

00611d88-0

Timo Kaikumaa

COLLABORATORS

	TITLE : 00611d88-0		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Timo Kaikumaa	November 24, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	00611d88-0	1
1.1	TruePub - The Front Page	1
1.2	TruePub - Introduction	2
1.3	TruePub - Theory of Operation	2
1.4	TruePub - Installation	3
1.5	TruePub - Usage	3
1.6	TruePub - Examples	5
1.7	TruePub - Preferences File	6
1.8	TruePub - Return Codes	6
1.9	TruePub - Bugs	7
1.10	TruePub - History	8
1.11	TruePub - Legalities	9
1.12	TruePub - Index	10
1.13	TruePub - Author	10

Chapter 1

00611d88-0

1.1 TruePub - The Front Page

TruePub V1.0 (31.12.1995)

Public Screen Promoter for Non-Public Screens

TruePub is a small program for promoting practically any screen as public one. Though small in size, TruePub offers full range of features. It can be started from CLI (auto-detaching, no need to run it) or from Workbench. System commodity properties are supported but not required. No reasonable limits exist for preferences that can be given as CLI arguments, Workbench tooltypes or in an external preferences file. All command, expression and argument parsing is done with system routines.

~~~~Introduction~~~~

~How~does~it~work?~~

~~~~Installation~~~~

~~~~~Usage~~~~~

~~~~~Examples~~~~~

~~~~Preferences~~~~

~~~~Return~Codes~~~~

~~~~~Bugs~~~~~

~~~~~History~~~~~

~~~~~Legalities~~~~~

~~~~~The~Author~~~~~

TruePub was developed on an A500, boosted by SupraTurbo28, running OS3.1, equiped with 2+8 megabytes of memory and 300 megs of hard disk storage.

1.2 TruePub - Introduction

Introduction

Amiga screens are really a somewhat clever way to keep different kind of windows apart from each other. But sometimes this is not so good. When there are a lot of screens opened, the user may find it hard to find the particular screen he/she is looking for. This can especially be the case if there is a need for using some Workbench utilities once a while.

Luckily there are so called public~screens, one of them being the default one. Normally this is the Workbench screen, but it can easily be changed (for example, I use commodity called MultiCX to make the frontmost screen as the default one). By changing the default public screen almost all programs normally using Workbench screen for their output windows will start using the new screen for the same purpose. So it's possible to have CLI (shell) or some other window you prefer opened on the same screen you are working on.

Alas, not all screens are public ones. Talking about good old OS1.3, there even wasn't such concept like public screen, so none of these programs will support this feature. Even now, some programs will strictly keep their screens non-public. But this is no longer. TruePub will force any intuition-based screen to become public one. The usage is simple. You just tell the screen names you want to promote and that's it.

You may now wonder why not to promote all screens, why to tell the screen names? This is because not all screens are very system friendly. Although it is possible to use patterns like "#?" matching everything, this is not recommended. Some programs, especially games, will write directly into their screen bitmap and using custom windows in that kind of situation may cause unexpected results. One thing for sure is that the window image seems to disappear.

Anyway, have fun with TruePub!

1.3 TruePub - Theory of Operation

Theory of Operation

TruePub patches three intuition functions, namely `OpenScreen()`, `OpenScreenTagList()` and `CloseScreen()`. `OpenScreen()` calls are redirected into `OSTagList()` to make things little easier and shorter.

When quitting, TruePub takes advantage of the programs called SetMan or PatchControl. Normally, when - let's say - program A patches a function and then there comes program B patching the very same function, the A can't remove patches before B. However, with SetMan (or PatchControl) this is possible because patches will be stored in a linked list. After removing has been done

TruePub waits some time to make sure that nobody is executing the code to be released.

Special care has been taken with overflow errors. The code contains really lot of checks against them.

1.4 TruePub - Installation

Installation

TruePub does not need any extra files, so you can just copy the program wherever you want to. CLI users may delete the icon file (TruePub.info).

In the case you are not familiar with this kind of programs you can now skip to the usage and examples chapters and experiment with TruePub.

If you find the program appropriate in an everyday use you can place it into the WBStartup drawer of your system disk and so it will be started always when you boot up your Amiga. The DONOTWAIT tooltype (see your computer manuals for further info about standard tooltypes) isn't actually required but in some heavy loaded situations - especially on slow machines like the one I have - the Workbench may get tired of waiting for the TruePub to finish all its initialization and that's where the tooltype in question is needed.

For the CLI users there is no need to "run" TruePub. TruePub can be started anywhere in the system startup scripts. The patch removing routines take advantage of the program called SetMan (or PatchControl) so if you have one of these installed then the idea would be not to start TruePub before SetMan (or PatchControl).

See also examples to see some useful hints and tips about using the program.

1.5 TruePub - Usage

Usage

Workbench

When started from Workbench, TruePub first looks for its icon. If it can't find it an error occurs. This can be the case if you use programs like ToolManager or WbRun to start TruePub only simulating the Wb mode.

Three kind of tooltypes will be recognized.

64KBUF

This will make TruePub to allocate 64 kilobytes buffer instead of the normal 1 kilobyte. Try it if TruePub complains about overflow error. TruePub isn't capable of handling expressions larger than 64 kb (should be enough).

SCREEN=<pattern>

Determines the name pattern for screens to be promoted. The SCREEN tooltype can appear more than once, total number of 65535 patterns will be accepted (including patterns in the external~preferences~file). Patterns are standard dos patterns, case sensitive. Furthermore, screens without names are treated as having a pseudo name "UNNAMED". The public screen name for screens promoted by this option will be SCREEN_nnnnn, where nnnnn stands for a numeric value 1..65535. First pattern will have the name SCREEN_1 associated with it and so on.

PREFS=<preferences file>

Reads additional preferences from an external~file. If no PREFS tooltype is specified TruePub search for a file called "TruePub.prefs" first in the current directory, then in the directory where TruePub is located (PROGDIR:), and finally in the ENV: and S: assignments. Using external promote preferences file is the only way to submit user-defined public screen name for the promoted screens.

See also the additional examples.

CLI

Syntax is almost the same as in the Workbench tooltypes. Four possible kinds of arguments will be recognized.

64KBUF

This will make TruePub to allocate 64 kilobytes buffer instead of the normal 1 kilobyte. Try it if TruePub complains about overflow error. TruePub isn't capable of handling expressions larger than 64 kb (should be enough).

QUIT

Quits TruePub. If promoted screens are open the child~process will actually wait for them to be closed.

SCREENS <pattern> <pattern> ...

Determines the name patterns for screens to be promoted. All arguments default to SCREENS type so you don't have to really write "SCREENS" on the command line. Total number of 65535 patterns will be accepted (including patterns in the external~preferences~file) and they are standard dos patterns, case sensitive. Furthermore, screens without names are treated as having a pseudo name "UNNAMED". The public screen name for screens promoted by this option will be SCREEN_nnnnn, where nnnnn stands for a numeric value 1..65535. First pattern will have the name SCREEN_1 associated with it and so on.

PREFS <preferences file>

Reads additional preferences from an external~file. If no PREFS tooltype is

specified TruePub search for a file called "TruePub.prefs" first in the current directory, then in the directory where TruePub is located (PROGDIR:), and finally in the ENV: and S: assignments. Using external promote preferences file is the only way to submit user-defined public screen name for the promoted screens.

See also the additional examples.

1.6 TruePub - Examples

Examples

Here are some examples for CLI users. They can mostly be used in the Workbench tooltypes too, but note that the correspond tooltype for SCREENS parameter is SCREEN (and while the former isn't obligatory the latter is).

TruePub ?

Shows the typical summary of parameters. All other arguments will be ignored. Works only from CLI.

TruePub QUIT

Quits TruePub. All other arguments will be ignored. Works only from CLI.

TruePub a#? "b#?" PREFS=RAM:TruePub.test

Screens having a name beginning with a or b will be promoted (public screen names will be SCREEN_1 and SCREEN_2, correspondingly). Some other preferences are read from the file "ram:truepub.test".

TruePub myscreen xyzy xyzy

Now there can be two promoted "xyzy" screens (plus one "myscreen"). Public screen name for the first "xyzy" will be SCREEN_2 and the second one has name SCREEN_3.

TruePub SCREENS myscreen xyzy xyzy

Exactly same as above.

TruePub PREFS "RAM:VeryBigPreferencesFile" 64KBUF

Maybe there are some huge expressions in the preferences file. Anyway, TruePub uses a buffer of 64 kilobytes instead of usual 1 kilobyte. You can try option 64KBUF if TruePub complains about overflow error.

TruePub (Final*|Word*|DataStore*|TypeFace*)

This will promote four kind of screens, but only one of them can be opened at a time. The asterisk sign (*) can be used as a wildcard (#?) if the corresponding bit in the system DosBase has been set (I myself use commodity

called MultiCX for that purpose).

1.7 TruePub - Preferences File

External Preferences File

The only way to submit a user-defined public screen name for the screens to be promoted is use of external preferences file. Notice that if this external file is present patterns both in the file and in the tooltypes~or~arguments will be processed. Note also that the public screen name doesn't override standard screen name (the name normally written in title bar). Public names will only be needed in programs dealing with public~screens.

Without specifying name for preferences file TruePub will look for a file called "TruePub.prefs" first in the default (current) drawer or directory, then in the directory containing the program itself (PROGDIR:), and finally in the ENV: and S: assignments. Only first occurrence of the preferences file will be noticed.

Just like CLI arguments and Workbench tooltypes the preferences file is dealt with system (dos) routines. Thus all patterns will be recognized. Pattern matching is case sensitive just like in everywhere else. Empty lines and lines beginning with space or tabulator will be skipped. Rest of the lines should satisfy the pattern

```
/A,AS=->=TO/A,/F.
```

Read further to find out what this means. The number of expressions shouldn't exceed 65535 (including those in arguments/tooltypes) and the same limit exists for the number of characters on one line. By the way, there is no need to hit return after the last line in preferences.

Okay, now for some examples. These will all work:

```
Final#?                FWScreen

"Screen name"          TO    "Public screen name"

(A*|B*)                ->    AB_Screen                ...some comments here

...comments also here (see space in the beginning of the line)...

TO "Public screen name" "Screen name"                (in opposite order)
```

1.8 TruePub - Return Codes

Return Codes

Because it's not very wise to promote all possible screens with patterns like "#?" or "UNNAMED" (see Introduction for more info about possible problems) and because some - especially not-so-system-friendly - programs tend to open screens without any names it is possible to use TruePub in scripts too. The promote preferences can be changed on the fly if no promoted screens are open. However, if they do, then the preferences will be updated right after the screens are closed. This will happen automatically although no process is left waiting for it. (TruePub returns as quick as normally in these cases too.)

Anyway, it may be a good idea to know the return codes TruePub has. Here they are:

- 0: - everything was ok - the natural case
 - started from workbench
 - cli argument error occurred
- 5: - tried quitting although already quited
 - the user-defined preferences file wasn't found
 - started TruePub with no promote preferences
 - the promote preferences couldn't be updated right now
- 10: - syntax or overflow error happened during parsing the promote preferences
 - couldn't allocate enough memory or any other error
- 20: - user break (ctrl-c)

Even the most serious errors return only with code 10. The usual code in these cases is 20, but this is because I don't see any point in having a script only for launching TruePub. All other programs should work whether TruePub is active or not and so an error in TruePub shouldn't prevent starting these other processes (script execution will usually stop if return code 20 is encountered).

1.9 TruePub - Bugs

The Bug Page

Well, nothing serious here. Just two things:

First, the way TruePub uses for launching its child~process isn't the most sophisticated one. The child is just started! No memory or segment lists will be added to the entries in the process structure. The child process will take care by itself to free its allocated memory before exit and everything should work fine. However, child's memory area (hunks) may appear somewhat strangely on monitor programs like ARTM or Scout.

While testing the external preferences option on my system with very large

preference files (say, over 200 kilobytes in length) I sometimes noticed TruePub saying that an overflow error occurred. But, there was nothing wrong in the file – and most of the time it worked fine. Don't know why this happened. Mail me if this kind of behaviour causes troubles and I'll try to track it down.

1.10 TruePub - History

History

- v0.01 first 'functional' version. promotion works as i thought. patched OpenScreen() and OpenScreenTagList()
 - v0.02 actually same as v0.01. seems to promote even public already screens
 - v0.03 patched CloseScreen() too. however, does nothing with it
 - v0.05 added check for visitor windows in CloseScreen()
 - v0.07 now promotes only one screen per entry (promoting more would be impossible)
 - v0.08 screens can be promoted again after CloseScreen(). should work even if someone uses external program – like artm or scout – to close the screen
 - v0.15 no more promotes all screens – just the requested ones (maximum length for an expression is 64 kbytes!)
 - v0.16 works only with processes (can task open a screen?)
 - v0.21 uses MatchPattern() for screen name check
 - v0.22 task switching disabled when checking screen attributes. messageport added
 - v0.28 screen attributes are copied to the stack – more safer now
 - v0.35 operates now from child process. no need for running anymore
 - v0.40t checks other alternative patterns too, if the screen whose pattern matched first is already opened. experimented with reading inputs
 - v0.50 handles inputs (wb: 64kbuf, prefs, screen; cli: quit, 64kbuf, prefs, screens) correctly. uses dos.library to parse screennames
 - v0.60 parses public screen names too
 - v0.62 responses are now valid. faked null screentitle used for screens with no title in internal routines
 - v0.66 patches use external preferences instead of inlined ones. use of
-

flags instead of forbid/permit

- v0.70 child's data area no longer exists in truepub. loader responses even more valid. fixed possible bug in patch (with null tag array). null screen names will be treated as having name UNNAMED to make pattern matching easier
- v0.74 communication between child and its parent is made with signals. patches recognizes quit option and signal child whenever promotion list becomes empty
- v0.77 code cleaned up a bit. buggy full functional version. supports setman
- v0.80 more cleaning up. commodity support
- v0.86 unsuccessful experiment
- v0.90 unsuccessful experiment
- v0.91 works also without commodities. quit option seems to work quicker (broker and messageport are removed immediately). child recognizes ctrl-c signal. return codes should be ok
- v1.0 fixed bug in 64kbuf. added a note if icon file couldn't be found. ctrl-c response incredible fast. everything seems to work now (the features have actually been tested). overall polishing

1.11 TruePub - Legalities

Legal Stuff

TruePub 1.x is malware!

If you think this program is worth of using or has serious bugs then please mail me and let me hear about your opinion. Without feedback this will most probably be the final release. After all, the features TruePub now offers should be part of some multi-purpose commodity program (this is what I think).

Copyright

TruePub is written by Timo~Kaikumaa. No parts of this program may be altered by any means (this includes editing, reprogramming, crunching, resourceing etc.), except archiving.

Disclaimer

You are using this program at you own risk! Under no circumstances will the author be liable for any direct or indirect damage or data loss resulting from the use or misuse of this software or the documents.

Distribution

TruePub can be freely copied as long as it is distributed with all the files in this package with it. Only a nominal fee for costs of magnetic media may be accepted for distributing this piece of software, the amount of US \$5 shouldn't be exceeded for a disk containing TruePub. CD Manufactures are specifically granted the right to include this program on CD collections, as long as they are for the Public Domain.

Special permission is given to Brian Aagaard Petersen to include TruePub in his EqEd program package as long as no profit is made with it.

1.12 TruePub - Index

The Index

Child process

When started for the first time, TruePub leaves a so called child process in the memory. This way there is no need for "running" the program (when started from CLI) and the child also handles commodity messages. The quit option removes child process from the memory.

Public screen

Unlike custom screens, public screens can be used by other programs too. For example, promoting the screen of FinalWriter (famous word processor) public would allow one to have EqEd (equation editor program) opened on the very same screen. This helps doing things as you don't have to jump between Workbench and FinalWriter screen all the time.

1.13 TruePub - Author

The Author

Timo Kaikumaa

Atanväylä 14 C 12
33580 Tampere

Finland

E-mail: timok@cs.tut.fi

Home page (currently in Finnish only):
<http://www.cs.tut.fi/~timok>
