

ixprefs

Kriton Kyrimis

Copyright © CopyrightÂ©1995,1996 Kriton Kyrimis

COLLABORATORS

	<i>TITLE :</i> ixprefs		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Kriton Kyrimis	January 15, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ixprefs	1
1.1	ixprefs user's guide	1
1.2	Introduction	1
1.3	Installation	2
1.4	Recompiling	2
1.5	The ixprefs window	2
1.6	Allow AmigaOS-style wildcards	3
1.7	Case sensitive pattern matching	3
1.8	Translate /	4
1.9	Do not flush library	4
1.10	Ignore global environment	4
1.11	Generate Enforcer hit when a trap occurs	4
1.12	Suppress the `Insert volume in drive' requester	4
1.13	Membuf size	4
1.14	Networking support	5
1.15	Profiling method	5
1.16	Physical blocks to build one logical block (for stdio)	6
1.17	Save	6
1.18	Use	6
1.19	Cancel	6
1.20	The ixprefs menu	6
1.21	CLI usage	7
1.22	About	8
1.23	Quit	9
1.24	Reset to defaults	9
1.25	Last saved	9
1.26	Restore	9
1.27	Program history	9
1.28	Author	12
1.29	Distribution	12
1.30	Requirements	12
1.31	Acknowledgments	12

Chapter 1

ixprefs

1.1 ixprefs user's guide

```
IXPREFS
A configuration program for ixemul.library v.45+
Version 2.4
```

```
Introduction
Distribution
Requirements
Installation
Recompiling
The ixprefs window
The ixprefs menu
CLI usage
Program history
Author
Acknowledgments
```

1.2 Introduction

Back when version of 41.0 of ixemul.library was released, it came with numerous new configuration options, one of which I wanted to use. By playing with ixconfig, the configuration program that accompanied that version of the library, I saw that this program had two deficiencies:

- a. It had almost as many command line options as UNIX ls, which are hard to

remember.

- b. Making the new configuration persistent was done using a horrible kludge, namely the `-s` option of `ixconfig`, which instructed `ixconfig` never to return, to make `ixemul.library` unflushable. Even if one had memory to burn, permanently allocating 120+ Kbytes, just to remember the handful of bits that define the configuration of `ixemul.library`, was a terrible waste.

Having a bit of spare time, I decided to try addressing these two points: By writing a graphical configuration program, I believe I addressed the first, and by not supporting the equivalent of the `-s` option, I hoped to give the maintainers of `ixemul.library` the incentive to read, at library startup, the configuration from a file similar to the one that `ixprefs` maintains, so that `ixemul.library` would not have to stay permanently in memory if it is to remember its configuration.

Starting with `ixemul.library` version 42.0 this functionality has been implemented.

1.3 Installation

Simply copy `ixprefs` (and, optionally, `ixprefs.info`) somewhere on your disk.

If you are upgrading from version 1.x, delete the old configuration file `S:ixprefs`.

If you are upgrading from version 2.0 beta 1, rename `ENVARC:ixprefs` to `ENVARC:ixemul.prefs` and `ENV:ixprefs` to `ENV:ixemul.prefs`.

1.4 Recompiling

`Ixprefs` is now integrated into the `ixemul.library` source distribution.

The makefile patches the C code generated by GadToolsBox (`ixprefs.c`) using a sed script, to add support for AmigaOS 3.x features. This script should work for other programs generated by GadToolsBox, so you might want to use it in other projects as well.

When creating C source code, you will create files `ixprefs.c` and `ixprefs_temp.c`. The latter contains templates for all functions called to handle gadgets and menus. If you have not added any new gadgets or menus, you can discard this file, otherwise cut the templates for the new functions and paste them to `functions.c`.

1.5 The `ixprefs` window

Allow Amiga wildcards

Case sensitive pattern matching

Translate /

Do not flush library

Ignore global environment (ENV:)

Suppress the "Insert volume in drive" requester

Physical blocks build one logical block (for stdio)

Membuf size

Create Enforcer hit on trap

<type>
Network support

<type>
Profiling method

Save

Use

Cancel

1.6 Allow AmigaOS-style wildcards

The ixemul.library startup code expands wildcards in the command line. If this option is set, AmigaOS-style wildcards are recognized, in addition to UNIX-style wildcards which are always recognized. Thus, if this option is set, you can use either *.c or #?.c to specify all C source files, while if this option is not set, you can only use *.c.

1.7 Case sensitive pattern matching

If this option is set, pattern matching in UNIX-style wildcard expansion is case sensitive, i.e., *.c expands to all files ending in .c (lower case), while *.C expands to all files ending in .C (upper case). If this option is not set, UNIX-style wildcard expansion is case insensitive (e.g., *.c expands to files ending in both .c and .C).

1.8 Translate /

If this option is set, `ixemul.library` interprets `"/foo"` as `"foo:"`, and `"foo//bar"` as `"foo/bar"`.

1.9 Do not flush library

If this option is set, then `ixemul.library` will always remain resident in memory, and will not be flushed when no programs are using it. This can be used to ensure that the library settings are not lost by flushing and then reloading the library, at the cost of making the memory occupied by the library unavailable to other programs. It is also slightly faster, because the library is loaded only once.

When this option is not set, then it will be possible to flush `ixemul.library` when no programs use it (e.g., by typing `"avail flush"` at the CLI prompt or by selecting the `Debug->flushlibs` item from the workbench menu).

1.10 Ignore global environment

If this option is set, the `getenv()` function returns local environment variables, i.e., those created by the AmigaOS `"set"` command. If this option is not set, the `getenv()` function returns global environment variables, i.e., those created by the AmigaOS `"setenv"` command, in addition to local ones. Local environment variables have precedence over global ones.

1.11 Generate Enforcer hit when a trap occurs

If this option is set then, whenever a trap (division by zero, bus error, etc.) occurs, `ixemul.library` writes the error code to the address `0xdeaddead`. Using the stack dump from Enforcer you can then try to find out where the problem occurred. If this option is not set, no Enforcer hit is generated.

1.12 Suppress the 'Insert volume in drive' requester

If this option is not set, whenever `ixemul.library` tries to access a non-existent volume, it will pop up a requester asking for the volume to be inserted. If this option is set, `ixemul.library` will not pop up this requester.

1.13 Membuf size

Files up to this size in bytes are cached in memory.

1.14 Networking support

This option allows the user to specify the kind of networking support they have:

0. Auto detect (default): ixemul tries to figure out what networking support the user has (none, AmiTCP, AS225).
1. No networking: all networking is turned off, regardless of whether the user has AmiTCP or AS225 or not.
2. AS225: set to AS225, if available. Otherwise no network support.
3. AmiTCP: set to AmiTCP, if available. Otherwise no network support.

Note that 'Auto detect' will check for AmiTCP first, and AS225 second. This is to circumvent problems when the socket.library emulation for AmiTCP is used.

1.15 Profiling method

Gcc provides the `-pg` flag to enable code profiling. When a program is compiled with `-pg`, a special function is called at the beginning of each function, which registers 1) how many times this function is called, 2) which functions called this function.

In addition to this, a function in `ixemul.library` is called 50 times per second. This function, among other things, increases timers and also, if the current program is compiled with `-pg`, samples the current program counter (PC) and registers this.

When a program that is compiled with the `-pg` option has finished, a file called `gmon.out` is produced, containing the above two sets of statistics. Using the `gprof` program, you can read this file and determine which programs were called and how much time was spent in each function. The timing information is derived from the PC sampling.

The problem with this scheme is that if the PC is outside the code in your program, nothing is registered. Suppose a function calls a ROM function or some library function that takes 10 seconds to finish. According to `gprof`, this function took 0 seconds, because the time spent elsewhere is not counted. This may be what you want (no 'pollution' from other sources), but there are cases where this is not what is needed.

Thus, `ixemul.library` provides three methods of sampling the PC:

1. Only in program (default): take samples only while your program is running and the PC is within your program.
2. Only in task: take samples while your program is running and if the PC is outside your program code (e.g. in `ixemul` or in `intuition.library`), use the last function your program entered before calling `ixemul` or whatever.
3. Profile always: always take samples, even if other programs are running. Again, the last recorded function gets the hit in those cases.

The choice of option depends on what you are interested in profiling: the

performance of just your own code (in UNIX terms, the ``user time''), the overall performance of your program (in UNIX terms, ``user'' plus ``system'' time), or the performance of your program plus any external programs that your program invokes (e.g., using the `system()` function).

1.16 Physical blocks to build one logical block (for stdio)

This number denotes the number of physical (usually 512-byte) blocks that `ixemul` treats as one block during I/O, for efficiency.

1.17 Save

Applies the user-specified configuration to `ixemul.library`, saves it in `ENV:ixemul.prefs` and `ENVARC:ixemul.prefs`, and exits.

1.18 Use

Applies the user-specified configuration to `ixemul.library`, saves it in `ENV:ixemul.prefs` and exits.

1.19 Cancel

Exits the program without making any changes to the configuration.

1.20 The ixprefs menu

```
Project Edit
Save
Reset to defaults
Use
Last saved
About
Restore
Quit
```

1.21 CLI usage

```
Type "ixprefs" without any arguments, to get the
ixprefs window
.
```

Alternatively, you can control the program from the CLI. You can use either short options, e.g., "-m 1024", or long options, e.g., "--membuf-limit 1024". If you use long options, you need only specify enough of the option name to disambiguate it from other option names, e.g., "--mem 1024".

The following CLI options are available:

```
-a 1, --allow-amiga-wildcard

    accept AmigaOS wildcard notation
-a 0, --no-allow-amiga-wildcard

    do not accept AmigaOS wildcard notation
-s 1, --unix-pattern-matching-case-sensitive

    use case sensitive UNIX pattern matching
-s 0, --no-unix-pattern-matching-case-sensitive
do not

    use case sensitive UNIX patter matching
-/ 1, --translate-slash

    translate /foo -> foo: and a//b -> a/b
-/ 0, --no-translate-slash
do not

    translate /foo -> foo: and a//b -> a/b
-m N, --membuf-limit N

    files up to N bytes are cached in memory
-b N, --fs-buf-factor N

    N physical blocks map into 1 logical (stdio) block
-i 1, --ignore-global-env

    ignore global environment
    (ENV:)
-i 0, --no-ignore-global-env

    do not ignore global environment
    (ENV:)
-e 1, --enforcer-hit

    generate Enforcer hit
    when a trap occurs
-e 0, --no-enforcer-hit

    do not generate Enforcer hit
    when a trap occurs
-v 0, --insert-disk-requester

    suppress the 'Insert volume in drive' requester
```

```

-v 1, --noinsert-disk-requester

do not suppress the 'Insert volume in drive' requester
-f 0, --flush-library

prevent ixemul.library from being flushed from memory
-f 1, --no-flush-library

do not prevent ixemul.library from being flushed from memory
-n 0, --auto-detect

set
networking support
to auto detect
-n 1, --no-networking
turn off
networking support
-n 2, --as225
use AS225
networking support
-n 3, --amitcp
use AmiTCP
networking support
-p 0, --profile program

profile
the program only
-p 1, --profile-task

profile
while the task is running
-p 2, --profile-always
always
profile
your program
-d, --default
reset settings to defaults (other options are ignored)
-L, --last-saved
reset settings from ENV:ixemul.prefs
(other options, including --default, are ignored)
-S, --save
save new configuration
-R, --report
display configuration
-V, --version
displays information about the
program
and its
author
(other options, including --default and --last-saved are ignored ↔
)
-h, --help
display the various ixprefs options

```

1.22 About

```
Displays information about the
program
and its
author
.
```

1.23 Quit

Exits the program without making any changes to the configuration.

1.24 Reset to defaults

Resets the various options to their default values.

1.25 Last saved

Reads the configuration stored in ENVARC:ixemul.prefs.

1.26 Restore

Sets the various options to the values that are currently being used by ixemul.library.

1.27 Program history

Version 1.0 June 23, 1995
First public release.

Version 1.1 June 26, 1995

- * Made the following enhancements, suggested by Kamil Iskra (iskra@student.uci.agh.edu.pl):
 - + Ixprefs window is now font adaptive.
 - + Ixprefs window is now activated when opened.
 - + GZZ adjust flag in GadToolsBox turned off.
 - + CheckBox gadgets are now scalable under AmigaOS 3.x.
 - + NewLook menus under AmigaOS 3.x
 - + Use OpenFont() rather than DiskFont() to access system font, as this font is always in memory.
 - + Slightly rearranged window layout to avoid problems with sizable gadgets.
- * Updated the documentation.

Version 1.2 July 3, 1995

- * Added a UNIX-style CLI interface using getopt_long().
 - * Modified showrequester() to print messages on stderr if the ixprefs window
-

- is not open, so that no requesters pop up when the CLI interface is used.
- * Updated ixemul.library version check, so that the program now runs under both version 41.0 and 41.1 without producing a warning.
 - * Fixed the dates in this section. Not that anyone cares, but the previous versions of this program were written in June, not May!
 - * Updated the documentation.

Version 1.3 (not released)

- * Made a very minor modification in gadget placement (some gadgets were off by one pixel).
- * Fixed a bug where if AmigaOS notation is turned off, it was impossible to load or save the configuration file. Remaining bug: if neither AmigaOS notation nor / translation is allowed, there is no way to access S:ixprefs, and loading/saving is impossible. This could be fixed by temporarily twiddling the flags. However, as the "allow AmigaOS notation" flag has been made obsolete in version 41.3, I don't plan to fix this.
- * Modified Makefile to remove ixprefs.c.orig.

Version 2.0

beta 1, July 12, 1995

- * Made the following modifications in preparation for the integration of ixprefs with the forthcoming cleaned-up version of ixemul.library:
 - + Dropped support for ix_unix_pattern_matching, ix_no_ces_then_open_console, ix_ignore_global_env, ix_translate_dots, ix_force_translation, and ix_translate_symlinks. These settings will no longer be honored by ixemul.library, as their values will be fixed.
 - + Added support for the new ix_allow_amiga_wildcard setting.
 - + Rearranged the GUI accordingly.
 - + Applied Kamil Iskra's modifications for AmigaOS 3.x to the resulting ixprefs.c, and created a new ixprefs.c.diff.
 - + Changed the save format from human-readable to machine-readable (struct settings). I know many people will disagree with this decision, but it was necessary for easier parsing of the configuration by ixemul.library.
 - + Added struct settings to ixemul.h.
 - + Modified the definition of the ixemul flags in ixemul.h from a bitfield to a long word, with a set of #define's for the various flags.
 - + Changed the location of the save file from S:ixprefs to ENVARC:IXPREFS. The "save" and "use" options also save the configuration in ENV:IXPREFS.
- * Modified main.c to call ixprefsrestore() to load the ixprefs settings from ixemulbase instead of duplicating ixprefsrestore()'s code.
- * Updated the documentation.

beta 2, September 16, 1995

- * Saved configuration now includes values for obsolete flags, so that ixprefs may work with earlier versions of ixemul.library.
- * Cleaned up the code by compiling with -Wall and eliminating the warnings.
- * Changed the name of the configuration file from IXPREFS to ixconfig.prefs, to avoid a clash with the ixprefs executable in the unlikely case that someone might have ENV: or ENVARC: in their path.
- * Fixed a typo ("upto" instead of "up to") in the text printed by the --help option.
- * Added a version string to main.c.
- * Added support for the new ix_do_not_flush_library setting.
- * Applied Kamil Iskra's modifications for AmigaOS 3.x to the new version of ixprefs.c, and created a new ixprefs.c.diff.
- * Updated the documentation.

beta 3, October 25, 1995

- * Corrected the punctuation of the configuration report in CLI mode.
-

- * Updated private copy of ixemul.h to reflect the changes in the official version (41.4).
- * Added two paragraphs that were missing from the GPL notice in the source files, Makefile, and documentation.
- * Changed the method of modifying ixprefs.c to add support for AmigaOS 3.x features: instead of applying a diff file to ixprefs.c (clumsy, as the diff file does not work if the GUI is changed) I now use a sed script that modifies ixprefs.c on the fly (this should work even if the GUI changes, and should work for other projects, too). Kept a copy of ixprefs.c.diff around, just in case...
- * Recomplied using -fbaserel. The resulting executable is almost 2K smaller.
- * Added back support for ix_ignore_global_env. Support for this option will not be dropped from future versions of ixemul.library.
- * Made minor modifications in the help text in cli.c (s/don't/do not/).
- * Updated the documentation.
beta 4, October 25, 1995 (not distributed)
- * ix_do_not_flush_library and ix_ignore_global_env were not being restored when doing a "last saved".

Version 2.1 November 14, 1995

Adapted for ixemul.library 42.0 by Hans Verkuil:

- * Independently applied a similar fix to that of version 2.0 beta 4.
 - * Finished merging ixprefs into the main distribution.
 - * Changed several things because all library base changes (such as the config settings) now go through library calls.
 - * Removed another setting (fibcache, was never used).
 - * Added a new cycle gadget for future networking support. The library accepts it, but currently ignores it.
 - * Rearranged the gadgets in the GUI, and changed the wording in the ixprefs window and the output of the --report option, for a more aesthetically pleasing appearance.
 - * Updated the version check in main.c to check for version 42 of ixemul.library, as ixprefs is now incompatible with version 41.0.
 - * As ixprefs is now incompatible with versions prior to 42.0, ixprefs no longer stores default values for obsolete flags.
 - * Switched back to compiling without -fbaserel, as we had a manifestation of the infamous "spilled register" problem.
-
- * Threw away ixprefs.c.diff, as the sed script proved capable of handling the massive rearrangement of the GUI that occurred.
 - * Removed the various tooltypes from ixprefs.info, which I had forgotten to delete when copying the icon from sys:prefs.
 - * Updated the documentation.

Version 2.2 March 17, 1996

- * Adapted for ixemul.library 43.0 by Hans Verkuil (added support for ix_create_enforcer_hit).
- * Minor cleanup (updated version to 2.2).
- * Updated the documentation.

Version 2.3 August 6, 1996

Adapted for ixemul.library 44.0 by Hans Verkuil:

- * Added support for ix_profile_method_mask.
 - * Updated the version check in main.c to check for version 44 of ixemul.library, as ixprefs is now incompatible with version 43.0.
-
- * Minor cleanup (updated version number, Copyright date, author's e-mail

address).
* Updated the documentation.

Version 2.4 September 23, 1996
Adapted for ixemul.library 45.0 by Hans Verkuil:
* Removed stack watcher and red zone.
* Updated the documentation.

1.28 Author

Kriton Kyrimis

e-mail: kyrimis@cti.gr

"Officially, I am here quite unofficially!"

1.29 Distribution

 Ixprefs v.2.4--ixemul.library configuration program
Copyright © 1995,1996
 Kriton Kyrimis
 This program is free software; you can redistribute it and/or ↔
 modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

GUI designed using GadToolsBox 2.0c by Jan van den Baard

1.30 Requirements

Ixprefs requires version 45.0 or higher of ixemul.library. It will refuse to
run under previous versions, as they had fewer configuration options,
maintained in a different order in the ixemul library base.

1.31 Acknowledgments

Kamil Iskra (iskra@student.uci.agh.edu.pl) made several ↔
enhancement
suggestions and contributed code for adapting ixprefs to AmigaOS 3.x
(see program
history
).

Hans Verkuil (hans@wyst.hobby.nl) was kind enough to consider integrating
ixprefs with ixemul.library, and to let me know in advance of the changes he
planned to make to the library, so that I could adapt ixprefs on time.
Hans also updated ixprefs from version 2.0 beta 4 to 2.1, and has also
updated the program to versions 2.2, 2.3 and 2.4.
