

RE

ZB

=M

EL

RO

ZF

=P

F

ZI

=R

G

RS

ZL

=S

H

RT

ZS

>

I

S

?

IB

T

Leftovers:

[

ID

U

@A

IH

UA

CD

@B

IN

V

=A

@D

IS

W

BS

@H

J

WB

AS

@N

K

WE

A

L

WL

AC  
M  
WN  
AD  
N  
WO  
AO  
O  
WP

## 1.15 The '! ' DLC

NAME

! - Quit/Restart ASM-One

SYNTAX

!

FUNCTION

This will bring up the QUIT-requester (when reqtools is enabled). Here you can choose to QUIT ASM-One, to RESTART ASM-One or to cancel your action.

If you choose RESTART, ASM-One will reset itself and start like if you started ASM-One for the first time.

RESULT

ASM-One will be quited/restarted.

SEE ALSO

The '!!!' DLC  
FROM VERSION

V1.01

## 1.16 The '!!!' DLC

NAME

!!

SYNTAX

---

!! - Quick Exit

FUNCTION

This will end ASM-One WITHOUT ANY REQUESTER.

RESULT

ASM-One will be quited

REMARKS

Note that nothing will be saved. If you have still unsaved sources, or are unsure, use the '!' DLC.

SEE ALSO

The '!' DLC  
FROM VERSION

V1.25

## 1.17 The '#' DLC

NAME

# - About...

SYNTAX

#

FUNCTION

This will show up the ASM-One 'About...' requester.

RESULT

When ReqTools enabled, you will get the 'About...' requester.

FROM VERSION

V1.01

## 1.18 The '=' DLC

NAME  
= - Object Info

SYNTAX

=

FUNCTION

To show the memory usage of your current source.

RESULT

Will show the memory usage of your current source.

SEE ALSO

---



The '=M' DLC  
FROM VERSION

V1.01

## 1.19 The '=C' DLC

### NAME

=C - Change the colors of ASM-One

### SYNTAX

=C

### FUNCTION

When Regtools enabled, you will get a Color Requester with the following functions:

- Copy - Will copy the selected color to another place
- Swap - Will swap the selected color with another color
- Spread - Will start to spread colors from the selected color up till the last color using the selected color as start
- Ok - Will install the new colors, but WILL NOT save them
- Undo - Will reset the colors as they where, but you will stay in the requester
- Cancel - Will reset the colors as they where before the =C DLC and quit the requester

### RESULT

Changed or NOT changed colors.

### REMARKS

Clicking on the requester's CLOSE gadget will have the same effect as CANCEL  
DO NOT FORGET to save your preferences if you have picked new colors (or the next time you start ASM-One, you will have the same old boring colors).

### FROM VERSION

V1.15 (approximally)

## 1.20 The '=A' DLC

### NAME

=A - Show the 5 sections of ASM-One

### SYNTAX

=A

### FUNCTION

So you can see where ASM-One is located in memory.

### RESULT

Something like this:

ASM-One Location	Start	End	Total
	-----	-----	-----
1st Code section:	072B1628	072B18D4	684
2nd Code section:	073289B0	0733363B	44171
1st Data section:	07309AF8	07326902	118282
2nd Data section:	07326910	073289A2	8338
BSS Data section:	07333648	0733AE41	30713

(V1.3x values !!!)

FROM VERSION  
V1.20

## 1.21 The '=F' DLC

NAME

=F - Change the Font of ASM-One

SYNTAX  
=F

FUNCTION

When Reqtools enabled, you will get a Font Requester with the following functions:

Ok - Will install the new font, but WILL NOT save it  
Cancel - Will cancel the Font Requester

RESULT

Changed or NOT changed font.

REMARKS

ASM-One can ONLY handle 8\*8 fixed fonts. Default the requester will only show 8 pixels height fonts. But the font maybe to width for ASM-One. The only way to check this is to select the topaz.font, and remember where te last character was. And compare that position with the font you want to select.

Changes will be made directly, accept for the menu's.

Save the changes with  
Write Prefereces  
or by choosing  
SAVE within the Preferences.

FROM VERSION  
V1.30

BUGS

If you select a font, and the requester says: 'Couldn't open Font'. Than DON'T SELECT it. If you do, ASM-One will use the default font (currently: topaz.font). But '=F' will no longer work.

## 1.22 The '=M' DLC

### NAME

=M - Add work memory

### SYNTAX

=M

### FUNCTION

Gives you the possibility to add workspace.

### RESULT

More workspace.

### REMARKS

No mater if you add or don't add memory, you will always get a nice overview of all memory your source uses:

	Start	End	Total
	-----	-----	-----
Workspace	: 004453C0	0045E3C0	102400
Source	: 004453C1	0044698B	5578
Label Pointers	: 00446B8E	0044BB8E	20480
Label	: 0044BB8E	0045E35C	75726
IncMem	: -----	-----	189628

If you have not assembled your source, you will get a shorter overview.

### SEE ALSO

The '=' DLC  
FROM VERSION

V1.01

## 1.23 The '=P' DLC

### NAME

=P - Show Project Info

### SYNTAX

=P

### FUNCTION

When Reqtools enabled, it will show up an requester with the current status of your Project.

### RESULT

A Requester.

FROM VERSION  
V1.28

## 1.24 The '=R' DLC

NAME

=R - Show Register Info

SYNTAX

=R [registernumber]

FUNCTION

Will show info on the requested Amiga Register.

INPUT

Nothing, or a [registernumber]. The [registernumber] should be the last three numbers of the hexadecimal address of the register

EXAMPLE

```
>=R 1EC
1EC   W   A           VRam (UHRES) bitplane pointer (hi 5 bits)..:

BPLHPTH - UHRES (VRAM) bit plane pntr (high 5 bits)
BPLHPTL - UHRES (VRAM) bit plane pntr (low 15 bits)
```

When UHRES is enabled, this pointer comes out on the 2nd 'free' cycle after the start of each horizontal line. It's modulo is added every time it comes out. 'free' means priority above the copper and below the fixed stuff (audio, sprites...).

BPLHDAT comes out as an identifier on the RGA lines when the pointer address is valid so that external detectors can use this to do the special cycle for the VRAMs, The SHRHDAT gets the first and third free cycles.

RESULT

If no [registernumber] was given, an explanation of the used symbols and a complete list of ALL Amiga registers.  
Otherwise the requested register (if the number was valid).

FROM VERSION  
V1.--

## 1.25 The '=S' DLC

NAME

=S - Show Symbols

SYNTAX

=S[characters]

---

**FUNCTION**

Will show ALL currently known Symbols.

You can also specify the first 2 characters of the symbol you are looking for at [characters].

That will list the symbols from that point.

**RESULT**

List with Symbols (= Labels).

**REMARK**

Think about UpperCase=LowerCase !!!

**FROM VERSION**

V1.01

Extra [characters option] from V1.16 and up

## 1.26 The '>' DLC

**NAME**

> - Redirect Output

**SYNTAX**

>[filename]

**FUNCTION**

Will redirect the output on the commandline to a file.

If no filename was given, and reqtools is enabled, a file requester will show up to specify the filename.

Choose cancel if you don't want to redirect the output any longer.

**RESULT**

File with output from ASM-One

**FROM VERSION**

V1.01

**BUGS**

The file contains a lot of trash characters.

## 1.27 The '?' DLC

**NAME**

? - Calculate

**SYNTAX**

?[value(s)]

---

## FUNCTION

Will convert between HEX, ASCII, Octal and binair. And will calculate between those formats.

## RESULT

An answer.

## FROM VERSION

V1.01

## 1.28 The '[' DLC

## NAME

[ - Calculate Float

## SYNTAX

[[value(s)]

## FUNCTION

Will calculate floats.

## EXAMPLE

```
> [.s383.3827*56.54
  2.16764579 E 4  $4D4EB90A.S
>
```

```
> [.x383.3827*56.54
  2.16764579 E 4  $401A0000CEB90A2947AE2000.X
>
```

```
> [2/3
  6.666666666>E-1  $401C00009EF21AAAAAAAAA.X
  |
  +- This means the number continues untill infinity.
```

```
> [2/3
  6.666666667\ensuremath{\pm}E-1  $401C00009EF21AAAAAAAA000.X
  |
  +- This means the number is inexact.
```

## RESULT

An answer B-)

## REMARKS

You can specify the the size (.S, .D, .X, .P), this will have no effect on the calculation, but on the hexadecimal output. Extended Precision (.X) is default.

The inexact and infinity flags only work at the commandline !!  
 Since during debugging ASM-One can not recognize which FPU register caused the inexact/infinity flag it will ignore them !!!

Entering a Hexadecimal number, you will have to enter the

FLOATING-POINT representation, NOT the INTEGER representation !!

FROM VERSION  
V1.20

## 1.29 The '@A' DLC

NAME

@A - Edit Memory

SYNTAX

@A[address/label]

FUNCTION

Will enable you to edit the memory at the adres after '@A' in hexidecimal.

EXAMPLE

Here's an example:

```
>@A$400
00000400 4e75
00000400
```

INPUT

Optional address or label, otherwise ASM-One will start at address \$00000000, or after the last address shown with @B, @D, @H, @N, D, H or N.

RESULT

Unknown at the moment

REMARKS

With [ESC] you end the session.

FROM VERSION  
V1.01

## 1.30 The '@B' DLC

NAME

@B - Show lines in binary

SYNTAX

@B[address/label]

FUNCTION

Will show 8 lines starting from address/label in binary.

EXAMPLE

```
>@Bstart
```

---

```

004484F4 %01110000
004484F5 %00000000
004484F6 %01001110
004484F7 %01110101
004484F8 %00010010
004484F9 %00110100
004484FA %01010110
004484FB %01111000

```

**INPUT**

Address or label.

**RESULT**

Eight binary lines.

**FROM VERSION**

V1.01

## 1.31 The '@D' DLC

**NAME**

@D - Show disassembled lines

**SYNTAX**

@D[address/label]

**FUNCTION**

Will show 12 disassembled lines starting from address/label.

**EXAMPLE**

```

>@Dstart
004621F4 23CF004623DC      MOVE.L    A7,$004623DC
004621FA 2C780004      MOVE.L    $0004.W,A6
004621FE 23CE004623D0      MOVE.L    A6,$004623D0
00462204 93C9          SUBA.L    A1,A1
00462206 4EAEFEDA      JSR      -294(A6)
0046220A 2840          MOVE.L    D0,A4
0046220C 6100019C      BSR.W    $004623AA
00462210 4AAC00AC      TST.L    $00AC(A4)
00462214 6724          BEQ.B    $0046223A
00462216 2C79004623D8      MOVE.L    $004623D8,A6
0046221C 4EAEFFC4      JSR      -60(A6)
00462220 23C0004623E8      MOVE.L    D0,$004623E8

```

**INPUT**

Address or label.

**RESULT**

Twelve disassembled lines.

**FROM VERSION**

V1.01



## 1.32 The '@H' DLC

### NAME

@H - Show hexadecimal lines

### SYNTAX

@H[address/label]

### FUNCTION

Will show 8 hexadecimal lines starting from address/label.

### EXAMPLE

```
>@Hmenuname
004625CE 4D 65 6E 75 00 52 65 73 65 74 00 51 75 69 74 00 "Menu.Reset.Quit."
004625DE 47 30 00 47 31 00 47 32 00 53 6F 6D 65 20 47 61 "G0.G1.G2.Some Ga"
004625EE 64 67 65 74 73 00 4E 65 77 20 54 69 74 6C 65 00 "dgets.New Title."
004625FE 69 6E 74 75 69 74 69 6F 6E 2E 6C 69 62 72 61 72 "intuition.librar"
0046260E 79 00 64 6F 73 2E 6C 69 62 72 61 72 79 00 54 68 "y.dos.library.Th"
0046261E 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73 74 20 "is program must "
0046262E 62 65 20 72 75 6E 20 66 72 6F 6D 20 74 68 65 20 "be run from the "
0046263E 57 6F 72 6B 62 65 6E 63 68 0A 12 34 56 78 01 01 "Workbench..4Vx.."
```

### INPUT

Address or label.

If you use a brace ({} after the @H, you will get indirect memory access:

```
@h{4      will give you the hexdump starting at the ExecBase.
@h{.14    will give you the hexdump starting at the ExecBase in longwords
@h4       will give you the hexdump starting at adress 4
```

### RESULT

Eight hexadecimal lines.

### FROM VERSION

V1.01

Indirect option from V1.16 and up

## 1.33 The '@N' DLC

### NAME

@N - Show ASCII lines

### SYNTAX

@N[address/label]

### FUNCTION

Will show 12 ASCII lines starting from address/label.

### EXAMPLE

```
>@Nmenuname
```

```

004625CE "Menu.Reset.Quit.G0.G1.G2.Some Gadgets.New Title.intuition.librar"
0046260E "y.dos.library.This program must be run from the Workbench..4Vx.."
0046264E ".....$......6.....P.....V.....\.....f.....n.....x"
0046268E ".....\textdegree{ }.....`.....`..... 0.....<.....L.....R ←
.....^....."
004626CE ".f.....l.....z.....,.....`.....`.....`....."
0046270E "...(......:.....B.....\.....`.....`.....`.....p"
0046274E ".....H.....^.....b.....n......¶.....`....."
0046278E "....."

```

**INPUT**

Address or label.

**RESULT**

Twelve ASCII lines.

**FROM VERSION**

V1.01

## 1.34 The 'A' DLC

**NAME**

A - Assemble the source currently in the editor.

**SYNTAX**

A

**KEY-SHORTCUT**

Right-Amiga+Shift+A

**FUNCTION**

This will start the assembler. The source in the current editor will be assembled.

**RESULT**

Various errors. If you encounter any errors, look them up in the list of errors

**FROM VERSION**

V1.01

## 1.35 The 'AC' DLC

**NAME**

AC - Assemble and Check the source currently in the editor.

**SYNTAX**

AC

**FUNCTION**

Will only CHECK your source on syntax errors. But will NOT assemble the source.

ASM-One will ONLY do Pass 1, NOT Pass 2.

**RESULT**

Various errors. If you encounter any errors, look them up in the list of errors

**FROM VERSION**

V1.29

## 1.36 The 'AD' DLC

**NAME**

AD - Assemble and Debug

**SYNTAX**

AD

**KEY-SHORTCUT**

Right-Amiga+Shift+D

**FUNCTION**

This will first assemble the source currently in the editor, and will then start the debugger.

**RESULT**

Various errors. As with the 'A' DLC, you can encounter errors. If an error occurs, ASM-One will NOT go to the Debugger.

**FROM VERSION**

V1.01

**BUGS**

Some times ASM-One goes to the Debugger and will end up in the remark lines (those starting with ';' or '\*'). This will display 'End of Source reached !!' in the menubar. Just type 'AD' again.

**SEE ALSO**

The Source-Level Debugger

## 1.37 The 'AO' DLC

**NAME**

AO - Assemble and Optimize

**SYNTAX**

AO

**KEY-SHORTCUT**

Right-Amiga+Shift+O

---

**FUNCTION**

ASM-One will start assembling your source. And optimize it when needed.

ASM-One will only optimize branches from WORD to BYTE, and from BYTE to WORD.

When Optimizing from Byte to Word, ASM-One will ALWAYS start reassembling. Since locations of labels AFTER the optimization will be changed from those locations ASM-One got during it's Pass 1.

This also happens when you normally assemble a source. So the only advantage of AO is currently the optimization of WORD to BYTE.

An optimization will result in a CHANGED source !!

**RESULT**

An optimized piece of code B-)

**REMARKS**

ASM-One also optimizes CLR.L Dn -> MOVEQ #0,Dn during assembling.

ASM-One will also change CMP/ADD/SUB instructions to appropriate instructions (like CMP.x (An)+, (An)+ -> CPM.x (An)+, (An)+) during assembling.

But ASM-One WILL NOT change your source for these optimizations !! ONLY the assembled code....

Other optimizations are under construction. Including changing your source.

Optimizing can take some time if your source is big and a lot has to be optimized. ASM-One has to move your source the be able to add characters to it.

**FROM VERSION**

V1.01

## 1.38 The 'AS' DLC

**NAME**

AS - Activate Source

**SYNTAX**

AS[source-number]

**FUNCTION**

Activates another source (same as F1-F10).

**INPUTE**

[source-number], can be in the range of 0 - 9.

---

## RESULT

The specified source is activated.

## FROM VERSION

V1.25

## 1.39 The 'B' DLC

## NAME

B - Bottom of source

## SYNTAX

B

## FUNCTION

With this command you will go to the bottom of the source.

## RESULT

When you go to the editor again, you will be at the bottom of the source.

## REMARKS

When your source is longer than 65535 lines, ASM-One will not go to the bottom of your source. This is due to the fact that the number of lines is calculated in WORD size.

## FROM VERSION

V1.01

## 1.40 The 'BS' DLC

## NAME

BS - Bootblock Simulator

## SYNTAX

BS

## FUNCTION

Allows you to simulate a BootBlock.

## INPUT

BEG> Start of the CODE in your bootblock.

## RESULT

At address/label specified by BEG> the code from your bootblock will start.

In other words: ASM-One will start at that address/label. All other stuff in front of it will be forgotten.

## EXAMPLE

Boot:                   DC.B           "DOS",0

```

                dc.l      0          ; for checksum
                dc.l      880        ; root block
Bootstart:     move.l      ...
                lea       ...
                ...         ...
                rts

```

Would be assembled and started at Bootstart: (so not Boot: !!)

#### REMARKS

Also allows you to assemble source from a certain address/label

#### FROM VERSION

V1.25

## 1.41 The 'C' DLC

#### NAME

C - Copy Memory

#### SYNTAX

C[.size]

#### FUNCTION

Will copy memory from one place to another.

#### INPUT

```

BEG> Start address or label
END> End address or label
DEST> Destination address or label

```

Nothing, or a [.size]. The [.size] can be BYTE (.B), WORD (.W) or LONGWORD (.L).

#### RESULT

Copied memory.

#### REMARKS

Be very CAREFULL with this DLC. Make sure you have saved all your work or otherwise you could destroy it !!!!

#### FROM VERSION

V1.01

## 1.42 The 'CC' DLC

#### NAME

C - Calculate Checksum

#### SYNTAX

CC[drive-number]

**FUNCTION**

Will calculate the Checksum of a floppy disk.

This is needed if you changed the bootblock of a floppy disk.

**INPUT**

You may specify a Drive number. Legal numbers are:

0: DF0  
1: DF1  
2: DF2  
3: DF3

DF0 is default, so if you don't specify a Drive number, the Checksum of the disk in DF0 will be calculated.

**RESULT**

A calculated Checksum.

**REMARKS**

Disks NEED a legal Checksum, or the Amiga will not BOOT from them.

**FROM VERSION**

V1.01

## 1.43 The 'CD' DLC

**NAME**

CD - Create directory

**SYNTAX**

CD [directory-name]

**FUNCTION**

Will create a new directory without leaving ASM-One.

**INPUT**

A [directory-name]

If none given, ASM-One will ASK for one:

DIRECTORYNAME>

**RESULT**

If succesfull, the directory will be created. If not, it will not be created B-)

**REMARKS**

ASM-One will create the directory in the current directory (unless an assign or drive label is used, ofcourse). Check where you are with

The 'V' DLC

.

**FROM VERSION**

V1.06

## 1.44 The 'CS' DLC

### NAME

CS - Create Sinus

### SYNTAX

CS

### FUNCTION

Will create a sinus at the specified memory address.

### INPUT

DEST>           Memory destination adress  
BEG>            Begin angle  
END>            End angle  
AMOUNT>         Number of values generated  
AMPLITUDE>      Amplitude  
YOFFSET>        Yoffset  
SIZE>           (B)ytes / (W)ords / (L)ongwords  
MULTIPLIER>     Afterwards value multiplier (0 = 1)  
HCORRECTION>    Half step correction (Against irritating ticks)  
RCORRECTION>    ROUND instead of INT (0,7 = 1 instead of 0)

### RESULT

A Sinus.

### FROM VERSION

V1.05

## 1.45 The 'D' DLC

### NAME

D - Edit Disassembled Memory

### SYNTAX

D[address/label]

### FUNCTION

You will jump to the monitor starting at the specified address or label. If you didn't specify an address or label, the Monitor will disassemble memory from address \$0.

### INPUT

Address or label or nothing.

### RESULT

Disassembled memory in the Monitor.

### REMARKS

---



If you want to disassemble your source (by using a label) you first have to assemble your source, otherwise ASM-One will not recognize the label.

FROM VERSION  
V1.01

SEE ALSO

The '@D' DLC  
, The Monitor

## 1.46 The 'E' DLC

NAME  
E - Load External binaries

SYNTAX  
E[number]

FUNCTION  
Loads the External binaries which where defined in the source  
by >EXTERN .

Loads the External binary with the specified [number] if a number  
was specified by >EXTERN .

RESULT  
The External binaries will be loaded at the specified addresse(s).

REMARKS  
This command is here for backwards compatibility with SEKA. It's  
better (and saver) to use INCBIN for including binaries,  
becose EXTERN will load binaries at an ABSOLUTE memory address, while  
INCBIN will load binaries at a RELATIVE address.

FROM VERSION  
V1.01

## 1.47 The 'EL' DLC

NAME  
EL - Extend Labels

SYNTAX  
EL

FUNCTION  
Will put a prefix or postfix to ALL labels in the current source:

---

## INPUT

Extend Labels with> Characters to extend labels with  
Prefix (Y/N)> Y = Prefix, N = Postfix  
Pass 1..  
Pass 2..  
\*\* End of File

## RESULT

New labels

## REMARK

This is a very slow procedure, so be patient.

Y will generate a Prefix, any other answer a Postfix.

## FROM VERSION

V1.25

## BUGS

There is now way back after you entered the 'Extend labels with>'.

## 1.48 The 'F' DLC

## NAME

F - Fill Memory

## SYNTAX

F

## FUNCTION

Will FILL the specified memory with the specified value.

## INPUT

BEG> Start address or label  
END> End address or label  
DATA> Data to fill memory with

## RESULT

A filled part of memory.

## REMARKS

DATA can only by a BYTE.

Be EXTREMELY carefull with this DLC, becose it could trash your memory.

## FROM VERSION

V1.01

## BUGS

ASM-One will accept any type of DATA (ASCII, HEX, Octall, Decimal), but will ALWAYS take the last BYTE to fill the memory with.

ASM-One will not warn you if you use other sizes than BYTE.

---

## 1.49 The 'G' DLC

### NAME

G - Start executing at Address untill BREAKPOINT reached

### SYNTAX

G[address/label]

### FUNCTION

Will start program at address/label, untill the breakpoint is reached

### INPUT

An address or label. And if you wish, a breakpoint.

### RESULT

How the program effected the registers. And maybe an error.

### REMARKS

If NO address or label is given, ASM-One will start at address \$0, this will most certainly crash your Amiga !!!!

### FROM VERSION

V1.01

## 1.50 The 'H' DLC

### NAME

H - Edit Memory in HEX

### SYNTAX

H[address/label]

### FUNCTION

Will switch to the Monitor in HEX format.

See The Monitor for more information.

### FROM VERSION

V1.01

### SEE ALSO

The '@H' DLC  
, {" The Monitor " LINK "monitor" }

## 1.51 The 'I' DLC

### NAME

I - Insert Source

---

**SYNTAX**

I [filename]

**FUNCTION**

Will insert the selected source.

If no filename is given, and reqtools is enabled, a file requester will show up to allow you to select a file to insert.

The new file will be inserted at the location of the cursor in the current source.

**RESULT**

An inserted source at the location of your cursor.

**FROM VERSION**

V1.01

## 1.52 The 'IB' DLC

**NAME**

IB - Insert Binary

**SYNTAX**

IB

**FUNCTION**

Will allow you to insert binary values as DC.B's into the source, starting at the current location of the cursor.

**INPUT**

BEG> Starting Address or Label  
END> Ending Address or Label

**RESULT**

Inserted DC.B's at the location of your cursor.

**FROM VERSION**

V1.01

## 1.53 The 'ID' DLC

**NAME**

ID - Insert Disassembly

**SYNTAX**

ID

**FUNCTION**

Will allow you to insert disassembled memory as source into the source, starting at the current location of the cursor.

---

## INPUT

BEG> Starting Address or Label  
END> Ending Address or Label  
Remove unused labels (Y/N)? Y = Yes, No = No

## RESULT

Inserted disassembled memory at the location of your cursor.

If you don't remove the unused labels, every line will be preceded by a label.

Y will remove the unused labels, they are not removed otherwise.

## FROM VERSION

V1.01

## BUGS

At the 'Remove unused labels (Y/N)?' question, there is no way back !!

## 1.54 The 'IH' DLC

## NAME

IH - Insert HEX lines

## SYNTAX

IH

## FUNCTION

Will allow you to insert hexadecimal values as DC.B's into the source, starting at the current location of the cursor.

## INPUT

BEG> Starting Address or Label  
END> Ending Address or Label

## RESULT

Inserted DC.B's at the location of your cursor.

## FROM VERSION

V1.01

## 1.55 The 'IN' DLC

## NAME

IN - Insert ASCII lines

## SYNTAX

IN

## FUNCTION

Will allow you to insert ASCII values as DC.B's into

---

the source, starting at the current location of the cursor.

**INPUT**

BEG> Starting Address or Label  
END> Ending Address or Label

**RESULT**

Inserted DC.B's at the location of your cursor.

**FROM VERSION**

V1.01

## 1.56 The 'IS' DLC

**NAME**

IS - Insert Sinus

**SYNTAX**

IS

**FUNCTION**

Will create a sinus at the current cursor location.

**INPUT**

DEST> Memory destination address  
BEG> Begin angle  
END> End angle  
AMOUNT> Number of values generated  
AMPLITUDE> Amplitude  
YOFFSET> Yoffset  
SIZE> (B)ytes / (W)ords / (L)ongwords  
MULTIPLIER> Afterwards value multiplier (0 = 1)  
HCORRECTION> Half step correction (Against irritating ticks)  
RCORRECTION> ROUND instead of INT (0,7 = 1 instead of 0)

**RESULT**

A Sinus.

**FROM VERSION**

V1.05

## 1.57 The 'J' DLC

**NAME**

J - Jump to address/label

**SYNTAX**

J[address/label]

**FUNCTION**

Will start program at address/label, as a subroutine.

---

**INPUT**

An address or label, if non given and the source was assembled, ASM-One will jump to the beginning of the source. If no source is assembled, ASM-One will return immediately.

**RESULT**

How the program effected the registers. And maybe an error.

**FROM VERSION**

V1.01

## 1.58 The 'K' DLC

**NAME**

K - Single step n steps

**SYNTAX**

K[steps]

**FUNCTION**

Will start program at address/label, step by step

**INPUT**

Number of steps, if non given , ASM-One will return immediately. If no source is assembled, ASM-One will return immediately.

**RESULT**

How the program effected the registers. And maybe an error.

**FROM VERSION**

V1.01

## 1.59 The 'L' DLC

**NAME**

L - Locate Text

**SYNTAX**

L[text]

**FUNCTION**

Text search function from the command line.

**INPUT**

Some text. If no text given, L will locate the last text it got with this DLC.

**RESULT**

First line which contains the located text.

---

FROM VERSION  
V1.01

## 1.60 The 'M' DLC

NAME  
M - Edit Memory

SYNTAX  
M[address/label]

FUNCTION  
Gives you the possibility to edit memory in HEX, Decimal and ASCII at the given address or label. If no address or label given, ASM-One will start at address \$0.

RESULT  
Changed memory.

REMARKS  
Becarefull, this DLC could ruine your day. Save your source before using this DLC.

FROM VERSION  
V1.01

## 1.61 The 'N' DLC

NAME  
N - Edit Memory in ASCII

SYNTAX  
N[address/label]

FUNCTION  
Gives you the possibility to edit memory in ASCII at the given address or label. If no address or label given, ASM-One will start at address \$0.

RESULT  
Changed memory.

REMARKS  
Becarefull, this DLC could realy ruine your day. Save your source before using this DLC.

SEE ALSO  
The Monitor

FROM VERSION  
V1.01

---



## 1.62 The 'O' DLC

### NAME

O - Rescue Source

### SYNTAX

O

### FUNCTION

If you by accident did ZS, O will restore your source.

### RESULT

Your source back.

### REMARKS

This may also work if the Amiga crashed. Be sure to allocate the same address space (although this maybe difficult with the ever increasing size of Memory B-), for the best results.

You may also use the 'ASM-One Source Rescuer', to try to restore old sources from memory.

### FROM VERSION

V1.01

## 1.63 The 'P' DLC

### NAME

P - Print Lines

### SYNTAX

P[number-of-lines]

### FUNCTION

Gives you the possibility to print a given number of lines from the CURRENT cursor position.

### RESULT

Source lines on paper.

### REMARKS

Don't forget to mark 'Printer Dump' in the 'General Preferences'. Or press Ctrl+P.

### FROM VERSION

V1.01

## 1.64 The 'PS' DLC

### NAME

PS - Set CLI Parameters

---

## SYNTAX

PS

## FUNCTION

Some programs need CLI input (like DIR) to function properly.  
With PS you can set the CLI parameters.

## INPUT

Startup parameters&gt; The parameters

## RESULT

The parameters (text) will be stored in a memory location.  
Every time you start debugging, the address of this location  
is put into A0 (just like the real thing B-)

D0 will contain how many characters there where (including  
the trailing \$a).

## REMARKS

PS will save the parameters, so you don't have to enter them  
every time you debug some source. So A0 will ALWAYS point  
to the parameters, and D0 will ALWAYS contain there number  
(only AFTER you once specified the parameters).

The text will always be ended with an \$a (= RETURN).

This ONLY works while DEBUGGING !!!

## FROM VERSION

V1.09

## 1.65 The 'Q' DLC

## NAME

Q - Compare Memory

## SYNTAX

Q

## FUNCTION

Compare Memory for differences.

## INPUT

BEG> Starting address/label of the first block  
END> Ending address/label of the first block  
DEST> Starting address/label of the second block

## RESULT

Will show TWO addresses. The first is the address from  
the byte in the first block that is different from the  
byte at the address shown as the second address (which  
should be in the second block).

If no differences where found, you will get the following  
message:

---

\*\* Equal Areas

If any difference was found:

<first address> <second address>

\*\* NOT Equal Areas

#### REMARKS

You can't specify the end of the second block. That's not strange, since you will compare memory against the first block. Everything that's bigger than the first block, will not be interesting to compare.

So, the first block will specify the size of the second block (which is: (END-BEG)+DEST).

#### FROM VERSION

V1.01

## 1.66 The 'R' DLC

#### NAME

R - Read Source

#### SYNTAX

R [filename]

#### FUNCTION

Read a source in the currently selected source of the editor (with a maximum of ten).

#### INPUT

A [filename], if non given and reqtools is enabled, a requester will show up to allow you to select a source.

#### RESULT

A source to work with in the Editor.

#### REMARKS

ASM-One will read any file, it will NOT check if the file is a source file.

#### FROM VERSION

V1.01, without requester: V1.05

#### SEE ALSO

The 'W' DLC

---

## 1.67 The 'RB' DLC

NAME

RB - Read Binary

SYNTAX

RB [filename]

FUNCTION

Read a binary file at the specified address/label.

INPUT

BEG> Starting address or label

END> Ending address or label

A [filename], if non given and reqtools is enabled, a requester will show up to allow you to select a source.

RESULT

A part of your memory filled with a binary file.

REMARKS

ASM-One will read any file, it will NOT check if the file is a binary file.

If a file is longer than specified by END>, it will be truncated to fit in the specified block.

FROM VERSION

V1.01, without requester: V1.05

SEE ALSO

The 'WB' DLC

## 1.68 The 'RE' DLC

NAME

RE - Read Environment (Project)

SYNTAX

RE [filename]

FUNCTION

Read a complete Environment (Project).

INPUT

A [filename], if non given and reqtools is enabled, a requester will show up to allow you to select a source.

RESULT

---

Every one of the 10 source you where working on, is loaded at the same place they where before..

#### REMARKS

This options doesn't work right yet. Many people (including me) have no idea how it really works. Another problem is that the same thing has different names, which is very confusing (like Read Environment (RE), Project Info (=P) and Update All (UA)).

#### SEE ALSO

The 'WE' DLC  
,  
The '=P' DLC  
,  
The 'UA' DLC  
FROM VERSION

V1.29

#### BUGS

Many... Use at your own risk..

## 1.69 The 'RO' DLC

NAME

RO - Read Object

#### SYNTAX

RO [filename]

#### FUNCTION

Read an object (executable) file at the specified address/label.

#### INPUT

BEG> Starting address or label  
END> Ending address or label

A [filename], if non given and reqtools is enabled, a requester will show up to allow you to select a source.

#### RESULT

A part of your memory filled with an object file.

The address where the file was loaded.

#### REMARKS

Memory is allocated for the object file, it's read into memory link if it was a program. The address that's returned is the first line of coding of the loaded binary (and can be executed by jumping to it).

#### SEE ALSO

---

The 'WO' DLC  
FROM VERSION  
V1.01, without requester: V1.05

## 1.70 The 'RS' DLC

### NAME

RS - Read Sector(s) from floppy disk

### SYNTAX

RS[drive-number]

### FUNCTION

Read some sectors from the floppy disk which is in the specified drive.

### INPUT

RAM PTR> Address/label where the sectors should be placed  
DISK PTR> Sector on disk to start with  
LENGHT> Number of sectors to read

You may specify a Drive number. Legal numbers are:

0: DF0  
1: DF1  
2: DF2  
3: DF3

DF0 is default, so if you don't specify a Drive number, the sectors will be read from the disk in DF0.

### RESULT

The sectors read from the disk will be located at the specified address or label.

### REMARKS

The Amiga floppy disk (880 Kb formatted) has 1760 sectors.  
Each sectors is 512 bytes in size.

### SEE ALSO

The 'WS' DLC  
FROM VERSION  
V1.01

## 1.71 The 'RT' DLC

### NAME

RT - Read Track(s) from floppy disk

---

## SYNTAX

RT[drive-number]

## FUNCTION

Read some tracks from the floppy disk which is in the specified drive.

## INPUT

RAM PTR> Address/label where the tracks should be placed  
DISK PTR> Track on disk to start with  
LENGHT> Number of tracks to read

You may specify a Drive number. Legal numbers are:

0: DF0  
1: DF1  
2: DF2  
3: DF3

DF0 is default, so if you don't specify a Drive number, the tracks will be read from the disk in DF0.

## RESULT

The tracks read from the disk will be located at the specified address or label.

## REMARKS

The Amiga floppy disk (880 Kb formatted) has 80 tracks.  
Each divided into 11 sectors.

## SEE ALSO

The 'WT' DLC  
FROM VERSION

V1.01

## 1.72 The 'S' DLC

## NAME

S - Search Memory

## SYNTAX

S

## FUNCTION

Search the specified block of memory for the specified data.

## INPUT

BEG> Starting address/label of the memory block  
END> Ending address/label of the memory block  
DATA> Data to be searched for

---

To search more specific, DATA can be specified with a size:

.B = Byte  
.W = Word  
.L = Longword

#### RESULT

If the data was found, every address at which it was found will be displayed.

#### REMARKS

Any DATA is allowed: ASCII, Hex or decimal.

For anyone interested, this DLC is perfect for cracking save-games B-)

#### FROM VERSION

V1.01

## 1.73 The 'T' DLC

#### NAME

T - Top of Source

#### SYNTAX

T[line-number]

#### FUNCTION

Jumps to the top of your source, or the specified line-number.

#### INPUT

Nothing or a line number.

#### RESULT

See 'FUNCTION'.

#### REMARKS

ASM-One will not jump to the top of the source if the source has more than 65535 lines. This is because ASM-One stores the number of lines in WORDS..

#### SEE ALSO:

The 'B' DLC  
FROM VERSION

V1.01

## 1.74 The 'U' DLC

---



## NAME

U - Update current source in the Editor

## SYNTAX

U

## FUNCTION

If the source has changed, ASM-One will update your source.

## RESULT

An updated source.

## REMARKS

ASM-One will check if the source is still there. If not, ASM-One will update the source anyway.

## FROM VERSION

V1.09, Deleted-source check from: V1.3x and up

## 1.75 The 'UA' DLC

## NAME

UA - Update ALL source currently open

## SYNTAX

UA

## FUNCTION

Will update all changed sources open at the moment.

## RESULT

Updated sources.

## REMARKS

This function is not completely perfect yet. And it may have something to do with the Project and Environment DLCs

## SEE ALSO

The 'RE' DLC

'  
The 'WE' DLC

'  
The '=P' DLC

FROM VERSION

V1.29

## 1.76 The 'V' DLC

---

**NAME**

V - Change current directory or view a directory

**SYNTAX**

V [directory-name]

**FUNCTION**

You can change your current working directory, or see the contents of a specified directory.

**INPUT**

A [directory-name], if none given, ASM-One will show you the current directory.

**RESULT**

New working directory, or contents of the specified directory.

**REMARKS**

If you do 'V' more than once in a directory, ASM-One will show the directory from memory the second and following times. The directory is only really read the first time.

**FROM VERSION**

V1.08

## 1.77 The 'W' DLC

**NAME**

W - Write source currently in the Editor

**SYNTAX**

W [file-name]

**FUNCTION**

Will write the current source, including the 10 Marks.

**INPUT**

A [file-name], if non given and reqtools enabled, a file requester will open to allow you to specify the file name.

**RESULT**

If good, the file will be written.

**SEE ALSO**

The 'R' DLC

FROM VERSION

V1.01, without requester: V1.05

---

## 1.78 The 'WB' DLC

NAME

WB - Write binary

SYNTAX

WB [file-name]

FUNCTION

Will write binary data to the specified file.

INPUT

BEG> Starting address/label of the binary

END> Ending address/label of the binary

A [file-name], if non given and reqtools enabled, a file requester will open to allow you to specify the file name.

RESULT

If good, the file will be written.

REMARKS

You can write an assembled source as binary, but you can also save blocks of memory as binary.

SEE ALSO

The 'RB' DLC

FROM VERSION

V1.01, without requester: V1.05

## 1.79 The 'WE' DLC

NAME

WE - Write Environment (Project)

SYNTAX

WE [file-name]

FUNCTION

Will save all sources currently loaded in ASM-One.

INPUT

A [file-name], if non given and reqtools enabled, a file requester will open to allow you to specify the file name.

RESULT

If good, the file (with extension .Aprj) will be written.

REMARKS

Doesn't work correct yet.. Use at own risk.

---

SEE ALSO

The 'RE' DLC  
'  
The '=P' DLC  
'  
The 'UA' DLC  
FROM VERSION

V1.29

## 1.80 The 'WL' DLC

NAME

WL - Write Link file

SYNTAX

WL [file-name]

FUNCTION

Will save an assembled source as an Link file, to be linked with an (for example: C) compiler.

INPUT

A [file-name], if non given and reqtools enabled, a file requester will open to allow you to specify the file name.

RESULT

If good, the file will be written.

REMARKS

The Link file is ALink and BLink compatible.

FROM VERSION

V1.29

BUGS

Seems to be giving problems with it's SECTIONS !!

## 1.81 The 'WN' DLC

NAME

WN - Write Source WITHOUT Marks

SYNTAX

WN [file-name]

FUNCTION

Will save the source currently in the editor as a plane ASCII file.

---

**INPUT**

A [file-name], if non given and reqtools enabled, a file requester will open to allow you to specify the file name.

**RESULT**

If good, the file will be written.

**FROM VERSION**

V1.3x

## 1.82 The 'WO' DLC

**NAME**

WO - Write Object

**SYNTAX**

WO [file-name]

**FUNCTION**

Will save an assembled source as an object (executable).

**INPUT**

A [file-name], if non given and reqtools enabled, a file requester will open to allow you to specify the file name.

**RESULT**

If good, the file will be written.

**SEE ALSO**

The 'RO' DLC

FROM VERSION

V1.01, without requester: V1.05

## 1.83 The 'WP' DLC

**NAME**

WP - Write Preferences

**SYNTAX**

WP

**FUNCTION**

Will save the current setting of the Preferences to the ASM-One.pref file.

**RESULT**

New ASM-One.pref file.

---

## REMARKS

This action CAN NOT be reversed !!! So take care..

## FROM VERSION

V1.01

## 1.84 The 'WS' DLC

## NAME

WS - Write Sector(s) to floppy disk

## SYNTAX

RS[drive-number]

## FUNCTION

Write specified sectors to the floppy disk which is in the specified drive.

## INPUT

RAM PTR> Starting address/label of data to be written  
DISK PTR> Sector on disk to start with  
LENGHT> Number of sectors to read

You may specify a Drive number. Legal numbers are:

0: DF0  
1: DF1  
2: DF2  
3: DF3

DF0 is default, so if you don't specify a Drive number, the sectors will be written to the disk in DF0.

## REMARKS

The Amiga floppy disk (880 Kb formatted) has 1760 sectors.  
Each sectors is 512 bytes in size.

Don't forget to use

The 'CC' DLC

if you have written

something over the bootblock (sector 0 and 1), if you want your disk to be bootabel...

## FROM VERSION

V1.01

## 1.85 The 'WT' DLC

## NAME

WT - Write Track(s) to floppy disk

---

## SYNTAX

WT[drive-number]

## FUNCTION

Write specified tracks to the floppy disk which is in the specified drive.

## INPUT

RAM PTR> Starting address/label of data to be written  
DISK PTR> Track on disk to start with  
LENGHT> Number of tracks to read

You may specify a Drive number. Legal numbers are:

0: DF0  
1: DF1  
2: DF2  
3: DF3

DF0 is default, so if you don't specify a Drive number, the tracks will be written to the disk in DF0.

## REMARKS

The Amiga floppy disk (880 Kb formatted) has 80 tracks. Each divided into 11 sectors.

Don't forget to use  
    The 'CC' DLC  
    if you have written  
something over the bootblock (sector 0 and 1), if you want your disk to be bootabel...

## FROM VERSION

V1.01

## 1.86 The 'X' DLC

## NAME

X - Look at/Change registers

## SYNTAX

X[register]

## FUNCTION

Allows you to change the contents of a register.

## INPUT

If no [register] given, ASM-One will show the values of the registers. Changes from the last time you saw the registers are printed inverted.

## RESULT

Changed registers (they now look like you intended them

---

to look B-)

**REMARKS**

Be very carefull, save your source before using this DLC.

**FROM VERSION**

V1.01

## 1.87 The 'Y' DLC

**NAME**

Y - Execute DOS program

**SYNTAX**

Y [file-name]

**FUNCTION**

Allows you to start DOS programs from within ASM-One.

**INPUT**

A [file-name], if non given and reqtools enabled, a file requester will open to allow you to specify the file name.

**RESULT**

New DOS program started.

**REMARKS**

Very simple version of the DOS 'RUN' command.

**FROM VERSION**

V1.07

## 1.88 The 'ZA' DLC

**NAME**

ZA - Zap Auto-Alloc memory

**SYNTAX**

ZA

**FUNCTION**

Will remove any memory that was Automatically allocated by ASM-One during the assembly of a source.

**RESULT**

More free memory.

**REMARKS**

Free's memory allocated during assembly. Memory is automatically allocated if you have the 'Auto-Alloc' option in the Preferences ON.

---



FROM VERSION  
V1.29

## 1.89 The 'ZB' DLC

NAME  
ZB - Zap Breakpoints

SYNTAX  
ZB

FUNCTION  
Zap Breakpoint set with  
The 'G' DLC  
RESULT  
No more breakpoints

FROM VERSION  
V1.01

## 1.90 The 'ZF' DLC

NAME  
ZF - Zap File

SYNTAX  
ZF [file-name]

FUNCTION  
Delete specified file.

INPUT  
A [file-name], if non given and reqtools enabled, a  
file requester will open to allow you to specify the file  
name.

RESULT  
Your file is GONE !!

REMARK  
NO questions ar asked..

FROM VERSION  
V1.01

## 1.91 The 'ZI' DLC

---

## NAME

ZI - Zap Includes

## SYNTAX

ZI

## FUNCTION

Remove all loaded includes.

## RESULT

No more resident includes.

## REMARK

Files which are loaded with INCBIN will stay resident. If you have changed some of your include files, you will have to use this DLC to remove the resident includes, so ASM-One will read your changed includes.

## FROM VERSION

V1.01

## 1.92 The 'ZS' DLC

## NAME

ZS - Zap Source

## SYNTAX

ZS

## FUNCTION

Remove source currently in the editor.

## RESULT

An empty editor (only current source !!)

## REMARK

You will get a requester, besides  
The 'O' DLC  
it's your last  
change to SAVE your source...

## FROM VERSION

V1.01

## 1.93 Warnings & Errors

Warnings & Errors  
-----

ASM-One knows more than 100 errors. Currently most errors are very cryptic, and

---

in some cases you can get different errors for what is actually the same problem.

This problem will be addressed as soon as possible.

Most of the errors are generated when assembling. The rest will show up when using

DLCs

.

Warnings are ONLY generated when assembling. They warn you about the CPU/MMU/FPU you where programming.

Errors

Warnings

## 1.94 Errors

Workspace Memory Full

Address Register Byte/Logic

Address Register Expected

Comma Expected

Data Register Expected

Double Symbol

Unexpected End of File

End of File

User made FAIL

Illegal Command

Illegal Address Size

Illegal Operand

Illegal Operator

Illegal Operator in BSS Area

Illegal Order

Illegal Register Size

Illegal Section Type

---

Illegal Size  
Illegal MACRO Definition  
Immediate Operand Expected  
Include Jam  
MACRO Overflow  
Invalid Address Mode  
LOAD without ORG

## 1.95 Errors

Missing Quote  
Conditional Overflow  
No Operand Space Allowed  
No Constant/Label  
Not in MACRO  
Out of Range 0/3/4/5/6/7/8/16/32 bit  
Relative Mode Error  
Reserved Word  
Closing Parenthesis Expected  
SECTION Overflow  
String Expected  
Undefined Symbol  
Register Expected  
Word at ODD Address  
Not a Local Area  
Code Moved During Pass 2..  
Bcc.B Out of Rang in MACRO  
Out of Range (20 to 100)

---

Out of Range (60 to 132)

Include Overflow

Linker Limitation

Repeat Overflow

Not in Repeat Area

Double Definition

## 1.96 Errors

Relocation made to EMPTY Section

File Error

No Files

No Object

No File Space

Printer Device Missing

Not Done

Illegal Path

Illegal Device

Write Protected

No Disk in Drive

Illegal Option !!

REM without EREM

TEXT without ETEXT

Illegal Xn Scale Size

Field Offset Expected: {OFFSET:width}

Missing Brace

Colon Expected

Missing Bracket

FPU needed for Operation !!

To Many Watchpoints (maximal 8)  
Illegal Source, not Activated !!  
No Valid Memory Directory Present  
AUTO Command Overflow (more than 256 characters)

## 1.97 Errors

End should be Higher than Start !  
Illegal Source Number  
Including Empty Source ?  
Include Source Jam  
Unknown Conversion Mode: should be RB or RN  
Unknown CMAP Place: should be (B)EFORE, (A)FTER or (N)ONE  
Unknown CMAP Mode: should be AGA or ECS  
Trying to Include NON IFF File  
IFF File is Not a ILBM File  
Can't handle a BODY before BMHD Chunk  
Illegal Floating Point Size: should be B, W, L, S, D, X or P  
Illegal Size for Data Register: should be B, W, L or S  
Bcc.W Out of Range in MACRO  
Floating Point Register Expected  
This is not an ASM-One Project File  
Bitfield can't be Out of 32 Bit Range  
Illegal CMAP Mode Error: expecting 12, 24, CE or CA  
Illegal Color Offset (Bank) Specified  
IFF Color(s) Out of Range  
To Many Errors  
Invalid Function Code: only SFC, DFC, Dn or #<data> Allowed

---

Address Register Indirect Mode Expected, for Return Address Register ←

Address Register Direct Mode Expected, for Return Address Register ←

Level Out of 3 Bit Range

Mask Out of 3 Bit Range

Mask Out of 4 Bit Range

## 1.98 Warnings & Errors

Function Code Out of 3 Bit Range

Function Code Out of 4 Bit Range

Insuficient Memory to change Source

MMU Register Expected as 1st Operand

MMU Register Expected as 2nd Operand

Invalid MMU Register as 1st Operand

Invalid MMU Register as 2nd Operand

Missing START address or Label

Incorrect Index Register Specified

PC/ZPC can't be used as Index Register

We haven't come up with a nice text for this error Error

ZPC Not allowed in this addressing mode

Field Width Expected {offset:WIDTH}

Field Width can't be ZERO

## 1.99 Workspace Memory Full

MESSAGE

Workspace Memory Full

CAUSE

When starting ASM-One, you are allocating an certain amount of memory

for the source.

If you assemble the source, some of the allocated memory is used to store pointers, reloc labels, the assembled source (depending on the type of sections) and other stuff. This is done to reduce memory fragmentation.

The allocated memory maybe enough to load the source, but not to assemble it.

SALVE

Use the

The '=M' DLC  
to allocate more memory.

## 1.100 Address Register Byte/Logic

MESSAGE

Address Regsiter Byte/Logic

CAUSE

- 1) You can't access an Address Register Direct with a Byte size
- 2) You can't use an Address Register in an logical opcode (like OR.x)

EXAMPLE

- 1) Address Register Direct with Byte size:

```
MOVE.B A0,D0
```

- 2) Address Register Direct in logical opcode:

```
OR.W A0,D0
```

SALVE

Move the Address Register to a Data Register.

## 1.101 Address Register Expected

MESSAGE

Address Register Expected

CAUSE

Some opcodes need an Address Register as source and/or destination operand.

EXAMPLE

```
CMPA.L (A0),D0
```

SALVE

Use another opcode with the same functionality but which accepts

---



the operand you want (not possible in all cases).

## 1.102 Comma Expected

### MESSAGE

Comma Expected

### CAUSE

Most opcodes need two operands. These must be separated by a comma.

### EXAMPLE

```
MOVE.L (A0)
```

### SALVE

Supply the missing operand.

## 1.103 Data Register Expected

### MESSAGE

Data Register Expected

### CAUSE

Some opcodes need a Data Register as source and/or destination operand.

### EXAMPLE

```
MOVEQ #4,A0
```

### SALVE

Use another opcode with the same functionality but which accepts the operand you want (not possible in all cases).

## 1.104 Double Symbol

### MESSAGE

Double Symbol

### CAUSE

It's not possible to have two labels with the same name.

### SALVE

Come up with another name for the label.  
Set 'UCase=LCase' off.  
Use local labels.

---

## 1.105 Unexpected End of File

### MESSAGE

Unexpected End of File

### CAUSE

Generated because a MACRO definition, REPT block, or IF statement was not properly closed.

### SALVE

Check your MACRO , REPT and IF definitions.

## 1.106 End of File

### MESSAGE

End of File

### CAUSE

The End of File mark was reached.

Mostly generated after a search/replace operation.

### SALVE

It's not really an error, more some kind of notification to the user.

## 1.107 User made FAIL

### MESSAGE

User made FAIL

### CAUSE

Generated when during assembling the FAIL directive is encountered.

### SALVE

Remove this directive.  
See FAIL

## 1.108 Illegal Command

### MESSAGE

Illegal Command

### CAUSE

While assembly, ASM-One found a word in the command column, that's unknown to ASM-One.

### SALVE

---

Remove that word.  
Check your spelling.

## 1.109 Illegal Address Size

MESSAGE

Illegal Address Size

CAUSE

- You used the Absolute Addressing mode with a Byte size
- You tried to clear a byte of an Address Register

EXAMPLE

```
MOVE.L ($FE).B,D0
```

```
CLR.B A0
```

SALVE

Use another size.

## 1.110 Illegal Operand

MESSAGE

Illegal Operand

CAUSE

The operand(s) used are illegal for this opcode  
There's an illegal operand somewhere on the source line

EXAMPLE

SALVE

Remove the offending operand.  
Check for spelling.

## 1.111 Illegal Operator

MESSAGE

Illegal Operator

CAUSE

A misspelled data directive.

EXAMPLE

```
test: dcg.w $0
```

SALVE

---

Check your spelling.

## 1.112 Illegal Operator in BSS Area

MESSAGE

Illegal Operator in BSS Area

CAUSE

Only the DS may be used in a BSS area.  
A misspelled data directive.

EXAMPLE

```
SECTION TEST,BSS_P
test: DC.W $0
```

SALVE

Check for spelling.  
See DS

## 1.113 Illegal Order

MESSAGE

Illegal Order

CAUSE

MOVEM expects Address and Dataregister as operands. The list should first contain the Data Registers and then the Address Registers, in an increasing order.

EXAMPLE

```
MOVEM.L A0,D4-D0.-(A7)
```

SALVE

Swap the Registers.

## 1.114 Illegal Register Size

MESSAGE

Illegal Register Size

CAUSE

Index registers can only have a WORD or LONGWORD size.

EXAMPLE

```
MOVE.L (A0,D0.B),D1
```

SALVE

---

Remove the size, default is WORD.  
Change the size to WORD or LONGWORD.

## 1.115 Illegal Section Type

MESSAGE

Illegal Section Type

CAUSE

The SECTION directive allows the following section types:

CODE = A code section

DATA = A data section

BSS = A BSS section

EXAMPLE

```
SECTION TEST,CHIP_P
```

SALVE

Use one of the legal section types

Remove the section.

See SECTION

## 1.116 Illegal Size

MESSAGE

Illegal Size

CAUSE

You specified an illegal size for an operand, or for the index register in the used addressing mode.

EXAMPLE

```
ADDA.B D0,A0
```

```
MOVE.L ([A0],D1.B*8),D3
```

SALVE

Change the size.

## 1.117 Illegal MACRO Definition

MESSAGE

Illegal MACRO Definition

CAUSE

You specified a MACRO within a MACRO.

---

SALVE

Recode your MACRO definition.  
See MACRO .

## 1.118 Immediate Operand Expected

MESSAGE

Immediate Operand Expected

CAUSE

Some opcodes allow ONLY Immediate Operands as first operand.

EXAMPLE

```
ADDI.B D0, (A0)
```

SALVE

Use another opcode (possible in most cases).

## 1.119 Include Jam

MESSAGE

Include Jam

CAUSE

A very unique error, only happens when you include includes during pass two.

SALVE

Check your include(s).  
Check conditional directives.

## 1.120 MACRO Overflow

MESSAGE

MACRO Overflow

CAUSE

MARCO's can only be nested 25 levels deep.

SALVE

See MACRO .

## 1.121 Invalid Addressing Mode

---

MESSAGE

Invalid Addressing Mode

CAUSE

Not all operands allow all addressing modes.

SALVE

Check your Programmers Manual.

## 1.122 LOAD without ORG

MESSAGE

LOAD without ORG

CAUSE

LOAD can't be used without ORG.

SALVE

Add an ORG directive.

See LOAD and ORG .

## 1.123 Missing Quote

MESSAGE

Missing Quote

CAUSE

Text blocks should have an ending Quote.

EXAMPLE

```
test: DC.B 'hallo
```

SALVE

Check your Quotes.

## 1.124 Conditional Overflow

MESSAGE

Conditional Overflow

CAUSE

Conditional directives can only be nested 25 deep

SALVE

Check your conditional directives.

---

## 1.125 No Operand Space Allowed

MESSAGE

No Operand Space Allowed

CAUSE

Operands can't be separated with spaces or tabs

EXAMPLE

```
MOVE.L D0, (A0)
```

SALVE

Remove offending space(s) and/or tab(s).

## 1.126 No Constant/Label

MESSAGE

No Constant/Label

CAUSE

Occurs when you use MACRO names in calculations.

SALVE

- Remove the MACRO name.
- Check your spelling.

## 1.127 Not in MACRO

MESSAGE

Not in MACRO

CAUSE

Some directives (like ENDM, MEXIT or CMEXIT) are not allowed outside a MACRO.

SALVE

Check you MACRO's.

## 1.128 Out of Range 0/3/4/5/6/7/8/16/32 bit

MESSAGE

Out of Range 0/3/4/5/6/7/8/16/32 bit

CAUSE

The used value is bigger than the maximum size allowed.

SALVE

Check you spelling.

---



## 1.129 Relative Mode Error

### MESSAGE

Relative Mode Error

### CAUSE

PC related addressing modes are only allowed in the SAME section.

### SALVE

Use a non PC related addressing mode.

## 1.130 Reserved Word

### MESSAGE

Reserved Word

### CAUSE

You used a Register name as label or constant.

### EXAMPLE

```
srp: JSR do_somthing_good
```

### SALVE

Check the  
Register Names  
list.

## 1.131 Closing Parenthesis Expected

### MESSAGE

Closing Parenthesis Expected

### CAUSE

ASM-One found the opening parenthesis, but missed the closing parenthesis.

### EXAMPLE

```
test: MOVE.L (A0,D0.W,D0
```

### SALVE

Check you spelling.

## 1.132 SECTION Overflow

---

MESSAGE

SECTION Overflow

CAUSE

The maximum number of SECTION's is 255.

SALVE

Delete some SECTION's .

### 1.133 String Expected

MESSAGE

String Expected

CAUSE

Some directive (like PRINTT ) expect a string as operand.

SALVE

Give them a string.

### 1.134 Undefined Symbol

MESSAGE

Undefined Symbol

CAUSE

The label/constant was not defined, and ASM-One doesn't recognize it either.

SALVE

Check your spelling.

### 1.135 Register Expected

MESSAGE

Register Expected

CAUSE

The index register in addressing modes (Xn) must be an Address or Data Register.

EXAMPLE

```
MOVE.L ([ $45, A0], # $43, # $23), D1
```

SALVE

Check your spelling.

---

### 1.136 Word at Odd Address

MESSAGE

Word at Odd Address

CAUSE

The 68020++ CPU allows data to be at an ODD address.  
Code, however, may not be on an ODD address.

SALVE

Set the data even with ALIGN , EVEN , ODD or CNOP

### 1.137 Not a Local Area

MESSAGE

Not a Local Area

CAUSE

Local labels are sperated by global labels.

SALVE

Read more about Labels .

### 1.138 Code Moved During Pass 2..

MESSAGE

Code Moved During Pass 2..

CAUSE

In Pass 1 all labels get an address, they may NOT  
move during Pass 2 of the assembling process.

SALVE

Check your conditional directives.  
Check your spelling.

### 1.139 Bcc.B Out of Range in MACRO

MESSAGE

Bcc.B Out of Range in MACRO

CAUSE

Bcc.B's within MACRO's ar not optimized to Bcc.W's if needed.

SALVE

Check your MACRO's .

---

### 1.140 Out of Range (20 to 100)

MESSAGE

Out of Range (20 to 100)

CAUSE

PLEN accepts ONLY values from 20 to 100.

SALVE

Check PLEN .

### 1.141 Out of Range (60 to 132)

MESSAGE

Out of Range (60 to 132)

CAUSE

LLEN accepts ONLY values from 60 to 132.

SALVE

Check LLEN .

### 1.142 Include Overflow

MESSAGE

Include Overflow

CAUSE

Includes can be nested 5 levels deep. Normal includes are 3 levels deep nested.

SALVE

Check your includes.  
Check INCLUDE

### 1.143 Linker Limitation

MESSAGE

Linker Limitation

CAUSE

You can't make calculations with two XREF's

SALVE

Check your calculations.  
Check XREF , XDEF , GLOBAL , ENTRY and EXTRN

---

### 1.144 Repeat Overflow

MESSAGE

Repeat Overflow

CAUSE

REPT can be nested 4 levels deep.

SALVE

Check REPT .

### 1.145 Not in Repeat Area

MESSAGE

Not in Repeat Area

CAUSE

ENDR can only be used in conjunction with REPT.

SALVE

Check REPT and ENDR .

### 1.146 Double Definition

MESSAGE

Double Definition

CAUSE

A register can only be defined once with BASEREG. Or ENDB should be between two BASEREG's for the same register.

SALVE

Check BASEREG and ENDB .

### 1.147 Relocation made to EMPTY Section

MESSAGE

Relocation made to EMPTY Section

CAUSE

You made an reference to an empty SECTION, so you can't write an object (executable) file.

SALVE

Check your SECTION's .

---

## 1.148 File Error

### MESSAGE

File Error

### CAUSE

Occurs when your file couldn't be loaded or written.

### SALVE

Check your (hard)drive, or the device you are using.

## 1.149 No Files

### MESSAGE

No Files

### CAUSE

When you changed to an empty directory with V, and do V again. ASM-One will have no files to show.

### EXAMPLE

```
> V empty:
> V
```

```
-----
Name : empty                               0 Bytes Free    346624 Bytes Used
-----
```

```
** No Files
>
```

### SALVE

See

The 'V' DLC

.

## 1.150 No Object

### MESSAGE

No Object

### CAUSE

Occurs when you didn't assemble something, and tried to write an object (executable) anyway.

### SALVE

See

The 'A' DLC

.

## 1.151 No File Space

MESSAGE

No File Space

CAUSE

Occurs when there is no space to write a file to the selected device.

SALVE

See

The 'ZF' DLC  
and  
The 'Y' DLC  
.

## 1.152 Printer Device Missing

MESSAGE

Printer Device Missing

CAUSE

Occurs when (one of) the following files are missing, while you try to print:

- printer.device
- port.handler
- parallel.device
- serial.device (if printer connected to the serial port)

SALVE

Check your Amiga configuration, and the Manual of your Printer or your Amiga about Printers.

## 1.153 Not Done

MESSAGE

Not Done

CAUSE

Occurs whenever an action was canceled.

## 1.154 Illegal Path

MESSAGE

Illegal Path

CAUSE

---

Occurs when do V to an illegal/non-existing path.

SALVE

See

The 'V' DLC  
and The 'CD' DLC LINK .

## 1.155 Illegal Device

MESSAGE

Illegal Device

CAUSE

Occurs when [drive-number] is higher than 3 or lower than 0.

SALVE

See

The 'WT' DLC  
,  
The 'WS' DLC  
,  
The 'RT' DLC  
and  
The 'RS' DLC  
.

## 1.156 Write Protected

MESSAGE

Write Protected

CAUSE

Occurs when try to write to a write protected device.

SALVE

Remove the protection from the device, or those another device.

## 1.157 No Disk in Drive

MESSAGE

No Disk in Drive

CAUSE

Occurs when try to write to a device which does NOT contain a disk.

SALVE

Insert a disk in the selected device.

---



### 1.158 Illegal Option !!

MESSAGE

Illegal Option !!

CAUSE

Implemented but not used at the moment.

### 1.159 REM without EREM

MESSAGE

REM without EREM

CAUSE

A REM block MUST be closed with the EREM directive.

SALVE

See REM and EREM .

### 1.160 TEXT without ETEXT

MESSAGE

TEXT without ETEXT

CAUSE

A TEXT block MUST be closed with the ETEXT directive.

SALVE

See TEXT and ETEXT .

### 1.161 Illegal Xn Scale Size

MESSAGE

Illegal Xn Scale Size, only nothing, 1, 2, 4 or 8 allowed

CAUSE

If an addressing mode allows scaling of the index register, the scaling MUST be 0 (or nothing), 2, 4 or 8.

SALVE

Use a valid scaling size.

### 1.162 Field OFFSET Expected: {OFFSET:width}

**MESSAGE**

Field OFFSET Expected: {OFFSET:width}

**CAUSE**

Some 68020++ bit functions expect an offset and a width as operand.

**EXAMPLE**

```
test: BFEXTU D0{1:5},D1
```

**SALVE**

Provide a correct offset and width.

### 1.163 Missing Brace

**MESSAGE**

Missing Brace

**CAUSE**

Occurs when a brace is missing in the operand

**EXAMPLE**

```
test: BFEXTU D0{1:5,D1
```

**SALVE**

Check your operands.

### 1.164 Colon Expected

**MESSAGE**

Colon Expected

**CAUSE**

Occurs when the offset and width are not separated by a colon.

**EXAMPLE**

```
test: BFEXTU D0{1515},D1
```

**SALVE**

Check your operands.

### 1.165 Missing Bracket

**MESSAGE**

Missing Bracket

---

**CAUSE**

Some 68020++ addressing modes use brackets. If one of them is missing, or not at the expected place, you will get this error.

**EXAMPLE**

```
test: MOVE.L  ([$45,A0,D4.W*8,$2439),D2
```

**SALVE**

Check your operands.

## 1.166 FPU needed for Operation !!

**MESSAGE**

FPU needed for Operation !!

**CAUSE**

To calculate floats (both with the '[' DLC or in your source, you will need a FPU.

**SALVE**

Buy a FPU.  
See

The '[' DLC

## 1.167 To Many Watchpoints (maximal 8)

**MESSAGE**

To Many Watchpoints (maximum is 8)

**CAUSE**

The number off Watch Points (in the debugger) is currently limited op to 8.

## 1.168 Illegal Source, not Activated !!

**MESSAGE**

Illegal Source, not Activated !!

**CAUSE**

With the 'AS' DLC you can switch between the 10 sources of ASM-One. The number must be, preferable, between 0 and 9.

Other characters will generate this error

**SALVE**

See

---

The 'AS' DLC  
.

## 1.169 No Valid Memory Directory Present

MESSAGE

No Valid Memory Directory Present

CAUSE

If you use the 'V' DLC to examine an illegal directory, and do 'V' again, this error will be shown.

SALVE

See

The 'V' DLC  
.

## 1.170 Auto Command Overflow (more than 256 characters)

MESSAGE

AUTO Command Overflow (more than 256 characters)

CAUSE

If you specify more than 256 characters long commands for the AUTO directive, than this error message is generated.

SALVE

See AUTO .

## 1.171 End should be Higher than Start !

MESSAGE

End should be Higher than Start !

CAUSE

When using DLCs which need starting and ending addresses/labels, this error may occur when the start is higher than the end.

There's no negative block possible.

SALVE

See

The 'C' DLC  
(for example).

---

## 1.172 Illegal Source Number

### MESSAGE

Illegal Source Number

### CAUSE

With the INCSRC directive you can switch between the 10 sources of ASM-One. The number must be, preferable, between 0 and 9.

Other characters will generate this error

### SALVE

See INCSRC .

## 1.173 Including Empty Source ?

### MESSAGE

Including Empty Source ?

### CAUSE

When the source you wanted to include with the INCSRC directive is empty, you will get this error.

### SALVE

Include a source with something in it.

See INCSRC .

## 1.174 Include Source Jam

### MESSAGE

Include Source Jam

### CAUSE

This error will occur when you try to include the source which is currently active into the source that's currently active with the INCSRC directive.

### SALVE

Include another source.

See INCSRC .

## 1.175 Unknown Conversion Mode: should be RB or RN

### MESSAGE

Unknown Conversion Mode: should be RB or RN

### CAUSE

This error will occur when you use anything else than RB (RawBlit) or RN (RawNorm) as one of operand of the

INCIFFF directive.

SALVE

Use RB or RN.

See INCIFFF .

### **1.176 Unknown CMAP Place: should be (B)EFORE,(A)FTER or (N)ONE**

MESSAGE

Unknown CMAP Place: should be (B)EFORE, (A)FTER or (N)ONE

CAUSE

This error will occur when you use anything else than B (Before), A (After) or N (None) as one of operand of the INCIFFF directive.

SALVE

Use A, B or N.

See INCIFFF .

### **1.177 Unknown CMAP Mode: should be AGA or ECS**

MESSAGE

Unknown CMAP Mode: should be AGA or ECS

CAUSE

This error will occur when you use anything else than AGA or ECS as one of operand of the INCIFFF directive.

SALVE

Use AGA or ECS.

See INCIFFF .

### **1.178 Trying to Include NON IFF File**

MESSAGE

Trying to Include NON IFF File

CAUSE

This error will occur when you try to include a non IFF file with the INCIFFF directive.

SALVE

Select a valid IFF file.

See INCIFFF .

---

### 1.179 IFF File is Not a ILBM File

MESSAGE

IFF File is Not a ILBM File

CAUSE

This error will occur when you try to include an IFF which is not of the ILBM (picture) type with the INCIFF directive.

SALVE

Select an ILBM IFF file.  
See INCIFF .

### 1.180 Can't handle a BODY before BMHD Chunk

MESSAGE

Can't handle a BODY before BMHD Chunk

CAUSE

This error will occur when you try to include an IFF which has its chunk at the wrong place, or is simply missing chunks, with the INCIFF directive.

SALVE

Select an ILBM IFF file.  
See INCIFF .

### 1.181 Illegal Floating Point Size: should be B, W, L, Q, S, D, X or P

MESSAGE

Illegal Floating Point Size: should be B, W, L, Q, S, D, X or P

CAUSE

Strange ?

SALVE

There are no other sizes ???

### 1.182 Illegal Size for Data Register: should be B, W, L or S

MESSAGE

Illegal Size for Data Register: should be B, W, L or S

CAUSE

Strange ?

SALVE

There are no other sizes ???

---

### 1.183 Bcc.W Out of Range in MACRO

MESSAGE

Bcc.W Out of Range in MACRO

CAUSE

Bcc.W's within MACRO's are NOT optimized.

SALVE

Keep the MACRO within a 16 bit range.  
See MACRO .

### 1.184 Floating Point Register Expected

MESSAGE

Floating Point Register Expected

CAUSE

FPU opcodes always expect an FPU register as first or second operand AT LEAST..

SALVE

Possible are FP0 - FP7, for the rest see the

Register Names  
List.

### 1.185 This is not an ASM-One Project File

MESSAGE

This is not an ASM-One Project File

CAUSE

This error will occur when you try to load a NON .Aprj file with the 'RE' DLC.

SALVE

See

The 'RE' DLC  
.

### 1.186 Bitfield can't be Out of 32 Bit Range

MESSAGE

Bitfield can't be Out of 32 Bit Range

CAUSE



When specifying the offset and the width of a bitfield with immediates, the total of offset+width may not exceed 32.

SALVE

Legal is 0 to 31. If you are unsure, add the offset to the width.  
Remember: a width of ZERO is actually a 32 bit width !!!!

### 1.187 Illegal CMAP Mode Error: expecting 12, 24, CE or CA

MESSAGE

Illegal CMAP Mode Error: expecting 12, 24, CE or CA

CAUSE

This error will occur when you use anything else than 12 (12 Bit), 24 (24 Bit, CE (ECS Copper) or CA (AGA Copper) as 'mode' in the INCIFFP directive.

SALVE

Use 12, 24, CE or CA.  
See INCIFFP .

### 1.188 Illegal Color Offset (Bank) Specified

MESSAGE

Illegal Color Offset (Bank) Specified

CAUSE

This error will occur when you use the wrong 'offset' with the INCIFFP directive.

SALVE

See INCIFFP .

### 1.189 IFF Color(s) Out of Range

MESSAGE

IFF Color(s) Out of Range

CAUSE

Unknown.

SALVE

Unknown.

### 1.190 To Many Errors

---

## MESSAGE

To Many Errors

## CAUSE

If a source generates more than 100 errors/warnings when assembling, this error will be displayed.

## SALVE

Unselect 'Processor Warn'.

### 1.191 Invalid Function Code: only SFC, DFC, Dn or #<data> Allowed

## MESSAGE

Invalid Function Code: only SFC, DFC, Dn or #<data> Allowed

## CAUSE

If a MMU opcode uses a FC (Function Code) as operand, the FC should be one of the following types:

- SFC (Source Function Code Register)
- DFC (Destination Function Code Register)
- Dn (Data Register)
- #<data> (Immediate)

## SALVE

Check your Programmers Reference Manual.

### 1.192 Address Register Indirect Mode Expected, for Return Address Register

## MESSAGE

Address Register Indirect Mode Expected, for Return Address Register

## CAUSE

The PTESTR/W opcode expects another addressing mode for the Return Address on the 68030 than on the 68851.

For the 68030:

```
PTESTR/W FC,<ea>,#level,An ; DIRECT
```

For the 68851:

```
PTESTR/W FC,<ea>,#level,(An) ; INDIRECT
```

## SALVE

Check your Programmers Reference Manual.

### 1.193 Address Register Direct Mode Expected, for Return Address Register

---

**MESSAGE**

Address Register Direct Mode Expected, for Return Address Register

**CAUSE**

The PTESTR/W opcode expects another addressing mode for the Return Address on the 68030 than on the 68851.

For the 68030:

```
PTESTR/W FC,<ea>,#level,An ; DIRECT
```

For the 68851:

```
PTESTR/W FC,<ea>,#level,(An) ; INDIRECT
```

**SALVE**

Check your Programmers Reference Manual.

### 1.194 Level Out of 3 Bit Range

**MESSAGE**

Level Out of 3 Bit Range

**CAUSE**

The level used in an MMU opcode may NOT exceed the 3 bit range (0 - 7).

**SALVE**

Check your Programmers Reference Manual.

### 1.195 Mask Out of 3 Bit Range

**MESSAGE**

Mask Out of 3 Bit Range

**CAUSE**

The mask used in an 68030 MMU opcode may NOT exceed the 3 bit range (0 - 7).

**SALVE**

Check your Programmers Reference Manual.

### 1.196 Mask Out of 4 Bit Range

**MESSAGE**

Mask Out of 4 Bit Range

**CAUSE**

The mask used in an 68851 MMU opcode may NOT exceed the 4 bit range (0 - 15).

SALVE

Check your Programmers Reference Manual.

### **1.197 Function Code Out of 3 Bit Range**

MESSAGE

Function Code Out of 3 Bit Range

CAUSE

The FC used in an 68030 MMU opcode may NOT exceed the 3 bit range (0 - 7).

SALVE

Check your Programmers Reference Manual.

### **1.198 Function Code Out of 4 Bit Range**

MESSAGE

Function Code Out of 4 Bit Range

CAUSE

The FC used in an 68851 MMU opcode may NOT exceed the 4 bit range (0 - 7).

SALVE

Check your Programmers Reference Manual.

### **1.199 Insufficient Memory to change Source**

MESSAGE

Insufficient Memory to change Source

CAUSE

The 10 loaded sources are kept in memory. This error occurs when there is not enough memory to switch from the current source, to another source.

SALVE

Buy more memory.

### **1.200 MMU Register Expected as 1st Operand**

## MESSAGE

MMU Register Expected as 1st Operand

## CAUSE

You used a NON MMU Register as first operand.

## SALVE

Check your Programmers Reference Manual.

## 1.201 MMU Register Expected as 2nd Operand

## MESSAGE

MMU Register Expected as 2nd Operand

## CAUSE

You used a NON MMU Register as second operand.

## SALVE

Check your Programmers Reference Manual.

## 1.202 Invalid MMU Register as 1st Operand: CAL, VAL or SCC Expected

## MESSAGE

Invalid MMU Register as 1st Operand: CAL, VAL or SCC Expected

## CAUSE

PMOVE.B does only allow CAL, VAL or SCC as 1st operand.

## SALVE

Check your Programmer's Reference Manual.

## 1.203 Invalid MMU Register as 2nd Operand: CAL, VAL or SCC Expected

## MESSAGE

Invalid MMU Register as 2nd Operand: CAL, VAL or SCC Expected

## CAUSE

POVE.B does only allow CAL, VAL or SCC as 2nd operand.

## SALVE

Check your Programmer's Reference Manual.

## 1.204 Missing START address or Label !!

---

MESSAGE

Missing START address or Label !!

CAUSE

The

RB

DLC does expect a STARTing address.

SALVE

Give a STARTing address

### **1.205 Incorrect Index Register Specified. Only An, Dn or SP allowed**

MESSAGE

Incorrect Index Register Specified. Only An, Dn or SP allowed

CAUSE

See MESSAGE

SALVE

Use the An, Dn or the SP..

### **1.206 PC/ZPC can't be used as Index Register. Only An, Dn or SP allowed**

MESSAGE

PC/ZPC can't be used as Index Register. Only An, Dn or SP allowed

CAUSE

See MESSAGE

SALVE

Use the An, Dn or the SP...

### **1.207 We haven't come up with a nice text for this error Error !!!!**

MESSAGE

We haven't come up with a nice text for this error Error !!!!

CAUSE

An error that we haven't figured out what actually heppens yet.

SALVE

Wait for a next release of ASM-One

### **1.208 The use of ZPC is not allowed in this addressing mode !!**

---

MESSAGE

The use of ZPC is not allowed in this addressing mode !!

CAUSE

You used the ZPC in a mode that doesn't allow that

SALVE

Well, I can't think of an addressing mode at the moment, to be honest B-)

## 1.209 Field Width Expected: {offset:WIDTH}

MESSAGE

Field Width Expected: {offset:WIDTH}

CAUSE

You didn't specify a WIDTH.

SALVE

Check your Programmer's Reference Manual

## 1.210 Field Width can't be ZERO

MESSAGE

Field Width can't be ZERO

CAUSE

If you used an immediate operand to specify the width, only numbers from 1 to 31 are allowed.

SALVE

Not used at the moment...

## 1.211 Warnings

ASM-One: Warnings

-----

ASM-One knows some warnings, which are genarly generated while assembling a source.

ASM-One will not stop assembling when it generates a warning.

Here's the list of known warnings:

>> Warning << Value Signed Extended to Longword

>> Warning << 68010++ Command Used

```
>> Warning << 68020++ Command Used
>> Warning << 68030++ Command Used
>> Warning << 68040++ Command Used
>> Warning << 68010 Specific Command Used
>> Warning << 68020 Specific Command Used
>> Warning << 68030 Specific Command Used
>> Warning << 68040 Specific Command Used
>> Warning << 68060 Specific Command Used
>> Warning << 68020/030/040 Specific Command Used
>> Warning << 68020/030 Specific Command Used
>> Warning << MOVEP not supported by the 68060
>> Warning << 68030/68851 Command Used
>> Warning << 68051 Command Used
>> Warning << 68881/68882 Command Used
>> Warning << PFLUSHA Assembled with 68030/68851
>> Warning << PFLUSHA Assembled with 68040
```

## 1.212 >> Warning <<, Value Signed Extended to Longword

### MESSAGE

```
>> Warning <<, Value Signed Extended to Longword
```

### CAUSE

```
Implemented but not used at the moment.
```

## 1.213 >> Warning << 68010++ Command Used

### MESSAGE

```
>> Warning << 68010++ Command Used
```

### CAUSE

```
You used an 68010++ command, which could crash an
68000 CPU.
```

### SALVE

---



Check your Programmers Reference Manual.

### **1.214 >> Warning << 68020++ Command Used**

MESSAGE

>> Warning << 68020++ Command Used

CAUSE

You used an 68020++ command, which could crash an 68010 or lower CPU.

SALVE

Check your Programmers Reference Manual.

### **1.215 >> Warning << 68030++ Command Used**

MESSAGE

>> Warning << 68030++ Command Used

CAUSE

You used an 68030++ command, which could crash an 68020 or lower CPU.

SALVE

Check your Programmers Reference Manual.

### **1.216 >> Warning << 68040++ Command Used**

MESSAGE

>> Warning << 68040++ Command Used

CAUSE

You used an 68040++ command, which could crash an 68030 or lower CPU.

SALVE

Check your Programmers Reference Manual.

### **1.217 >> Warning << 68010 Specific Command Used**

MESSAGE

>> Warning << 68010 Specific Command Used

CAUSE

You used an 68010 specific command, which could crash other 680x0 CPU's.

---

SALVE

Check your Programmers Reference Manual.

## 1.218 >> Warning << 68020 Specific Command Used

MESSAGE

>> Warning << 68020 Specific Command Used

CAUSE

You used an 68020 specific command, which could crash other 680x0 CPU's.

SALVE

Check your Programmers Reference Manual.

## 1.219 >> Warning << 68030 Specific Command Used

MESSAGE

>> Warning << 68030 Specific Command Used

CAUSE

You used an 68030 specific command, which could crash other 680x0 CPU's.

SALVE

Check your Programmers Reference Manual.

## 1.220 >> Warning << 68040 Specific Command Used

MESSAGE

>> Warning << 68040 Specific Command Used

CAUSE

You used an 68040 specific command, which could crash other 680x0 CPU's.

SALVE

Check your Programmers Reference Manual.

## 1.221 >> Warning << 68060 Specific Command Used

MESSAGE

>> Warning << 68060 Specific Command Used

CAUSE

You used an 68060 specific command, which could crash other 680x0 CPU's.

---

SALVE

Check your Programmers Reference Manual.

## **1.222 >> Warning << 68020/030/040 Specific Command Used**

MESSAGE

>> Warning << 68020/030/040 Specific Command Used

CAUSE

When using MOVEC, the following registers are NOT supported by the 68060, or CPU's lower than the 68020:

CAAR  
MSP  
ISP  
MMUSR

SALVE

Check your Programmers Reference Manual.

## **1.223 Warning << 68020/030 Specific Command Used**

MESSAGE

Warning << 68020/030 Specific Command Used

CAUSE

When using MOVEC, the CAAR is only allowed on the 68020 and 68030.

SALVE

Check your Programmers Reference Manual.

## **1.224 >> Warning << MOVEP not supported by the 68060**

MESSAGE

>> Warning << MOVEP not supported by the 68060"

CAUSE

The MOVEP opcode is NOT supported by the 68060, and will therefore generate an Illegal Instruction Trap if executed on the 68060.

SALVE

MOVEP will be assembled according to the 68010/20/30/40.  
Check your Programmers Reference Manual.

---

**1.225 >> Warning << 68030/68851 Command Used**

## MESSAGE

>> Warning << 68030/68851 Command Used

## CAUSE

You used an 68030/68851 command, which could crash other 68k MMU's.

## SALVE

Check your Programmers Reference Manual.

**1.226 >> Warning << 68851 Command Used**

## MESSAGE

>> Warning << 68851 Command Used

## CAUSE

You used an 68851 command, which could crash other 68k MMU's.

## SALVE

Check your Programmers Reference Manual.

**1.227 >> Warning << 68881/68882 Command Used**

## MESSAGE

>> Warning << 68881/68882 Command Used

## CAUSE

You used an 68881/68882 command, which could crash other 68k FPU's.

## SALVE

Check your Programmers Reference Manual.

**1.228 >> Warning << PFLUSHA Assembled with 68030/68851**

## MESSAGE

>> Warning << PFLUSHA Assembled with 68030/68851

## CAUSE

PFLUSHA is assembled according to the 68030/68851 specifications.

## SALVE

Check your Programmers Reference Manual.

---

## 1.229 >> Warning << PFLUSHA Assembled with 68040

### MESSAGE

>> Warning << PFLUSHA Assembled with 68040

### CAUSE

PFLUSHA is assembled according to 68040/68060 specifications.

### SALVE

Check your Programmers Reference Manual.

## 1.230 Register Names & Private Symbols

### Register Names & Private Symbols

Some Names are used by ASM-One itself, and may NOT be used by the programmers.

Using those symbols will generally generate the following error:

```

Reserved Word
Most registers are only valid on certain processors. Check the

Valid Register Tabel
on which processor registers are valid.

```

#### Register Names:

```

An - Address Register (n stands for a number from 0 to 7)
Dn - Data Register (n stands for a number from 0 to 7)
FPn - FPU Register (n stands for a number from 0 to 7)
BADn - MMU Register (n stands for a number from 0 to 7)
BACn - MMU Register (n stands for a number from 0 to 7)
SR - Status Register
CCR - Condition Code Register
USP - User Stack Pointer
SFC - Source Function Code Register
DFC - Destination Function Code Register
CACR - Cache Control Register
CAAR - Cache Address Register
TC - Translation Control Register
ITT0 - Instruction Transparant Translation Register 0
ITT1 - Instruction Transparant Translation Register 1
IACR0 - Instruction Access Control Register 0
IACR1 - Instruction Access Control Register 1
DTT0 - Data Transparant Translaton Register 0
DTT1 - Data Transparant Translation Register 1
DACR0 - Data Access Control Register 0
DACR1 - Data Access Control Register 1
FPIAR - Floating-Point Instruction Address Register
FPSR - Floating-Point Status Register

```

FPCR - Floating-Point Control Register  
 VBR - Vector Base Register  
 MSP - Master Stack Pointer Register  
 ISP - Interrupt Stack Pointer Register  
 URP - User Root Pointer Register  
 SRP - Supervisor Root Pointer Register  
 DRP - DMA Root Pointer Register  
 CRP - CPU Root Pointer Register  
 CAL - Current Access Level Register  
 VAL - Valid Access Level Register  
 SCC - Stack Change Control Register  
 AC - Access Control Register (AC0 en AC1 ?)  
 MMUSR - Memory Management Unit Status Register  
 TT0 - Transparent Translation Register 0  
 TT1 - Transparent Translation Register 1  
 PSR - PMMU Status Register  
 PCSR - PMMU Control Register  
 PCR - Processor Configuration Register  
 BUSCR - Bus Control Register  
 (S)SP - Supervisor Stack Pointer Register

Private Symbols:

ZPC - Zero PC Notation  
 FileSize- Used to determine the FilzeSize()

### 1.231 Valid Register Tabel

Some registers have the same meaning for different processors. It's important that you use the registers as described here, because ASM-One will generate warnings & errors according to the used registers in some cases (mainly for the MMU) rather than the processor chosen in the Preferences.

	68000	68010	68020	68030	68040	68060	68851	68882
An	x	x	x	x	x	x		
Dn	x	x	x	x	x	x		
SR	x	x	x	x	x	x		
CCR	x	x	x	x	x	x		
USP	x	x	x	x	x	x		
ISP	x	x	x	x	x			
(S)SP	x	x	x	x	x	x		
VBR		x	x	x	x	x		
SFC		x	x	x	x	x		
DFC		x	x	x	x	x		
CACR			x	x	x	x		
CAAR			x	x	x	x		
MSP			x	x	x			
TC			x	x	x	x		
URP			x	x	x	x		
SRP			x	x	x	x		
ITT0				x	x		Not 68EC040	
ITT1				x	x		Not 68EC040	

DTT0						x		x				Not 68EC040
DTT1						x		x				Not 68EC040
FPIAR						x		x			x	
FPSR						x		x			x	
FPCR						x		x			x	
FPn						x		x			x	
PCR						x						
BUSCR						x						
MMUSR						x						
TT0						x						
TT1						x						
CRP						x				x		
BADn								x				
BACn								x				
DRP								x				
CAL								x				
VAL								x				
SCC								x				
AC								x				
PSR								x				
PCSR								x				
AC0												68EC030
AC1												68EC030
ACUSR												68EC030
IACR0												68EC040
IACR1												68EC040
DACR0												68EC040
DACR1												68EC040

Note: ASM-One doesn't support the LC and EC versions of the 68030, 68040 and the 68060.

## 1.232 ASM-One.Pref File

ASM-One.Pref File

The ASM-One.Pref file holds all the preferences that can be set and saved.

Sometimes it's handy to know WHAT it all means, so you can change preferences without starting ASM-One (or maybe you changed something so you can't start ASM-One anymore B-)

ASM-One also has some default preferences build-in. These are the preferences I (Boushh) use myself. They are ONLY used when no ASM-One.Pref file was found (it should be located in the ENVARC: directory).

Once you made a comfortable ASM-One.Pref file, make a copy ! It may come in handy..

Here are the possible codes that could be in your ASM-One.Pref file: (a PLUS sign means: Active, a MINUS sign means: INactive).

RS

--

Rescue Source - Believe it or not, but with this on ASM-One will try to save your source, so you can retrieve it using the

0  
DLC.

L7

--

Level 7 - Will start ASM-One's own Level 7 code. A level 7 interrupt can only be activated by adding external/extra hardware to your Amiga (like the Action Replay Cartridge).

NL

--

NumLock - If ON, the keypad will function like cursor keys:

7 - Top    8 - Up    9 - Page Up  
4 - Left   5 - Center   6 - Right  
1 - Bottom   2 - Down   3 - Page Down

AA

--

AutoAlloc - if ON. ASM-One will locate the CORRECT memory for your section. If OFF: your sections will be placed where there is room in memory. It's a good thing to set this option ON, specially when you work with ChipMem.

RL

--

ReqTools - If ON, ASM-One will use the ReqTools.library for all it's requesters. If OFF, you will save some 8 Kb of memory.

XR

--

Extended ReqTools - If ON, also the search/replace functions will have requesters (instead of being in the menubar).

PD

--

Printer Dump - This option will activate the dumping of text to the printer. This includes the DLC's.

LF

--

List File - If ON, will list the complete file, with line numbers, addresses, the assembled code, the source as comment and a list of all used symbols.

---



PG

--

Paging - If ON, will start a new page after the previous one was full.

HP

--

Halt Page - If ON, will stop after the screen has been filled with data (in combination with Paging).

AE

--

All Errors - If ON, will assemble, regardless of the errors found. ASM-One will keep the location of 100 errors/warnings, after assembly you can reach them with Amiga+e.

DB

--

Debug - If ON, could spare time when debugging. Because ASM-One will NOT reassemble the source if nothing was changed.

IL

--

InterLaced - If ON, will activate InterLace (not fully functional anymore in R399e, will be skipped in later releases).

1B

--

One Bitplane - If ON, will start ASM-One with only ONE bitplane, to save memory (partly functional in R399e, will be skipped in the next release).

.S

--

Source .S - If ON, will use the specified source-extension when writing source.

CW

--

Close WorkBench - If ON, ASM-One will try to close the WorkBench to save memory.

LN

--

Line Numbers - If ON, shows line numbers.

AI

---

---

--

Auto Indent - If ON, will set the tabs for you.

L:

--

Label : - If ON, ALL labels have to be ending with a colon.

DA

--

DisAssemble - If ON, will show the next line you want to debug, disassembled in the debugger.

SS

--

ShowSource - If ON, will change the debugger in a SourceLevel Debugger. If OFF, the actual assembled code will be shown in the debugger, instead of the Source.

OA

--

Only ASCII - If ON, will skip unreadable characters when viewing memory in ASCII.

UL

--

UpperCase = LowerCase - If ON, ASM-One will make no difference between Upper- and LowerCase. Should be OFF when using the Official Amiga includes !!!!

;C

--

;Comment - If ON, comments HAVE to be preceded by a semi-colon.

PI

--

Progress Indicator - If ON, will show how much ASM-One has assembled of your source.

PL

--

Progress Line - If ON, ASM-One will show which line is assembled.

PS

--

Startup Parameters - If ON, will allow you to debug sources like if they were executed from the CLI with parameters. Use the

---

PS  
DLC to set the parameters.

RR  
--

Resident Registers - If ON, will make the REGSDATA file resident in memory if called upon the first time. If OFF, the REGSDATA file will be closed after each action (see the =R DLC).

AU  
--

Update Check - If ON, ASM-One will only update sources that have been changed. If OFF, ASM-One will ALWAYS update the source.

SM  
--

Save Marks - If ON, will save the 10 marks you can make in the source. If this is on, and you want to save a source without marks, use the

WN  
DLC.

FW/PW  
--

Processor Warn - If ON, Processor warns will not be displayed.

FP  
--

FPU Present - If ON, makes ASM-One believe you have a real FPU in your Amiga (so ASM-One will skip FPU specific warnings and errors).

OD  
--

68020++ Odd Data - If ON, ASM-One will NOT give any errors if data (and ONLY data) is at an odd address.

DC  
--

DS Clear - If ON, ASM-One will clear BSS sections for you automatically. But remember, the Amiga will NOT clear BSS sections for you if you start your program outside ASM-One.

UD  
--

CTRL Up/Down - If ON, using the CTRL key in combination with the cursor keys, change their function:

---

CTRL Up/Down OFF CTRL Up/Down ON

Up One line Up To the Beginning of the source  
Down One line Down To the Bottom of the source

EP  
--

Enable/Permit - If ON, ASM-One will call Enable/Permit before  
executing a program within ASM-One

LD  
--

Libcalls dec - If ON, will show library calls in decimal instead of  
hexadecimal numbers.

KX  
--

Keep X - If ON, ASM-One will try to keep the cursor in the same  
colom when going to another line.

MP  
--

68851 Present - If ON, ASM-One will belive you actualy have  
an 68851 (MMU) on board.

SO  
--

Safety - If ON, there will be NO default option when a requester  
pops up (normaly RETURN will activate the default option).

RD  
--

Realtime Deb - Unkown

WF  
--

WorkBench to Front - If ON, will bring the WorkBench to front when  
you execute a program.

!.xxx  
-----

Source Extension: If you use .S, then this is the place to specify  
which extension should be used. The 'xxx' should be the 3 letter  
extension.

\*XXXXXXXXX  
-----

---

Default Directory - If specified, ASM-One will change to this directory on startup. The 'xxxxxx' should be the directory to change too.

```
|x1x|x2x|x3x|x4x|
-----
```

Colors - These are the colors used by ASM-One:

- 1)
- 2)
- 3)
- 4)

```
[CPUx
-----
```

Chosen CPU - States the CPU you have chosen in your prefs:

```
0 = 68000
1 = 68010
2 = 68020
3 = 68030
4 = 68040
5 = 68060
```

```
[MMUx
-----
```

Unknown

```
^xxxxxx|xxxxxx|xxxxxx|
-----
```

Selected Screen Mode - First the ScreenMode ID, then the screen height, and then the screen width. All in hexadecimal.

```
\<text>
-----
```

BootUp string - A string (command(s)) ASM-One will execute at startup.

```
&xxx.font
-----
```

Default Font - Specifies the selected font for ASM-One.

And that where ALL options currently available. These options can change in the future. We don't guarantee that an option will be available in the next version of ASM-One

---

### 1.233 M68000 Family Instruction Set

M68000 Family Instruction Set

-----

Here is a quick overview of all known instructions.

	68000	68010	68020	68030	68040	68060	68881 68851	68882	
ABCD	x	x	x	x	x	x	x		
ADD	x	x	x	x	x	x	x		
ADDA	x	x	x	x	x	x	x		
ADDI	x	x	x	x	x	x	x		
ADDQ	x	x	x	x	x	x	x		
ADDX	x	x	x	x	x	x	x		
AND	x	x	x	x	x	x	x		
ANDI	x	x	x	x	x	x	x		
ANDI to CCR	x	x	x	x	x	x	x		
ANDI to SR\$^1\$	x	x	x	x	x	x	x		
ASL, ASR	x	x	x	x	x	x	x		
Bcc	x	x	x	x	x	x	x		
BCHG	x	x	x	x	x	x	x		
BCLR	x	x	x	x	x	x	x		
BFCHG			x	x	x	x	x		
BFCLR			x	x	x	x	x		
BFEXTS			x	x	x	x	x		
BFEXTU			x	x	x	x	x		
BFFFO			x	x	x	x	x		
BFINS			x	x	x	x	x		
BFSET			x	x	x	x	x		
BGND									
BKPT		x	x	x	x	x	x		
BRA	x	x	x	x	x	x	x		
BSET	x	x	x	x	x	x	x		
BSR	x	x	x	x	x	x	x		
BTST	x	x	x	x	x	x	x		
CALLM			x						
CAS, CAS2			x	x	x	x	x, 3		
CHK	x	x	x	x	x	x	x		
CHK2			x	x	x	x	3		
CINV\$^1\$						x	x		
CLR	x	x	x	x	x	x	x		
CMP	x	x	x	x	x	x	x		
CMPA	x	x	x	x	x	x	x		

### 1.234 M68000 Family Instruction Set

	68000	68010	68020	68030	68040	68060	68881 68851	68882	
CMPI	x	x	x	x	x	x	x		
CMPM	x	x	x	x	x	x	x		
CMP2			x	x	x	x	3		

cpBcc				x	x							
cpDBcc				x	x							
cpGEN				x	x							
cpRESTORE\$^1\$				x	x							
cpSAVE\$^1\$				x	x							
cpScc				x	x							
cpTRAPcc				x	x							
CPUSH\$^1\$							x	x				
DBcc	x	x	x	x	x	x	x					
DIVS	x	x	x	x	x	x	x,3					
DIVSL	x	x	x	x	x	x	x					
DIVU	x	x	x	x	x	x	x,3					
DIVUL	x	x	x	x	x	x	x					
EOR	x	x	x	x	x	x	x					
EORI	x	x	x	x	x	x	x					
EORI to CCR	x	x	x	x	x	x	x					
EORI to SR\$^1\$	x	x	x	x	x	x	x					
EXG	x	x	x	x	x	x	x					
EXT	x	x	x	x	x	x	x					
EXTB				x	x	x	x					
FABS							x,2	x,2			x	
FSABS, FDABS							x,2	x,2				
FACOS							2,3	2,3			x	
FADD							x,2	x,2			x	
FSADD, FDADD							x,2	x,2				
FASIN							2,3	2,3			x	
FATAN							2,3	2,3			x	
FATANH							2,3	2,3			x	
FBcc							x,2	x,2			x	
FCMP							x,2	x,2			x	
FCOS							2,3	2,3			x	
FCOSH							2,3	2,3			x	
FDBcc							x,2	x,2			x	
FDIV							x,2	x,2			x	
FSDIV, FDDIV							x,2	x,2				
FETOX							2,3	2,3			x	
FETOXM1							2,3	2,3			x	

### 1.235 M68000 Family Instruction Set

	68000	68010	68020	68030	68040	68060	68851	68881	68882
FGETEXP						2,3	2,3		x
FGETMAN						2,3	2,3		x
FINT						2,3	x,2		x
FINTRZ						2,3	x,2		x
FLOG10						2,3	2,3		x
FLOG2						2,3	2,3		x
FLOGN						2,3	2,3		x
FLOGNP						2,3	2,3		x
FMOD						2,3	2,3		x
FMOVE						x,2	x,2		x
FSMOVE						x,2	x,2		
FDMOVE						x,2	x,2		

FMOVECR						2,3	2,3			x	
FMOVEM						x,2	x,2,3			x	
FMUL						x,2	x,2			x	
FSMUL, FDMUL						x,2	x,2				
FNEG						x,2	x,2			x	
FSNEG, FDNEG						x,2	x,2				
FNOP						x,2	x,2			x	
FREM						2,3	2,3			x	
FRESTORE\$^1\$						x,2	x,2			x	
FSAVE\$^1\$						x,2	x,2			x	
FSCALE						2,3	2,3			x	
FScC						x,2	2,3			x	
FSGLDIV						2,3	2,3			x	
FSGLMUL						2,3	2,3			x	
FSIN						2,3	2,3			x	
FSINCOS						2,3	2,3			x	
FSINH						2,3	2,3			x	
FSQRT						x,2	x,2			x	
FSSQRT						x,2	x,2				
FDSQRT						x,2	x,2				
FSUB						x,2	x,2			x	
FSSUB, FDSUB						x,2	x,2				
FTAN						2,3	2,3			x	
FTANH						2,3	2,3			x	
FTENTOX						2,3	2,3			x	
FTRAPcc						2,3	2,3			x	
FTST						x,2	x,2			x	
FTWOTOX						2,3	2,3			x	

### 1.236 M68000 Family Instruction Set

	68000	68010	68020	68030	68040	68060	68851	68881	68882	
ILLEGAL	x	x	x	x	x	x	x			
JMP	x	x	x	x	x	x	x			
JSR	x	x	x	x	x	x	x			
LEA	x	x	x	x	x	x	x			
LINK	x	x	x	x	x	x	x			
LPSTOP							x			
LSL, LSR	x	x	x	x	x	x	x			
MOVE	x	x	x	x	x	x	x			
MOVEA	x	x	x	x	x	x	x			
MOVE from CCR			x	x	x	x	x			
MOVE to CCR	x		x	x	x	x	x			
MOVE from SR\$^1\$	4		x	x	x	x	x			↵
MOVE to SR\$^1\$	x		x	x	x	x	x			↵
MOVE USP\$^1\$	x		x	x	x	x	x			↵
MOVE16						x	x			
MOVEC\$^1\$			x	x	x	x	x			↵
MOVEM	x	x	x	x	x	x	x			





SBCD		x		x		x		x		x		x					
Scc		x		x		x		x		x		x					
STOP\$^1\$		x		x		x		x		x		x					
SUB		x		x		x		x		x		x					
SUBA		x		x		x		x		x		x					
SUBI		x		x		x		x		x		x					
SUBQ		x		x		x		x		x		x					
SUBX		x		x		x		x		x		x					
SWAP		x		x		x		x		x		x					
TAS		x		x		x		x		x		x					
TBLS, TBLSN																	
TBLU, TBLUN																	
TRAP		x		x		x		x		x		x					
TRAPcc						x		x		x		x					
TRAPV		x		x		x		x		x		x					
TST		x		x		x		x		x		x					
UNLK		x		x		x		x		x		x					
UNPK						x		x		x		x					

**NOTE:**

- 1 = Privileged (Supervisor) Instruction
- 2 = Not applicable to the MC68EC040, MC68LC040, MC68EC060 and MC68LC060
- 3 = These are software-supported instructions on the MC68040 and MC68060
- 4 = This instruction is not privileged for the MC68000 and MC68008
- 5 = Not applicable to MC68EC030
- 6 = All MC68060 and MC68040 Floating-point instructions require software assistance for unimplemented data types (MC68040 and MC68060) and unimplemented effective addresses (MC68060 only)

Copyright MOTOROLA, 1994

## 1.238 M68000 Family Mnemonic Description

### M68000 Family Mnemonic Description

-----

If you would program in Assembler some 15 years ago, you would be programming numbers. This is because the processor ONLY knows the number 0 and 1 (the binary number system).

Programming in 0's and 1's is a painstaking job, so people invented the Octal number system (representing 8 binary numbers) and the Hexadecimal number system (representing 16 binary numbers).

But people still used numbers to program in assembler.

Then someone had the brilliant idea of giving names to the hexadecimal numbers: Mnemonics...

So Mnemonics are alphabetical representations of numbers. And they allow today's assembler programmers to program in assembler quite easily.

An Assembler (like ASM-One) translates the Mnemonics to numbers again, so the processor understands them.

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
ADDA	Add Address
ADDI	Add Immediate
ADDQ	Add Quick
ADDX	Add with Extend
AND	Logical AND
ANDI	Logical AND Immediate
ANDI to CCR	AND Immediate to Condition Code Register
ANDI to SR	AND Immediate to Status Register
ASL, ASR	Arithmetic Shift Left and Right

### 1.239 M68000 Family Mnemonic Description

Mnemonic	Description
Bcc	Branch Conditionally
BCHG	Test Bit and Change
BCLR	Test Bit and Clear
BFCHG	Test Bit Field and Change
BFCLR	Test Bit Fields and Clear
BFEXTS	Signed Bit Field Extract
BFEXTU	Unsigned Bit Field Extract
BFFFO	Bit Field Find First One
BFINS	Bit Field Insert
BFSET	Test Bit Field and Set
BFTEST	Test Bit Field
BGND	Enter Background Mode
BKPT	Breakpoint
BRA	Branch
BSET	Test Bit and Set
BSR	Branch to Subroutine
BTST	Test Bit
CALLM	CALL Module
CAS	Compare and Swap Operands
CAS2	Compare and Swap Dual Operands
CHK	Check Register Against Bound
CHK2	Check Register Against Upper and Lower Bounds
CINV	Invalidate Cache Entries
CLR	Clear
CMP	Compare
CMPA	Compare Address
CMPI	Compare Immediate
CMPM	Compare Memory to Memory
CMP2	Compare Register Against Upper and Lower Bounds
cpBcc	Branch on Coprocessor Condition
cpDBcc	Test Coprocessor Condition Decrement and Branch
cpGEN	Coprocessor General Function

cpRESTORE	Coprocessor Restore Function
cpSAVE	Coprocessor Save Function
cpScc	Set on Coprocessor Condition
cpTRAPcc	Trap on Coprocessor Condition
-----+	
DBcc	Test Condition, Decrement and Branch
DIVS, DIVSL	Signed Divide
DIVU, DIVUL	Unsigned Divide
-----+	

## 1.240 M68000 Family Mnemonic Description

Mnemonic	Description
-----+	
EOR	Logical Exclusive-OR
EORI	Logical Exclusive-OR Immediate
EORI to CCR	Exclusive-OR Immediate to Condition Code Register
EORI to SR	Exclusive-OR to Status Register
EXG	Exchange Registers
EXT, EXTB	Sign Extend
-----+	
FABS	Floating-Point Absolute Value
FSABS, FDABS	Floating-Point Absolute Value (Single/Double Precision)
FACOS	Floating-Point Arc Cosine
FADD	Floating-Point Add
FSADD, FDADD	Floating-Point Add (Single/Double Precision)
FASIN	Floating-Point Arc Sine
FATAN	Floating-Point Arc Tangent
FATANH	Floating-Point Hyperbolic Arc Tangent
FBcc	Floating-Point Branch
FCMP	Floating-Point Compare
FCOS	Floating-Point Cosine
FCOSH	Floating-Point Hyperbolic Cosine
FBDcc	Floating-Point Decrement and Branch
FDIV	Floating-Point Divide
FSDIV, FDDIV	Floating-Point Divide (Single/Double Precision)
FETOX	Floating-Point $e^{\$}$
FETOXM1	Floating-Point $e^{\$-1}$
FGETEXP	Floating-Point Get Exponent
FGETMAN	Floating-Point Get Mantissa
FINT	Floating-Point Integer Part
FINTRZ	Floating-Point Integer Part, Round-to-Zero
FLOG10	Floating-Point $\log_{10}$
FLOG2	Floating-Point $\log^2$
FLOGN	Floating-Point Loge
FLOGNP1	Floating-Point $\log_e(\$ \times \$^1)$
FMOD	Floating-Point Modulo Remainder
FMOVE	Move Floating-Point Register
FSMOVE, FDMOVE	Move Floating-Point Register (Single/Double Precision)
FMOVECR	Move Constant ROM
FMOVEM	Move Multiple Floating-Point Registers
FMUL	Floating-Point Multiply
FSMUL, FDMUL	Floating-Point Multiply (Single/Double Precision)
FNEG	Floating-Point Negate

FSNEG, FDNEG | Floating-Point Negate (Single/Double Precision)

## 1.241 M68000 Family Mnemonic Description

Mnemonic	Description
FNOP	Floating-Point No-Operation
FREM	IEEE Remainder
FRESTORE	Restore Floating-Point Internal State
FSAVE	Save Floating-Point Internal State
FSCALE	Floating-Point Scale Exponent
FScc	Floating-Point Set According to Condition
FSGLDIV	Single-Precision Divide
FSGLMUL	Single Precision Multiply
FSIN	Sine
FSINCOS	Simultaneous Sine and Cosine
FSINH	Hyperbolic Sine
FSQRT	Floating-Point Square Root
FSSQRT, FDSQRT	Floating-Point Square Root (Single/Double Precision)
FSUB	Floating-Point Subtract
FSSUB, FDSUB	Floating-Point Subtract (Single/Double Precision)
FTAN	Tangent
FTANH	Hyperbolic Tangent
FTENTOX	Floating-Point 10 <sup>x</sup>
FTRAPcc	Floating-Point Trap on Condition
FTST	Floating-Point Test
FTWOTOX	Floating-Point 2 <sup>x</sup>
ILLEGAL	Take Illegal Instruction Trap
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link and Allocate
LPSTOP	Low-Power Stop
LSL, LSR	Logical Shift Left and Right
MOVE	Move
MOVEA	Move Address
MOVE from CCR	Move from Condition Code Register
MOVE from SR	Move from Status Register
MOVE to CCR	Move to Condition Code Register
MOVE to SR	Move to Status Register
MOVE USP	Move User Stack Pointer
MOVE16	16-Byte Block Move
MOVEC	Move Control Register

## 1.242 M68000 Family Mnemonic Description

Mnemonic	Description
----------	-------------

MOVEM	Move Multiple Registers
MOVEP	Move Peripheral
MOVEQ	Move Quick
MOVES	Move Alternate Address Space
MULS	Sign Multiply
MULU	Unsigned Multiply
-----+-----	
NBCD	Negate Decimal with Extend
NEG	Negate
NEGX	Negate with Extend
NOP	No Operation
NOT	Logical Complement
-----+-----	
OR	Logical Inclusive-OR
ORI	Logical Inclusive-OR Immediate
ORI to CCR	Inclusive-OR Immediate to Condition Code Register
ORI to SR	Inclusive-OR Immediate to Status Register
-----+-----	
PACK	Pack BCD
PBcc	Branch on PMMU Condition
PDBcc	Test, Decrement and Branch on PMMU Condition
PEA	Push Effective Address
PFLUSH	Flush Entry(ies) in the ATCs
PFLUSHA	Flush Entry(ies) in the ATCs
PFLUSHR	Flush Entry(ies) in the ATCs and RPT Entries
PFLUSHS	Flush Entry(ies) in the ATCs
PLOAD	Load an Entry into ATC
PLPA	Load Physical Address
PMOVE	Move PMMU Register
PRESTORE	PMMU Restore Function
PSAVE	PMMU Save Function
PScc	Set on PMMU Condition
PTEST	Test a Logical Address
PTRAPcc	Trap on MMU Condition
PVALID	Validate a Pointer
-----+-----	
RESET	Reset External Devices
ROL, ROR	Rotate Left and Right
ROXL, ROXR	Rotate with Extend Left and Right
RTD	Return and Deallocate
RTE	Return from Exception

## 1.243 M68000 Family Mnemonic Description

Mnemonic	Description
-----+-----	
RTM	Return from Module
RTR	Return and Restore
RTS	Return from Subroutine
-----+-----	
SBCD	Subtract Decimal with Extend
Scc	Set Conditionally
STOP	Stop
SUB	Subtract
SUBA	Subtract Address

```

SUBI    | Subtract Immediate
SUBQ    | Subtract Quick
SUBX    | Subtract with Extend
SWAP    | Swap Register Words
-----+-----
TAS     | Test Operand and Set
TBL$S, TBL$SN | Signed Table Lookup with Interpolate
TBL$U, TBL$LUN | Unsigned Table Lookup with Interpolate
TRAP    | Trap
TRAPcc  | Trap Conditionally
TRAPV   | Trap on Overflow
TST     | Test Operand
-----+-----
UNLK    | Unlink
UNPK    | Unpack BCD
-----+-----

```

## 1.244 Contacting the Author

Contacting the Author  
-----

Remco Weg (aka Boushh of TFA) likes you to contact him by E-Mail:

remco.weg@endemol.nl

## 1.245 Thanks to....

Thanks to...  
-----

I like to thank:

- Rune Gram-Madsen, for the most excelent ASM-One V1.01
- Price, for developing ASM-One for the last 6 years
- All those people who use ASM-One (you keep me going B-)
- Amiga Inc., for developing one of the (if not THE) greatest computers of all time ↔

My taglines:

- Keep supporting the Amiga.
- TFA, Artists with an Attitude