

**TolleUhr**

**COLLABORATORS**

	<i>TITLE :</i> TolleUhr		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 19, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>TolleUhr</b>	<b>1</b>
1.1	TolleUhr Documentation - Contents . . . . .	1
1.2	TolleUhr Documentation - Introduction . . . . .	2
1.3	TolleUhr Documentation - Requirements . . . . .	2
1.4	TolleUhr Documentation - Installation . . . . .	2
1.5	TolleUhr Documentation - Usage . . . . .	3
1.6	TolleUhr Documentation - Known Bugs . . . . .	5
1.7	TolleUhr Documentation - Technical stuff . . . . .	5
1.8	TolleUhr Documentation - History . . . . .	6
1.9	TolleUhr Documentation - Acknowledgments . . . . .	7
1.10	TolleUhr Documentation - Disclaimer . . . . .	8
1.11	TolleUhr Documentation - Copyright . . . . .	8
1.12	TolleUhr Documentation - Author . . . . .	9

---

# Chapter 1

# TolleUhr

## 1.1 TolleUhr Documentation - Contents

TolleUhr 1.31

written in 1993/94 by Matthias Fleischer  
enhanced in 1994/95 by Gunther Nikl

Public Domain

Introduction	What is TolleUhr?
Requirements	What system do I need?
Installation	Getting started
Usage	All available options
Known bugs	What does not work?
Technical stuff	About the Source
History	What is new in this version?
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)
- 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 version requires a 68020 or higher. TolleUhr should work with gfx-cards. The same applies to the libraries. 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

## 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 don't want TolleUhr reside on your system partion, you can place a project 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. Both versions support the locale.library. A catalog for german can be found in the catalogs drawer. To install the catalog copy it to "LOCALE:catalogs/deutsch".

Since both the asm and the c version are now pure I turned TU into a shared library. Why one will ask. Ok, the explanation:

Some time ago there was a discussion on comp.sys.amiga.programmers that making programs resident is not a clever decision since that has to be done by hand (also to remove the program later). A library has the advantage that the system will handle this. If memory is needed the system will flush the library.

## 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.

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 picture (including the path)  
 USEIMAGE - show the image  
 NOREMAP - by default the 'picture.datatype' remaps a picture. This prevents color remapping - check it out (seems to be useful for MagicWB 2.0)

Please note, that this feature can need \*much\* memory! Window resizing will become very slow on a screen with many colors. This should affect only native graphics, not graphic cards.

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, CHIME/N, CLOSEGAD/S, USEIMAGE/S, NOREMAP/S, 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 - 00000000000000-3333333333333333  
 BORDERTYPE- 0 no border 5 full border  
 CHIME - 0 no chime 4 full functioning chime  
 CLOSEGAD - activate the window close gadget  
 PUBSCREEN - Name of public screen  
 TASKPRI - Priority of the clock-task (\_should\_ be between -5 and 5)  
 IMAGE - name of a bitmap to be used as background

USEIMAGE - display the image (if available)

NOREMAP - disable color mapping

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 - Known Bugs

Known Bugs

\*\*\*\*\*

Currently none :^)

## 1.7 TolleUhr Documentation - Technical stuff

TolleUhr exists as assembler and as C version. They differ only marginally:

The assembler version accepts DRAWPENS and PATTERN starting with '0x' or '\$', not only 'x'. (The C version will look a bit strangely with those prefixes ;-) Furthermore the asm version is a little bit shorter :-)



All other things work in both versions the same! A plain and a 68020 version is available (check the version strings).

C  
-

The C versions have been compiled with gcc (2.7.0 was used for the plain version and 2.3.3 for the 020). Since the new 1.3 version only gcc is able to compile the the program in its current form since it uses one feature not available with other compilers (AFAIK).

(For the freaks: With gcc one can declare global register variables. TU uses such a variable to keep all variables 'globally' visible that are needed by several functions and to give gcc 'global' library bases. There is no other way to use structure elements as library bases with gcc's inlines in its current form. There exists a solution, but until now nobody created new inlines that would support this.  
With SAS/C there would be a better solution. There one would always simply pass the address of the "global structure" as the first function argument since SAS/C can use non-global library bases.)

I used 2.7.0 and 2.3.3 to compile the clock. Since it doesn't need '-fbaserel' anymore one can use every other gcc version (I tested 2.6.3). The 'new' proto includes for gcc have to be installed (distributed since gcc 2.6.1).

Asm  
---

The assembler version also comes in two flavors. Its my first try with 68020+ instructions and I am not sure its worth the efforts. Anyway there is a version for 'bigger' processors. The source for this version is supplied as a diff-file, so you need a 'patch' program, eg. from the gcc-distribution :-). Additionally you need an assembler capable of handling 68020 instructions. I tested PhxAss but I used SNMA (a really good one, but only few people seem to use it) to create the 68020 executable. A68k (v2.71) is sufficient to create the plain version.

WARNING: SNMA 2.03 (8-Oct-95) \*cannot\* be used to assembler TU since it has a bug with pc-relative addressing. So be careful!

## 1.8 TolleUhr Documentation - History

History  
\*\*\*\*\*

\* 1.31: both:

- can now be turned into a shared library

asm:

- fixed some wrong comments in the source

c:

- completely rewritten to get almost the same structure as the asm

- version (only 2k larger than the asm version!)
- does now also support catalogs (uses the same as the asm one)
- residentable without the help of a startup-code

\* 1.3 : both:

- 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 !?)

asm:

- fixed alarmSet() - didn't set st/mi
- reduced overall stackusage
- savePrefs() rewritten

\* 1.21: both:

- window will now be opened with 'WA\_AutoAdjust' (so the window will open almost always)

asm:

- several minor improvements (can't remember exactly what)
- reduced stackusage
- tweaked :-)

c:

- removed a bug in saveprefs() that created blank lines
- some code cleanups

\* 1.12: - added NewLookMenus for OS3.x  
 - area buffer was to small  
 - assembler version created

\* 1.11: - initial release

## 1.9 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

- Mario Cattaneo

for his suggestions and testing various intermediate versions

- Henryk Richter

for his interest in the further development of TolleUhr

- Markus Wild

for porting gcc to the amiga (for -fbaserel I am using still his port of 2.3.3) ←

- Samu Nuojuua

for his assembler SNMA (used to assemble the asm version)

## 1.10 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.

## 1.11 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.

## 1.12 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, he dropped the development - G.N.)

---