

---

Subject: : Official AmigaOS4 feedback

Topic: : Public screen closing bug in OS4.1 (FIXED)

Re: Public screen closing bug in OS4.1

Author: : whose

Date: : 2011/6/16 18:02:36

URL:

@ChrisH:

I still dont believe in an OS4 bug, despite your experiments with MUI programs. The interesting point is the following:

**New for V36:** this function will refuse to close the screen if there are windows open on the screen or if there are any outstanding screen locks (see LockPubScreen()) when CloseScreen() is called. This avoids the almost certain crash when a screen is closed out from under a window

This one quite clearly states, that the calling process is responsible for a sensible reaction to the case, that a public screen cannot be closed. Its easy as that.

In case of MUI (and related programs) this still doesnt take away the responsibility from MUI or the application using the MUI autoscreen features to check for a successful close of the public screen it opened.

There are **still** cases, where a process opening a public screen could be prevented from closing it. I.e., if a foreign process (CygnusEd in your case, this one doesnt use MUI AFAIK) holds a lock on this public screen on closing time.

Turn it as you like. Its just another application development quirk. People doesnt obey AmigaOS software development rules just to save some cleanup work.

If MUI isnt able to close the screen after the locking process frees all of its screen locks and terminates, its broken. If the OS4.x PublicScreen automatic open feature isnt able to close a (after some time) lock-free public screen it opened, its broken. If any application that opened a public screen itself isnt able to close it after the last visitor window vanished, its broken.

Let me recall this: You **have** to check for a successful close **since V36** and you **have** to take measure for the negative case (i.e. waiting or give up in a Wait() loop, i.e. waiting for an Intuition message. SDK isnt very clear here, sadly).

Some day there may be a more elegant solution to this problem, I suggested some kind of "SAFE\_TO\_CLOSE\_PUBSCR" message from Intuition, maybe some people come up with another solution.

Until that day, all applications/application layers have to check for a CloseScreen() fail and response to it accordingly. This includes a "I give up on this and I free all resources I can free" in the hardest case.

There is no need for a patch or a mere bug fix. All we need is some discipline for the clean up part of our software pieces.

Edit: Forgot to tell, that you as application developer are still free to request the user to close all possible visitor windows in the case CloseScreen() fails on a public screen your application opened. IIRC, there were some applications in ancient times which did exactly that.