

**proopts**

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

	<i>TITLE :</i> proopts	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		July 1, 2022
<i>SIGNATURE</i>		

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

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>proopts</b>	<b>1</b>
1.1	proopts.guide . . . . .	1
1.2	about this manual . . . . .	2
1.3	notices . . . . .	3
1.4	system requirements . . . . .	3
1.5	introduction . . . . .	4
1.6	running proopts . . . . .	4
1.7	using proopts . . . . .	4
1.8	pro.guide/Assembler Options . . . . .	7
1.9	pro.guide/Code Options . . . . .	8
1.10	pro.guide/Default Options . . . . .	10
1.11	pro.guide/Miscellaneous Options . . . . .	10
1.12	pro.guide/Optimization Options . . . . .	11
1.13	pro.guide/List Options . . . . .	13
1.14	pro.guide/Linker Options . . . . .	14
1.15	author . . . . .	15

---

# Chapter 1

## proopts

### 1.1 proopts.guide

---

P R O O P T S

v1.0

---

Copyright © 1993–1996 by Daniel Weber

\*\*\* This Manual \*\*\*

TABLE OF CONTENTS

---

Notices

Introduction

System Requirements

Running ProOpts

Using ProOpts

Assembler Options

Code Options

Default Options

Miscellaneous Options

Optimization Options

List Options

Linker Options

---

## Acknowledgments

Contacting the Author

Additional Manuals:

ProAsm      ASX      ProUtils

## 1.2 about this manual

About This Manual

=====

Welcome to the AmigaGuide version of the ProOpts manual.

This manual is a part of the ProAsm manual, which was originally written in TeX and texinfo. You are currently looking at the AmigaGuide version of the ProOpts part. I used various tools to convert the primal TeX source into AmigaGuide. They did quite a good job, but the result was over 700KBytes of length! As a consequence I had to shorten the whole guide file by hand. Now it is very possible that when browsing through this manual you will find sentences like:

This node (page) intentionally left blank.

or

This node (page) was intentionally shortened.

Please forgive me for that. I had also to skip various tables that can be found in the original printed manual.

Caused by the conversion of the TeX source of the manual into this AmigaGuide file and the need of some small reorganisation of it, some of the links go nowhere. Sorry for that, but I had not the time to check all links. I did check a lot of them, but unfortunately there are still some of these dead links in the guide file. If you should find any, please feel free to inform me (

Daniel Weber  
)

If you find some strange (unusal) parts in this AmigaGuide file, please keep in mind, that this guide initially was written as a book. You can order a copy of the printed manual, if you wish. Please read the registration document (register.doc or Registration) for further information.

Daniel Weber  
Zurich  
November 1995

## 1.3 notices

### Copyrights

=====

ProOpts is copyrighted 1993-1996 by Daniel Weber. All rights are reserved worldwide.

The ProOpts User's Manual is copyrighted 1994-1996 by Daniel Weber and Bryan Ford.

This AmigaGuide file reflects version 1.0 of ProOpts.

### Trademarks

=====

ProAsm and ASX are trademarks of Daniel Weber. MC68000, MC68008, MC68010, MC68020, MC68030, MC68040, MC68060 MC68EC020, MC68EC030, MC68EC040, MC68LC040, MC68881, MC68882, MC68851 and Motorola are trademarks of Motorola, Inc. Amiga is a registered trademark of ESCOM AG. AmigaDOS, Kickstart, and Workbench are trademarks of ESCOM AG. SAS and Lattice are registered trademarks of SAS Institute, Inc. Aztec and Manx are trademarks of Manx Software Systems. ARexx is a trademark of The Wishful Thinking Development Corp. UNIX is a registered trademark of AT&T. OS-9 is a registered trademark of Microware Systems Corporation. All products mentioned in this manual are trademarks of their respective owners.

### Disclaimer

=====

The information, the ASX program, and all the associated utilities are provided "as is" without warranty of any kind, either expressed or implied. The entire risk as to the accuracy of the information herein is assumed by you. Daniel Weber does not warrant, guarantee, or make any representations regarding the use of, or the results of the use of, the information, the ASX program, or the associated utilities in terms of correctness, accuracy, reliability, currentness, or otherwise. In no event will Daniel Weber be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the information, the ASX program, or the associated utilities even if he has been advised of the possibility of such damages.

## 1.4 system requirements

---

## System Requirements

=====

ProOpts requires Kickstart and Workbench 2.0 or higher.

## 1.5 introduction

### ProOpts - The ProAsm Configuration Program

=====

ProOpts provides a simple method of generating or changing a configuration file for the ProAsm assembler. Since ProAsm supports configuration files to be loaded each time when it is called, the user is able to customize ProAsm to suit his particular wishes and needs. The options for a project can be set by clicking on the gadget that corresponds to the option. Even options for which the ProOpts utility does not have a gadget can be specified by a special string and listview gadget.

The generated configuration file is an ASCII file that contains the specified options as assembly directives. It is similar to an include file except that it is loaded at the very beginning of each assembly. There is no restriction on the use of assembler directives or even code in the configuration file. The user can also re-edit or add options using a texteditor. Previously generated config files can be loaded into ProOpts and then be changed to the new option settings.

## 1.6 running proopts

### Running ProOpts

-----

To run ProOpts, you can either double-click on its icon, or enter ProOpts on the shell command line. ProOpts requires Kickstart and Workbench 2.0 or higher to run. There is no need to use the AmigaDOS Run command to start ProOpts since it will automatically self-detach from the shell or CLI that it was started from.

## 1.7 using proopts

### Using ProOpts

-----

If started successfully, ProOpts opens the ProOpts Index window (shown above). This window contains seven gadgets that open additional windows. Each of these windows allows you to set options sorted after their main topic. The seven windows are:

Assembler Options  
Code Options  
Default Options  
Miscellaneous Options  
Optimization Options  
List Options  
Linker Options

In addition to these topic gadgets that represent these seven windows, three cycle gadgets can be found:

No Optimize

Optimize

Disables respectively enables the selected optimizations. By default, the optimizations are turned off.

No List

List

Disables or enables the listing generation. By default, the listing is disabled.

Executable

Linkable

Binary

Preasm

S-record

No Object

With this cycle gadget, the type of the output file can be selected. By default, the assembler creates an executable object file.

Three other action-gadgets can be found in the bottom row, that control ProOpts:

Save

Stores the current settings to ENVARC:proasm.config and ENV:proasm.config.

Save as

Save the current settings to a user specified file. After selecting this gadget, a file requester is opened that allows you to specify the name of the file the configuration is written to. The file requester will always appear with this command, whereas the Save command will assume the predefined names.

Cancel

Simply quits ProOpts.

ProOpts has also a menu, the same for all the different windows. The following menu items can be found:

Open...

The Open... (AMIGA-O) command allows you to load an existing

---



configuration file using a file requester. ProOpts updates all gadgets according to the new loaded configuration file. All lines that could not be recognized by ProOpts are added to the 'Custom Lines' in the assembler options window (see Assembler Options ).

#### Save

The Save (AMIGA-S) command stores the current settings to ENVARC:proasm.config and ENV:proasm.config.

#### Save as...

The Save as... (AMIGA-A) command is very similar to the Save command. It allows you to store the current settings to a user specified file. A file requester is opened that allows you to specify the name of the file the configuration is written to. The file requester will always appear with this command, whereas the Save command will assume the predefined names.

#### About

The About (AMIGA-?) selection displays the copyright and version notice for ProOpts.

#### Quit

The Quit (AMIGA-Q) command quits ProOpts.

#### Reset To Defaults

This command (AMIGA-D) resets all settings to their default.

#### Types of Gadgets used

-----

When using ProOpts, you may encounter one or more of the following basic types of gadgets:

##### action

Action-gadgets appear as raised buttons. Normally the buttons are labeled with a small text that is written inside the gadget. Selecting such a button causes an immediate action. For example, the Cancel gadget in the main window causes ProOpts to quit.

##### cycle

Cycle-gadgets look quite similar to action-gadgets, except that a small symbol of a curled arrow that points to itself is on the left side. By selection the gadget you can cycle through the available settings for that specific option. The cycle direction can be reversed by pressing the SHIFT key as you click the left mouse button.

##### list

##### listview

Generally list-gadgets appear as a set of control gadgets. A small scrollbar, arrow gadgets, and a string area. ProOpts has additional control gadgets attached to its list-gadgets. These gadgets are a string-gadget, an add-gadget (in one case), del-gadget, top-, up-, down-, and a bottom-gadget. Use the scroll

---

bar and the arrow gadgets to position the scrolling area to a specific entry. To add new entries to the list, select the add-gadget (if available) or type the new entry name into the string-gadget (below the scroll area) and press RETURN. Entries can be removed from the list, by selecting the entry to be cleared in the scroll area (to copy its name into the string-gadget), and selecting the del-gadget.

The top-, up-, down-, and a bottom-gadget can be used to move the entry to the desired position in the list. Top and bottom move the item directly to either the top end or the bottom end of the list. Up and down move the selected item one position upwards or downwards in the list. Some gadgets can be disabled if they cannot be used (e.g. on empty lists).

#### string

String-gadgets appear as raised ridge around a text area. Any string can be entered by clicking (activating) within the text area and typing your string. When you have finished typing, press the RETURN key to confirm or the TAB key to confirm and jump to the next string-gadget.

#### integer

Integer-gadgets are in function and look identical to the string-gadgets. But they accept only decimal integer numbers as input.

#### check

Check-gadgets appear as small rectangular raised buttons, labeled with a text on the left or the right side of the gadget. When the gadget is selected, a small checkmark appears, or disappears when unselected.

The rest of this section describes all the different windows that ProOpts offers to set and change the assembler options. Everything in these windows should be fairly self-explanatory. The individual options are not described to their last details, since they all can be found somewhere else in this manual. The discussed options carry references that leads you to their descriptions in this manual. The saved configuration file will be used the next time ProAsm is run.

## 1.8 pro.guide/Assembler Options

### Assembler Options

.....

### Include Directories

Include search pathes tell the assembler where to look for files to be included by the INCLUDE or INCBIN directives, or their synonyms. A new path can be added by either selecting the Add gadget that opens a file requester to select the directory, or its name can be typed into the string-gadget. You do not have to enclose the path name in double or single quotes if it contains

one or more whitespaces. ProOpts encloses automatically all entries with double quotes if needed.

When the settings are stored, these include directories are all written as arguments of the INCDIR directive to the configuration file. See INCDIR.

#### Custom Lines

When a configuration file is loaded at startup or via the Open... menu item, all lines that could not be recognized by ProOpts are put in this list. See ProOpts Directives Summary, for a list of all valid directives.

The custom lines list can also be used to add directives to the configuration file that are not supported by ProOpts. You may even put complete macro definitions into the list. Awkward, but it works.

#### No XPK Support

This check-gadget controls the XPK support. XPK support is disabled when the gadget is set. See OPT NOXPK.

#### Assembly Priority

Using the integer-gadget you have the possibility to set the task priority of the assembler while running (see ASMPRI).

Note that ProOpts does not check the range of the entered value to its validity. The range of the priority are the integers from -128 to 127. The CLI and the most other tasks have a priority of 0, and it is suggested that you do not set the priority higher than 10. A priority higher than zero means that the assembler will not be slowed down as much by other tasks. Setting the priority lower than 0 will prevent ProAsm from slowing down other programs with normal priority.

#### Symbol Search

With this cycle-gadget you can select an appropriate symbol search algorithm for your source codes. You can choose between Standard (default), ESS1, ESS2, and ESS3. See OPT ESS1, for more details.

## 1.9 pro.guide/Code Options

### Code Options

.....

#### CPU

This cycle-gadget specifies the processor to be programmed for. See Processor Type, for more information.

Note that the cycle-gadget entry Any represents the MCRELAX directive.

#### FPU

This cycle-gadget allows you to specify the floating-point

---

coprocessor you want to program for to allow its instruction set. See MC68881, for more details.

Note that the MC68040 and MC68060 processor have built-in FPU. Their FPU instruction sets are automatically allowed when the specific processor is selected (see the CPU gadget above).

#### MMU

This cycle-gadget only toggles between None, for no MMU, and MC68851. See MC68851, for more details.

Note that the MC68030, MC68040, and MC68060 MMU instruction set will automatically be enabled when the specific CPU has been selected (see the CPU gadget above).

#### Case

With this cycle-gadget you can toggle between case sensitive and unsensitive symbols. See OPT C, for more information.

#### Symbol Significance

Specifying a value using the integer-gadget makes the first part of symbol case sensitive or unsensitive, according to Case (see above). See OPT C, for more information.

Note that ProOpts does not perform a range check on the entered value. Valid values must be from 0 to 256, where 0 turns the symbol significance off.

#### Underscore for Local Labels

If this gadget is selected, the underscore (\_) rather than the dot (.) is used to introduce a local label (see OPT U).

#### Warn not Supervisor Instructions

This check-gadget allows you to enable or disable all warnings on supervisor-only opcodes. The warnings are disabled when the gadget is selected. See SUPER, for more details.

#### Warn not Read-Modify-Write Instr.

You can suppress all warnings for read/modify/write instructions by selecting this check-gadget. See READMODWRITE.

#### Ensure PC-Relative Code

Selecting this check-gadget forces the assembler to report a relocation not allowed error on any instructions that require relocation. See OPT P, for more information.

#### Immediate Data Check

All absolute addresses will be checked for missing pound signs (#), when this check-gadget is selected (see OPT I).

#### Automatic PC-Relative

In selected state this gadget forces the assembler to use the PC-relative addressing mode where possible instead of the absolute long addressing mode. See OPT AUTOPC, for more information.

#### Control software-supported instructions

These two cycle-gadgets allow you to control the use of

software-supported instructions.

The first cycle-gadget enables or disables the software-supported floating-point instructions of either the MC68040, the MC68060, or both processors. The second gadget controls only the use of the MC68060 software-supported integer instructions.

See OPT FPSP40, and OPT SP60, for more detailed information.

#### Bit Number

Using this cycle-gadget you can decide whether a warning, no warning, or an error should be reported if ProAsm encounters a bit number that is not in bounds. See OPT CHKBIT.

#### (Syntax) Model

This gadget can be used to determine whether the assembler should accept the new syntax model, the old one, or both. See NEWSYNTAX.

## 1.10 pro.guide/Default Options

### Default Options

.....

In the default options window you have the possibility to define the default behaviour of ProAsm. Refer to Assembly Options, for more details on these and possible settings. ProOpts uses the DEFAULT directive for all these cycle-gadgets when saved to the configuration file. See DEFAULT, for information about this directive and its arguments. The cycle-gadgets let you specify the mode arguments of the DEFAULT directive.

#### Default Branch Size

Corresponds to DEFAULT `_BRANCH,mode`.

#### Default Base Displacement

Corresponds to DEFAULT `_BASEDISP,mode`.

#### Default Outer Displacement

Corresponds to DEFAULT `_OUTERDISP,mode`.

#### Default PC Displacement

Corresponds to DEFAULT `_PCBASEDISP,mode`.

#### Default Address Displacement

Corresponds to DEFAULT `_ADRBASEDISP,mode`.

#### Linker References

Corresponds to DEFAULT `_LINKREF,mode`.

## 1.11 pro.guide/Miscellaneous Options

---

## Miscellaneous Options

.....

## Suppress All Warnings

This check-gadget lets you suppress all warning messages (see OPT NOWARN).

## Timing Information

Using this cycle-gadget you can select the way the assembler should report timing information and their quantity. See VERBOSE.

## Suppress All Errors

This check-gadget lets you suppress all error messages (see OPT E).

## Error Failat Level

The value in the integer-gadget specifies the maximum number of errors to be reported before aborting assembly. A value of zero sets the limit to unlimited number of errors. See FAILAT.

## Enable Escape Strings

This check-gadget controls the allowance of escape sequences withing strings. If it is selected, escape sequences are allowed. See ESCAPESTR.

## Enable Type Checking

Select this check-gadget to enable type checking. Refer to Type Checking, for more detailed information.

## Do Not Report Odd Addresses

If you select this check-gadget, all misalignment and odd access checks are disabled and no errors will be issued (see ODDOK).

## Report Odd PC Relative Displacements

By selecting this check-gadget additional misalignment and odd access checks are performed by the assembler. See ODD2ERROR.

## Report Multiple Includes

This check-gadget can be used to tell the assembler to report multiple includes, instead of ignoring them. See OPT Y.

## Show Filenames

Select this check-gadget to let the assembler report the name of all loaded include files (see OPT F).

## 1.12 pro.guide/Optimization Options

## Optimization Options

.....

The optimization options are spread over two windows. The descriptions of the individual options can be found by either the OPT O option (see OPT O), or the OPT Q (see OPT Q), unless otherwise noted. In the following tables you find only the options the gadgets

correspond to.

#### No Optimization Warnings

This gadget disables all warnings that are generated by optimizations. See OPT OW.

#### Multipass optimization

If this check-gadget is selected, multipass optimization is enabled (see MULTIPASS).

#### Use proasmoptim.library

This gadget enables the use of the optimizer library (see OPT OPTIMLIB).

#### Backward Branches

Corresponds to the OPT O1+ option.

#### Warn Forward Branches

Corresponds to the OPT O6+ option.

#### Zero Offsets

Corresponds to the OPT O2+ option.

#### Absolute Addressing

Corresponds to the OPT O3+ option.

#### MOVE.L to Quick

Corresponds to the OPT O4+ option.

#### ADD/SUB to Quick

Corresponds to the OPT O5+ option.

#### CMPI/CMPA.x #0 to TST

Corresponds to the OPT O9+ option.

#### Short Base Displacement

Corresponds to the OPT O13+ option.

#### Short Outer Displacement

Corresponds to the OPT O14+ option.

#### Short Immediate Data

Corresponds to the OPT O8+ option.

#### CLR.L Dn to MOVEQ #0,Dn

Corresponds to the OPT Q1+ option.

#### ASL/ROXL.x #1,Dn to ADD/ADDX.x

Corresponds to the OPT Q2+ option.

#### Next Page

By pressing this gadget, you change to the next optimization page.

#### PC-Relative Addressing

Corresponds to the OPT O7+ option.

#### MOVE.B #-1,<EA> to ST.B <EA>

Corresponds to the OPT Q11+ option.

Skip ADDA/SUBA #0,An  
Corresponds to the OPT Q4+ option.

MOVEQ #0,An to SUBA.L An,An  
Corresponds to the OPT Q3+ option.

MOVE.L #x,Dn to MOVEQ #y,Dn & NOT.x Dn  
Corresponds to the OPT Q5+ option.

MOVE.L #x,Dn to MOVEQ #y,Dn & SUBQ.L #y,Dn  
Corresponds to the OPT Q7+ option.

MOVE.L #x,Dn to MOVEQ #y,Dn & SWAP Dn  
Corresponds to the OPT Q8+ option.

MOVE.L #x,Dn to MOVEQ #y,Dn & ADD.B Dn,Dn  
Corresponds to the OPT Q9+ option.

EOR(I).x #-1,<EA> to NOT.x <EA>  
Corresponds to the OPT Q12+ option.

## 1.13 pro.guide/List Options

List Options  
.....

List Macro Expansion  
If this gadget is selected, the complete macro text for all macro expansions is included in the listing. See OPT M.

List Macro Definitions  
Macro definitions are included in the listing when this gadget is selected. See OPT MD.

List Symbol Table  
By selecting this gadget you tell the assembler to add a symbol listing to the listing file. See Symbol listing.

List Include Files  
The contents of included files is included in the listing when this gadget is selected (see FORMAT).

List Conditional Directives  
Select this gadget to include conditional assembly directives in the listing file (see OPT CL).

No Listing Header  
This gadget can be used to suppress all titles and subtitles. See FORMAT, for more information.

No Line Numbers  
If this gadget is selected, no line numbers are written to the

---



listing. See FORMAT.

#### No Section# and Offset

The section numbering and program counter offset are suppressed when this gadget is selected (see FORMAT).

#### Bytes Opcode Dump

This cycle-gadget allows you to specify the width of the hexdump field in bytes. The range of the width is from 2-22 bytes, or No to suppress the hexdump field completely. See FORMAT.

#### PLEN

The PLEN integer-gadget can be used to set the page length for the assembly listing. The page length may be a value between 12 and 255. See PLEN.

#### LLEN

By entering a value into the integer-gadget, you can set the line length for the listing. The line length should be from 38 to 255. A value of zero instructs ProOpts not to write the line length to the configuration file. See LLEN, for more details.

#### Listing File

Using the string-gadget you can specify the filename of the listing file (see Auxiliary Output Files). The name can be set by either typing it into the string-gadget, or by using the file requester that opened when the Select gadget was pressed. See LISTFILE.

#### Listing Title

You can specify a page title by typing it into this string-gadget (see TTL).

## 1.14 pro.guide/Linker Options

### Linker Options

.....

#### Small Object

Select this gadget when you want to have all hunks merged into one single CODE hunk. See SMALLOBJ, for more details.

#### Auto XRef

When this gadget is selected, the assembler treats all undefined symbols as external symbols (see AUTOXREF).

#### SmallCode

When this check-gadget is selected, all code hunks are merged into one single hunk (see SMALLCODE).

#### SmallData

When this check-gadget is selected, all data hunks are merged into one single hunk (see SMALLDATA).

---

### SmallBSS

When this check-gadget is selected, all bss hunks are merged into one single hunk (see SMALLBSS).

### Debug Symbols

With this cycle-gadget you can particularize the way symbol information is written to the object file. You can choose between:

None

for no symbol information,

All

when the symbol information for all defined symbols should be written to object file (see ADDSYM), or

Exports Only

that restricts the symbol information generation to the exported (XDEF) symbols only (see OPT X+).

### Line Debug

This gadget allows you to set the format of the debugging information. See DEBUG, for more detailed information, especially for the LINE and HCLN keywords.

### Debugging Information

You can specify the type of debugging information to be produced. See DEBUG.

### All Includes

When this check-gadget is set, debugging information for all included files is generated (see DEBUG, for the ALL keyword).

### Origin

This integer-gadget can be used to set the load address for the S-record file (see SREC).

### S-Record Type

This cycle-gadget allows you to set the Motorola Record format type for the S-record file. See SREC.

### Length

You can set the record length of the S-record file by typing the value into the integer-gadget. The value must be from 16 to 255. See SREC, for more information.

## 1.15 author

Author

=====

If you have bugreports, questions, ideas, flames or complaints (constructive criticism is always welcome), or if you just want to contact me, write or send a letter to:

---

Daniel Weber

Internet:       dweber@amiga.icu.net.ch     (preferred)  
                  dweber@iic.ethz.ch

Mail:            Daniel Weber  
                  Hoeflistrasse 32  
                  CH-8135 Langnau  
                  Switzerland.

---