

**6824d7f0-0**

**COLLABORATORS**

	<i>TITLE :</i> 6824d7f0-0		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

**REVISION HISTORY**

<i>NUMBER</i>	<i>DATE</i>	<i>DESCRIPTION</i>	<i>NAME</i>

# Contents

<b>1</b>	<b>6824d7f0-0</b>	<b>1</b>
1.1	ReFluX . . . . .	1
1.2	What is it? . . . . .	1
1.3	Contact . . . . .	2
1.4	Usage . . . . .	2

# Chapter 1

## 6824d7f0-0

### 1.1 ReFluX

R e F l u X v1.0B

What is it?

Usage

Contact

### 1.2 What is it?

What is it?

ReFluX is a little tool, inspired by TaskersFlush, which basically does the same as calling "Avail Flush" in the CLI, except that it does it once every 10 seconds, until you send it a Control-C command to stop (using XOpa, Executive's Commander, or similar programs).

Why?

Well, it keeps the maximum amount of memory available at all times. When I run a program (especially MUI programs like Voyager & Miami), I can be left short of 300k or more. I could free it up by typing Avail Flush, but this does it automatically.

The reason I wrote it, is because I wrote to the author of TaskersFlush, asking for something to be added, and haven't yet received a reply, which is fair enough - he might be really busy - but I just decided to have a go myself, and - oh my god! - it works!

How?

I sneaked a look at the PoolMem docs, as referenced by the TaskersFlush docs, which tell you how Avail Flush works. ReFluX

---

just tries to allocate a very large block of memory which magically clears things out. It calls it tens times every ten seconds (Avail seems to do it some weird random number of times, but this works nicely). It takes about 2 "ticks" (1/50ths of a second) to perform this clearance each time, so it's negligible.

What the FluX?!

The name came from Flush, to ReFlush (cos it does it over and over), to ReFlux, because it sounds good :) I don't even know what flux means ;)

## 1.3 Contact

Contact

If you want to make any comments or bug reports (whatever they are - I wouldn't know ;) you can get me at :

jamesboyd@all-hail.freeseerve.co.uk

The source is included for Blitz programmers - if you make any improvements, send 'em to me for inclusion! You'll need to put Blitzlibs:Amigalibs.res into the Compiler Options before compiling...

See ya :)

## 1.4 Usage

Usage

R e F l u X is © James L Boyd 1999

Run it from the Workbench icon, or from CLI (ideally, stick it in your WBStartup drawer).

If you run from the icon, it automatically uses a default delay of 10 seconds between flushes.

I may add a tooltype option to set the number of seconds delay, but you'll have to let me know if you want this, bearing in mind it'll add quite a bit to the file size (at the moment, it's packed with StoneCracker, but it's considerably smaller than TaskersFlush anyway ;)

TaskersFlush's "Tiny" version is a lot smaller still, but you can't quit, and it causes some programs to crash (which this doesn't seem to be doing so far ;) If it causes any crashes for you, let me know, and I'll try and add a check to avoid clearing when certain tasks are running).

---

If you run it from the CLI, you can specify the number of seconds to wait, from 1 to 300 (5 minutes). If you enter an invalid number, it defaults to 10 seconds.

Example (to clear once every minute) : `reflux 60`

Type `reflux ?` , `reflux help` , or `reflux -h` to be reminded ;)

To end ReFluX, just send it a Control-C command, either directly in the CLI you're running from, if applicable, or from a program such as XOPA, ARTM, etc. NOTE : it may take up to the length of time you've set as the delay before it quits, but it will!

That's it :)

---