

TolleUhr

COLLABORATORS

	<i>TITLE :</i> TolleUhr		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	TolleUhr	1
1.1	TolleUhr Documentation - Contents	1
1.2	TolleUhr Documentation - Introduction	1
1.3	TolleUhr Documentation - Requirements	2
1.4	TolleUhr Documentation - Installation	2
1.5	TolleUhr Documentation - Usage	3
1.6	TolleUhr Documentation - Arexx interface	5
1.7	TolleUhr Documentation - Technical stuff	6
1.8	TolleUhr Documentation - History	6
1.9	TolleUhr Documentation - Future	8
1.10	TolleUhr Documentation - Catalogs	8
1.11	TolleUhr Documentation - Acknowledgments	8
1.12	TolleUhr Documentation - Disclaimer	9
1.13	TolleUhr Documentation - Copyright	10
1.14	TolleUhr Documentation - Author	10

Chapter 1

TolleUhr

1.1 TolleUhr Documentation - Contents

TolleUhr 1.7

written 1993/94 by Matthias Fleischer
enhanced 1994/97 by Gunther Nikl

Public Domain

Introduction	What is TolleUhr?
Requirements	What system do I need?
Installation	Getting started
Usage	All available options
Arexx support	Arexx interface
Technical stuff	About the Source
History	What is new in this version?
Future	Possible enhancements
Acknowledgments	Who I want to thank.
Disclaimer	Short: Use it at your own risk!
Copyright	About legal issues.
Author	Where you can reach me.

1.2 TolleUhr Documentation - Introduction

Introduction

This is yet another Workbench-Clock-Utility ???!?! Not quite :-)

TolleUhr is an analogous clock for the workbench (or any other public screen) and looks really nice.

The functionality in short:

- nice outfit
- mini-menu
- cli && workbench support
- optional second hand
- custom background-pattern
- custom background-image (3.x only)
- current date in the screen title
- can grab its screen background
- localized
- and more

1.3 TolleUhr Documentation - Requirements

Requirements

TolleUhr requires at least v37 of the OS. The plain version will run on every 680x0, the other versions require a 68020 or higher. TolleUhr (standalone or library) does work with gfx-cards.

One last requirement: you need an Amiga (but this should be obvious :)

The program has been tested on:

- an A1200, 68020, WB3.0
- an A4000, 68030/882, WB3.0
- an A4000, 68030/882, WB3.0, Retina Z3
- an A4000, 68060, WB3.0, PicassoIV

NOTE: The version compiled for the 68060 does require a 68020 only. The old 020 version compiled with gcc 2.3.3 was already a "060" version due to the code generation of this ancient gcc version. Nethertheless I do provide a version compiled for a 68020+.

1.4 TolleUhr Documentation - Installation

Installation

You have to decide which kind of TolleUhr you want use. You can select between the normal one and the shared library.

For the ordinary type only two files have to be copied (the program and its icon), so I do not provide an installation script. TolleUhr likes to be placed in the WBStartup drawer, but you can copy it to any location (preferable on a harddisk). The library version needs some more work: a library version has to be copied to a LIBS: location and 'TuLoader' should go to the WBStartup drawer as 'TolleUhr'.

Hint: If you don't want TolleUhr reside on your system partition, you can place a project info file in the WBStartup drawer with TolleUhr as default tool. This technique can be used for any other program.

Please select yourself a version that suits your needs and/or requirements.

TolleUhr supports the locale.library. All available catalogs can be found in the catalogs drawer. To install a catalog copy it to "LOCALE:catalogs/<your language <-> >".

Since 1.31 the program is pure thus I turned TU into a shared library. I recommend to use the library version since the library checks for the required processor (<-> the standalone program itself does not).

1.5 TolleUhr Documentation - Usage

Usage

TolleUhr can be started either from CLI/Shell or from the Workbench. Settings will always be saved to the programs icon (will be created if it doesn't exist). Sending a Ctrl-c to TolleUhr causes the program to abort immediately. Please note: its not recommended to edit tooltypes that require a numerical value, better configure this with from the menus.

Menus:

You can set anything but pubscreen, taskpriority and background image from menus and save it to the tooltypes field with the 'Save prefs' menu. There is also an alarm-function from menus.

Gadgets:

There is an invisible close-gadget in the upper left corner of the window, an (also invisible) depth-gadget in the upper right, a sizing-gadget in the lower right corner and a drag-gadget in the middle of the window.

Background image:

An image as window background is only available with OS3.x, since this feature uses the datatypes.library. You can use any image format as long as a datatype for the image type has been installed (and activated).

IMAGE - name of a custom picture (including path)
WBIMAGE - use wb-window image. By default the program will use the wb-screen image if no custom image was given. If the Workbench does not use background images TU won't have a background image either.
REMAP - by default the 'picture.datatype' remaps a picture. Here one can specify how the remapping should be done (see below)

The 'automatic' image grabbing can be disabled by supplying an empty image name with IMAGE="".

Please note, that this feature can need *much* memory! Window resizing will become very slow on a screen with many colors. However, this affects only native graphics (with a slow processor :)

The arguments:

Shell-Template: (for a short description see below at the tooltypes)

TOP/N, LEFT/N, WIDTH/N, HEIGHT/N, SECONDS/S, PATTERN/K, OVAL/S, SHADOW/S, SHOWFACE/N,
HANDTYPE/N, HANDWIDTH/N, DRAWPENS/K, BORDERTYPE/N, WINDOWTYPE/N, CHIME/N,
ALARM/N, SCREENNOTIFY/S, WBIMAGE/S, REMAP/N, IMAGE/K, PUBSCREEN/K, TASKPRI/N

Tooltypes:

TOP - number of Pixels in top of the window

LEFT - number of Pixels left of the window

WIDTH - width of the window in pixels

HEIGHT - height of the window in pixels

SECONDS - 1: seconds on 0: seconds off

PATTERN - 0000-3333

OVAL - 1: oval face 0: rectangle

SHADOW - 1: shadow on 0: shadow off

SHOWFACE - 0: show minute marks 4: show no marks at all

HANDTYPE - 0: line 1: triangle 2: rhombus 2: rectangle

HANDWIDTH - 0: very thin 4: very thick

DRAWPENS - 00000000000000-33333333333333

BORDERTYPE- 0 no border 5 full border

CHIME - 0 no chime 4 full functioning chime

ALARM - (saved) alarm time
(not recommended to set from outside the program!)

PUBSCREEN - name of public screen

TASKPRI - priority of the clock-task (should be between -5 and 5)

IMAGE - name of a picture to be used as background

WBIMAGE - use wb window image (default: wb screen image if no custom
image was given)

REMAP - used to specify how to remap an image with the picture.datatype

0 - do NOT remap

1 - remap with PRECISION_IMAGE

2 - remap with PRECISION_EXACT

SCREENNOTIFY

- use 'screennotify.library' by Stefan Becker. Screen notification will only be activated if TU resides on the workbench screen. TU does *NOT* require 'screennotify.library' to run!

WINDOWTYPE- sets certain properties for the main window (numerical!)

- > Close Gadget: activates the window close gadget
- > Backdrop : open main window behind all other windows
- > Immovable : resizing and moving impossible
- > Use Image : display an image (if available)
- > Transparent : grab the screen background covered

Please configure THIS from the corresponding menu entry!

The background-pattern:

It is set by a decimal number of 4 digits (abcd) and gives a background-pattern that looks like this:

```

abababababab  For example: 0110 gives dark grey
cdcddcdcddcd           0220 gives light grey
abababababab           0123 looks a bit like jeans-style
cdcddcdcddcd

```

numbers greater than nine are set as: xaabbccdd (hex)

The drawpens:

```

abcdefghijklmn
a color for second hand
b color for minute hand
c color for minute hand outline
d color for hour hand
e color for hour hand outline
f color for shadow
g color for 12'o clock mark
h color for quarters
i color for hour marks
j color for minute marks
k border color 0
l border color 1
m border color 2
n border color 3

```

You also can use xaabbccddeeffgghhijjkkllmmnn.

1.6 TolleUhr Documentation - Arexx interface

AREXX interface

When TolleUhr is running it creates a public port called "TOLLEUHR.x" with x being a number between 1 and 99. The actual value for this number depends on

the count of running instances of TolleUhr and which value is not used. TU shows its AREXX portname as title of the about request.

The AREXX interface does (currently) know only three commands but this may grow in the future :-)

Supported commands:

QUIT - causes TolleUhr to shutdown

HIDE - sends TolleUhr in the background

SHOW - opens the main window of TolleUhr

1.7 TolleUhr Documentation - Technical stuff

Technical stuff

TolleUhr was compiled with gcc 2.7.2.1 (ade version from april '97 with 68060 patch applied). Since it doesn't need '-fbaserel' the program can be compiled with every other gcc version >= 2.3.3. To be able to recompile the program the 'new' proto includes have to be installed (distributed since gcc 2.6.1). Since version 1.5 both old and new inlines are used thus it was possible to make one library base local in a function. I didn't completely switch to the new inlines since the program became bigger than with with the old ones. TolleUhr can be compiled with slightly changes with other compilers although to make sense it needs some heavier changes. The program is designed for compilation with gcc since it uses a feature not available with other comilers (AFAIK). With gcc one can declare a `_global_` register variable and assign a certain processor register to this declaration. Thus the register will be reserved and not allocated for any other purpose. That gives global library bases needed for the old inline functions. Although the new inlines would allow to pass the pointer to the global variable space as the first function argument and the library bases would be taken from this argument, I still prefer the old method.

Initially, TolleUhr was compiled with **cross-compilers** on a P90 running Linux (thanks RANDi :-). Without the ability to compile on his machine the development would have been a real pain. One compilation cycle with full optimization lasts about 180 seconds on an A4000/030 compared to about 10 seconds on the P90... Now, that I own a 68060 and enough ram I can compile TolleUhr without too much waiting on my amiga again - it takes only about 20 seconds to compile :^)

1.8 TolleUhr Documentation - History

History

- * 1.7: - doesn't require audio.device to be present anymore so it works on a DRACO (thanks to Bernhard 'ZZA' Moellemann for the hint :)
- background pattern clears the window rastport first for workbench

- emulations that don't support patterns (Tobias, when will P96 can do it? ;-)
 - does support the picture.datatype V43 now - PMODE_V43 and friend bitmap tags used (Frank Mariak)
 - replaced NOREMAP switch with REMAP where one can specify which precision to use for remapping
 - added an arexx port to be able eg. to shutdown TU through AREXX (implemented on request by Georg 'Gio' Magschok)
 - added an alarm time switch (several requests for this)
 - library initialization rewritten in C which fixes a bug in LibInit() and utilizes an V37 introduced feature of InitResident() (NOT documented in the AUTODOCS but in the V37 release notes as Olaf 'Olsen' Barthel told me)
 - 68060 version which *does* work on an 020+
 - removed 'MatschigWB' icon from distribution (better use NewIcons (tm) instead)
 - more catalogs (czech and suomi)
 - does use HUNK_RELOC32SHORT ('manually' converted with the Hunk program from T.Richter)
- * 1.6:
- added a 'transparent' switch -> grabs the screen background
 - bitmap handling now OS3.x compliant (AllocBitMap,GetBitMapAttr)
 - fixed a screennotify problem (would forget all graphics stuff)
 - crashed on exit if datatypes.library wasn't available
 - could make enforcer hits if no main window was open
 - backdrop mode doesn't have a depth-gadget anymore
 - does again use 'DoMethod()' for bitmap rendering
 - alarm aborts if main window becomes activated
 - does finally recognize a time change properly
 - initial alarm hour wasn't properly set
 - fixed broken smart hourchime
- * 1.5:
- fixed a really serious bug introduced in the first rewritten 1.3 version :-((I guess it was 'cut and paste'...)
 - found an ancient 'bug' that delayed the display of a doubled second hand when using a pattern
 - screentitle shows the current date with activated main window
 - new option WINDOWTYPE to configure the main window (replaces CLOSEGAD and USEIMAGE)
 - linking with libamiga.a no longer required
 - svenska and french catalog included
 - bigger minute marks ;-)
 - some minor optimizations
- * 1.4:
- fixed stack problem when loading an image with the default stack of 4096 bytes (only affected the CLI, wb had a stacksize of 6144 ...)
 - uses a workbench image if no custom image was given
 - didn't always disable 'Use Image' menuitem
 - forgot to free a buffer - oops
 - changed public screen handling
 - added screen notification support
 - fixed some low memory problems...
 - assembler version removed ;-(
- * 1.31:
- completely rewritten to get almost the same structure as the asm version (only 2k larger than the asm version!)
 - residentable without the help of a startup-code
-

- can be turned into a shared library
- does now support locale.library
- * 1.3: - changed window handling to shared IDCMP
- fixed low-memory bugs (eg. return value of BuildEasyRequest)
- added support for a background image via datatypes (OS3.x only!)
- better icon handling (if started from CLI)
- taskpriority now defaults to 0 (values >0 cause the mysterious system 'hangs' - a v39 only problem !?)
- * 1.21: - removed a bug in saveprefs() that created blank lines
- window will now be opened with 'WA_AutoAdjust' (so the window will open almost always)
- some code cleanups
- * 1.12: - added NewLookMenus for OS3.x
- area buffer was to small
- * 1.11: - initial release

1.9 TolleUhr Documentation - Future

Future

As you may know a program is never ever complete :^) There are always requests for some enhancements or changes. This includes:

- turning TU into a commodity
- additional seconds displays
- launching commands at alarm time
- nicer alarm beep (playing samples through the datatypes)
- a powerpc version (easy, just recompile 8)
- more catalogs
- ...

1.10 TolleUhr Documentation - Catalogs

I always appreciate new or improved catalogs. So if you have a translation or an improved catalog, please send it to me. Eg., I know there exists an italian catalog (thanks for the postcard Melo).

1.11 TolleUhr Documentation - Acknowledgments

Acknowledgments

Thanks to all who supplied suggestions for improvements, enhancements or showed general interest.

- Matthias Fleischer
for writing the original program and for libnix :^)
- Piere Carette && Walter Doerwald
for their ImageBackfill example, that showed me, how easy it is to use the datatype.library and for the great "CopyTiledBitMap()" function
- Thomas Clancy
for his suggestion to 'grab' a workbench image (so one can have random background images on the wb and TU will use that image)
- Markus Wild
for porting gcc to the amiga (for -fbaserel I am using still his port of ↔
2.3.3)
- Randolph 'RANDi' Schultz
for allowing me to install 3 cross-compilers on his P90 running linux and stressing his computer
- Samu Nuojuua
for his assembler SNMA (used to assemble the - now removed - asm version, btw. A68k is incredible fast on a P90 :-)
- Mario Cattaneo
for his suggestions and testing of various intermediate versions
- Georges 'Melkor' Goncalves, Mattias Sandgren, Kertai Gábor and Petr Skala
for supplying catalogs
- Norma Lopez de Chavez
for updating two translations

1.12 TolleUhr Documentation - Disclaimer

Disclaimer

Standard disclaimer:

THERE IS NO WARRANTY FOR THE SOFTWARE TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHERE OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE SOFTWARE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY

SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE SOFTWARE AS PERMITTED BELOW, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE SOFTWARE TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.13 TolleUhr Documentation - Copyright

This program is public-domain, that means you can copy, modify or use it as long as you wish and you needn't pay anything.

BUT: It must be absolutely clear at any time if and by what person this program was modified - there must be a FULL list of ALL the authors who made ANYTHING about this program at least in the source code and also in any about-requesters.

1.14 TolleUhr Documentation - Author

Author

To send suggestions, bug reports, comments, gifts, flames, etc., you can contact me at one of the following addresses:

email: gnikl@informatik.uni-rostock.de (prefered)

or

snail: Gunther Nikl
Ziegendorfer Chaussee 96
Parchim
19370
GERMANY

The first released version of this program was written by Matthias Fleischer.

email: fleischr@izfm.uni-stuttgart.de

or

snail: Matthias Fleischer
Adlerstraße 30
73760 Ostfildern 2
GERMANY

(don't ask `_him_` for any enhancements since this is `_my_` part since ages :-)
