# MUI-ASL

The MUIfied ASL replacement
Version 42.2

This manual has been typeset by TeX and `Texinfo`.

# 1 Introduction to MUI-ASL

Now that MUI applications are so many and widespread, and that even *Amiga Technologies* sold some MUI products, several people complain that the standard ASL library clashes with MUI's ultra-configurable look – and don't ask for keyboard control.

In the past years there have been many attempts at providing a standardized requester library, or at least a good file requester. The most successful were **ARP**[1] and '`req.library`' (which also provided some other kinds of useful requesters), but they didn't survive the advent of Release 2.0 and ASL. Today the most widespread alternative is Nico François' **ReqTools**, which you may regard as better or worse than ASL depending on your personal taste, but in the end it's not that different: you get keyboard control and resizable windows, but the GUI look is still plain **GadTools**.

An interesting full-featured replacement for all your file requesters is Stefan Stuntz's[2] first well-known project, **MagicFileRequester** (**MFR**). But as the name implies it is only a *file requester* – no font, no screenmode – and the look is neither configurable nor, in my opinion, very nice. It also suffers from some bugs that make it pretty unstable.

These replacements, however, have a problem: they are not *the* standard. Only ASL is a standard, because it is part of the OS and will grow with it (and so will MUI-ASL, I hope).

After an intresting travel through object-oriented programming and BOOPSI, I am proud to present **MUI-ASL**, the MUIfied ASL library. The ambitious aim of MUI-ASL is to put a good looking, highly configurable user interface into ASL, giving many benefits to innumerable programs.

---

[1] The AmigaDOS Replacement Project replaced the entire '`dos.library`', most DOS commands and the `CLI` with much more powerful functions, and was the first widespread file requester. In fact, '`dos.library`' V36+ and the ASL file requester derive from ARP.

[2] The author of MUI.

# 2  Requirements

You need the following items to run this software:

- An Amiga with a 68020 CPU (or better).

  Believe me, you wouldn't like to run this thing on a 68000. In fact, you shouldn't run anything on a 68000 these days, except maybe old games.

- AmigaOS Release 3.0 or later.

  If you haven't upgraded yet, then it's definitely time to do it: you are missing a lot of new features and good software and, what's worse, you are holding Amiga technology behind.

  Don't bother me because you want it to run on 2.x. I won't even respond.

- MagicUserInterface by Stefan Stuntz, version 3.3 or later.

  MUI is not distributed with this software: you can get it through various public domain, freeware or shareware collections, through electronic networks, or directly from the author. See Section 2.1 [MUI], page 2, for copyright and registration information.

  We will assume that MUI is already installed and working on your system.

## 2.1  MUI

This package uses

### MUI – MagicUserInterface

© Copyright 1993-96 by Stefan Stuntz

`MUI` is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

`MUI` is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called '`muiXXusr.lha`' (`XX` means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send DM 30.- or US$ 20.- to:

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

Support and online registration is available at

`http://www.sasg.com/`

# 3 Disclaimer

THERE IS NO WARRANTY, EITHER EXPRESSED OR IMPLIED, AS TO THE QUAL-
ITY OR PERFORMANCE OF THIS PROGRAM. THE ENTIRE RISK FOR THE USE OF
THIS PROGRAM IS ASSUMED BY THE USER (YOU). THE AUTHOR MAY NOT BE
HELD RESPONSIBLE FOR ANY LOSS OR CORRUPTION OF DATA, EQUIPMENT OR
OTHER GOODS RESULTING BY THE USE OR POSSESSION OF THIS PROGRAM, EVEN
IF IN KNOWLEDGE THAT SUCH LOSS OR CORRUPTION MIGHT HAPPEN.

SHOULD APPLICABLE LAWS LIMIT THE EXTENT OF THIS DISCLAIMER, OR IM-
POSE ANY FORM OF RESPONSIBILITY, THEN YOU ARE NOT ALLOWED TO USE
THIS SOFTWARE.

I did my best to write a reliable piece of software. I tested it extensively. Enforcer, MungWall
and PoolWatch didn't complain. It works on my machine. If it doesn't work on yours, it's your
problem, or maybe even your fault. However, I would like to know. See Chapter 11 [Bug
Reports], page 11.

# 4 License

MUI-ASL is offered to you under the concepts of *shareware*. You can use it for an evaluation period of up to 4 weeks without paying any charge, but if you are going to use MUI-ASL after the evaluation period, you are morally obliged to register, or your conscience will haunt you forever! See Chapter 5 [Registering], page 5, for more details.

In the following paragraphs, MUI-ASL and all the related distribution files will also be indicated as *the software* or *the files*. The licensee will also be indicated as *you*.

By copying, distributing and/or using the software you indicate your acceptance of this license to do so, and all its terms and conditions. If you don't agree on the terms of this license, you are required to delete all the copies of the software in your possession.

You may:

- Install and use the software on any computer you want to.
- Make verbatim copies of the original distribution archive, without any files being omitted, added or changed. Lossless compression of the files is permitted. Adding bulletin board banners (see below) is discouraged, but tolerated.
- Distribute copies of the software for no charge, except for a reasonable fee for the media and your copying work. Allowed distribution media include electronic media, such as bulletin boards or the Internet.
- Include the software, in compressed and/or uncompressed form, in public domain, freeware or shareware collections such as Fred Fish's Amiga Library Disks or CD-ROMs, provided that the price is reasonable. The author reserves the right to decide how much "reasonable" means.

  Fred Fish and the Aminet guys are expressly authorized and encouraged to include the software in their collections.

You may **not**:

- Copy and/or distribute the software without the accompanying documentation and any other additional files that come with the original distribution archive.
- Alter the software and/or the accompanying files in any way.
- Copy and/or distribute modified versions of the software and/or the accompanying files.
- Grant sublicenses, leases or other rights in the software, except as explicitly authorized in this license.
- Forbid other parties to use or distribute the software.

You agree to cease distribution of the software if the author requests so.

The program is released "as is", without any warranty, either expressed or implied, and in no event will the author be liable for any damages caused by the use of this program. See Chapter 3 [Disclaimer], page 3.

# 5 Registering

If you think that MUI-ASL is useful and you wish to continue using it, you should register by sending a small amount of money to the author. By "small amount" I mean enough to buy a pizza and a coke, or a box of chocolates[1].

The registration will buy you nothing but my gratitude: I feel that the project is not so big that I have to "encourage" registration by means of limitations in functionality (or enhancements for registered users, as you may see it). That's why I said "a small amount of money". Feel free to send just about anything: it will encourage me to continue the development of MUI-ASL, and possibly of some other software.

To register, send your gift to the author (see Chapter 14 [The Author], page 14), along with your physical and email addresses. If you choose to send money, I strongly suggest to use an international Postal Money Order because the currency exchange is handled automatically and you don't take the risk of sending cash. If possible, I will acknowledge your registration by email.

**REMEMBER:** If you choose to send cash, use a thick envelope so that it cannot be seen through, and send it by registered letter!

---

[1] You can even send chocolate directly: I especially like bitter chocolate with about 60% of cocoa an I love white Lindt Lindor. If you live in Switzerland or northern Europe, this is definitely the preferred gift. Don't send chocolate during summer, though.

# 6 The File Requester

The file requester features a directory list and a volume list, in a tribute to the good old 'req.library' requester and to **MFR**. There is a balance bar between the two lists that allows you to resize them as you wish.

**Please note:** Currently MUI does not save the position of the balance bar. A future release of MUI-ASL will supply a preferences program that will allow you to configure the relative weights of the two lists.

In the following sections you will find more informations on some aspects of the file requester.

## Features of the file requester:

- If you enter a directory name in the 'Drawer:' field and then click 'Ok' without hitting RETURN in the 'Drawer:' field itself, the directory name will be returned unchanged, even if it is a nonexistent directory.

## 6.1 File Requester Menus

The file requester menu allows you to do some operations on the files you are browsing – kind of a minimal directory utility.

'Control/All'
'Control/None'
        Select all or none of the files in the file requester.
'Actions/New drawer...'
        Create a new drawer. You will be asked for the drawer name.
'Actions/Rename...'
        Rename the active file. You will be asked for the new name.
'Actions/Delete...'
        Delete the active file. You will be asked for confirmation.

Moreover, the directory listing in the file requester has its own context menu for changing the sort ordering of the list. The menu is as follows:

```
Settings
   Sort drawers  ≫
      First
      Last
      Mixed
   Sort order    ≫
      Name
      Date
      Size
   -------------
   * Reject Icons
```

## 6.2 File Requester Caveats

- Using 'Sort drawers/Mixed' with 'Sort order/Size' will yield strange results in the directory ordering: directories don't have a size, so they cannot be mixed correctly with files.

# 7 The Font Requester

*This section of the manual will be enchanced in a future release.*

The font requester remembers the list of available fonts, so you will not have to wait for the list being read every time a requester is opened.

## 7.1 The font preview area

The area at the bottom of the font requester will show you a preview of the currently selected font. The preview will show designed[1] fonts only, in order to prevent any delay when you change font[2]. As a side effect, if you select a non-existent size the preview will show the closest size for that font, or nothing if that font doesn't have any designed sizes.

There is a balance bar just above the preview area: it allows you to resize the preview area at the expense of the font list.

## 7.2 Tech notes

*This section is intended mainly for developers.*

### 7.2.1 MaxFrontPen and MaxBackPen

During the development of MUI-ASL I discovered a small debatable point in the ASL documentation, related to the interpretation of font requester's MaxFrontPen and MaxBackPen attributes. In the ASL autodoc they are explained as "the maximum number of pens that are offered to the user"; indeed, the font requester palette gadgets do have `maxpen` buttons.

According to my logic, though, "max pen" should mean "the maximum pen to let the user choose", i.e. the palette gadgets should have `maxpen+1` buttons. This is even more logic if you consider that the default maximum value is 255: the definition in the ASL autodoc implies that by default you can't choose the 256th color of a 256-color screen, and in fact you can't specify 256 pens at all, because these attributes are stored as UBYTEs, so they can range from 0 to 255 only.

MUI-ASL behaves like ASL, but only after I saw that my logic was wrong.

---

[1] Technically speaking, the `FPF_DESIGNED` flag must be set.
[2] When you change font, the font size field stays the same: if the new font didn't have that particular size ready, the font engine would calculate it on the fly by rescaling an existing size or, in case of an outline font, by rendering the given size. It takes time.

# 8  The Screen Mode Requester

*This section of the manual will be enchanced in a future release.*

There is not much to say about the screen mode requester that you don't know already. You get a list of available display modes and, if the requesting program asked for them, fields for setting overscan, width, height, depth and autoscroll mode.

The author could not yet test the requester on graphic cards or RTG systems such as CyberGraphX[1]. If you have such a system, please report any problems.

---

[1]  The first beta release indeed had problems with CyberGraphX – see Chapter 13 [History], page 13.

# 9 Compatibility with the Real ASL

This chapter lists some ASL features that MUI-ASL does not currently have, with a short explanation; most of these misfeatures are a consequence of MUI, and implementing them is either impossible or against MUI philosophy[1].

See also Chapter 10 [Known Bugs], page 10, and Chapter 11 [Bug Reports], page 11.

- All requesters ignore PrivateIDCMP and IntuiMsgFunc because MUI transcends IDCMP messages completely. Basically, MUI-ASL always behaves as if you asked for a private IDCMP port and never calls IntuiMsgFunc.

  This feature is most useful for refreshing simple-refresh application windows when the user waves requesters over them; MUI applications, in particular, should have a minimal IntuiMsgFunc[2] for this purpose.

  I'm planning to solve this in a future release.

- As of this writing, Locale is not implemented; I'll work on it as soon as the rest works. For now I'd better keep overall complexity low, or debugging could become just too hard.

- All requesters ignore LeftEdge, TopEdge, Width, Height and TextAttr because tampering with an application's appearance at this level is against MUI philosophy.

  However, when the requester is closed the values of LeftEdge, TopEdge, Width and Height will be set correctly, in case someone is using them.

- All requesters ignore FilterFunc and HookFunc: the callback protocol for these functions provides for a pointer to the requester structure, which is not available at the point these functions should be called due to the deep class structure of the underlying MUI application.

  This problem could probably be solved with a lot of work and/or some bad tricks to pass the requester pointer that far, but I won't even think about this problem unless I see a real application that is using the filter hooks. So live with it for now.

- The screen mode requester does not have the information window yet.

- Reserved fields in the requester structures are used for my very own purposes, so evilish programs that peek them will most probably crash. Well, that's *their* problem.

---

[1] Anyway, if you have a program that needs one of these features, let me know. Philosophy is useless if things don't work.

[2] Have a look at Application.mui/MUIM_Application_CheckRefresh.

# 10  Known Bugs

"Bugs? What bugs? My code doesn't have bugs!"

Alas, no software I know of is free of bugs. However, it is better to know where bugs are, so here it is:

- Activating the volume list of the file requester by pressing the `TAB` key will sometimes cause an Enforcer[1] hit. As far as I can tell, the problem is in the MUI Volumelist class, so I cannot correct it. However, it seems that it is harmless.

- There are also some unimplemented features which shouldn't affect end user functionality. See Chapter 9 [Compatibility], page 9.

**Please note:** There is no doubt that some programs will behave badly when confronted with MUI-ASL. Nobody's perfect, or at least not me, so there will be problems somewhere; but then consider that first of all I wrote MUI-ASL for myself, and I started using it well before it was complete, so I had chances to eradicate any problems I encountered with the programs I use daily.

So, if something doesn't work, don't say "MUI-ASL sucks": tell me which program behaves badly and wait for me to check it out and see who is wrong. If the problem is in my code, be sure that I will fix it yesterday.

---

[1] **The Enforcer** is a programmer's tool used to detect illegal memory accesses. Causing so called *Enforcer hits* is usually a clear sign of a programming mistake.

# 11  Bug Reports

If you discover a problem in MUI-ASL, first make sure that I didn't know already: in other words, read carefully Chapter 9 [Compatibility], page 9, and Chapter 10 [Known Bugs], page 10. Then, please write to the author (see Chapter 14 [The Author], page 14) and provide the following information:

- The version of MUI-ASL which shows the problem.
- The exact version of the operating system (see Workbench's About requester), MUI and any other relevant software.
- Your full Amiga configuration (model, CPU, peripherals, etc.).
- A detailed explanation on how to reproduce the problem. If I can't reproduce the problem, I'll have lots of troubles locating it, and I will probably end up thinking it is your fault. Bug reports like "it doesn't work sometimes" are definitely not what I need.
- If MUI-ASL irritates tools like The Enforcer or MungWall, please include their output in your bug report.

If you are missing something in MUI-ASL, please write to the author too. Check out Chapter 12 [Future], page 12, and tell me what you would like to see first.

# 12  Planned Enhancements for the Future

- A preferences program. That's one of the biggest ASL misfeatures.
- Buffering for the file requester. ReqTools has it.
- How about a color requester? ReqTools has it, too. However, standard ASL doesn't, so it could cause compatibility problems with future ASL releases – unless Amiga Technologies adopts MUI-ASL.
- The compatibility issues listed before (see Chapter 9 [Compatibility], page 9) are of course good candidates for future development; tell me if you need one.

# 13 History

42.1ß (31.08.96) – first beta release

This release had severe problems with CyberGraphX and any other program that might open 'asl.library' early in the startup-sequence, when the MUI assignments usually have not been set up yet.

Otherwise, it was just a beta: most things worked, most other were simply missing.

42.2ß (12.09.96) – second beta release

- You can select all or none of the files in the file requester via two menu items (applicable only when the file requester is in multiselect mode).
- MUI-ASL opens MUI libraries only when needed (i.e. when AslRequest() is called). This should solve problems with CyberGraphX and other programs.
  
  (reported by lots of people)
- Enhanced the layout algorythm of the font requester's palette gadgets. Previously the layout was a long row of color boxes, which was too big to be displayed on deep screens. Note that there still could be problems on small screens with many colors.
  
  (reported by helix2@ios.com)
- MUI-ASL did not check if the requester window did open, so it could enter the input loop without a window! This often happened as a consequence of the aforementioned layout problem. Now an alert is shown and the requester is aborted.
- Corrected an annoying bug: the requester structures were filled with current values even when the user hit 'Cancel'.
- Implemented the 'New drawer...' function in the file requester. In fact it was there in the previous release too, but I forgot to enable the menu item after correcting a nasty stack problem.
  
  (requested by someone whose mail I didn't keep – sorry dude)
- Fixed Enforcer hits when MonitorSpec not available for a display mode.
  
  (reported by Stefan Becker)
- The 'Pattern' gadget is now above 'Drawer' and 'File' as in the real ASL. Previously the 'Pattern' gadget was below them.
  
  (requested by Marcin Orlowski)
- The file requester window allows drag'n'drop of icons on it.
  
  (requested by several people)
- The file requester lists (drawers/files and volumes) use the fixed-width font now. Just to see if you like it. ;-)
  
  The font will be configurable (at least among MUIV_Font_List and MUIV_Font_Fixed) as soon as I write a configuration program.
- The file/font requesters correctly select the initial file/font. However, they don't jump to it. I'm working on this.
  
  (reported by several people)

# 14  The Author

This software was written by Flavio Stanchina. You can reach me at these addresses, via standard or electronic mail.

Flavio Stanchina
Loc. Montevaccino 39
38040 Trento
ITALY

FidoNet: 2:333/801.9
Internet: flavio@iestn.inet.it

If you want a reply, please use e-mail.

# 15 Thanks

I would like to thank the following people:

Helix <helix2@ios.com>
Martin Steigerwald <steigerw@stud.uni-frankfurt.de>
Mats Nyman <matsn@tripnet.se>

They managed to notice MUI-ASL on Aminet, download it, test it and write bug reports in less than 20 hours after I uploaded the first beta release. That's what I call fast feedback!

Many other people contributed with bug reports and suggestions just a few days after the first release, but they are too many to list here. Thank you!

# Concept Index

# Table of Contents