

# **RunLameInfiltrator**

Bilbo the first

COLLABORATORS
---------------

	TITLE : RunLameInfiltrator		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Bilbo the first	July 26, 2024	

REVISION HISTORY
------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>RunLameInfiltrator</b>	<b>1</b>
1.1	RunLameInfiltrator documentation . . . . .	1
1.2	Index table . . . . .	1
1.3	Disclaimer . . . . .	2
1.4	PrayWare concept explanation . . . . .	4
1.5	RunLameInfiltrator introduction . . . . .	4
1.6	RunLameInfiltrator features . . . . .	5
1.7	RunLameInfiltrator disadvantages . . . . .	6
1.8	Installing RunLameInfiltrator . . . . .	7
1.9	RunLameInfiltrator operation description . . . . .	7
1.10	File(Un)Linker general notes . . . . .	8
1.11	FileLinker operation description . . . . .	9
1.12	FileUnlinker operation description . . . . .	9
1.13	Configuring RunLameInfiltrator . . . . .	9
1.14	RunLameInfiltrator `Problems - solutions - corner' . . . . .	11
1.15	What or who is lame? . . . . .	12
1.16	What a TLA is . . . . .	13
1.17	What a CPU is . . . . .	13
1.18	What the VBR is . . . . .	13
1.19	What a MMU is . . . . .	13
1.20	What OCS is . . . . .	13
1.21	What ECS is . . . . .	13
1.22	What AGA is . . . . .	14
1.23	What a GUI is . . . . .	14
1.24	What MUI is . . . . .	14
1.25	What an API is . . . . .	14
1.26	What is Enforcer . . . . .	14
1.27	Other Degraders available . . . . .	14
1.28	Beta testing of RunLameInfiltrator . . . . .	15
1.29	Known bugs in RunLame . . . . .	16

---

1.30 Future plans with RunLame . . . . . 16

1.31 Features that won't be implemented in RunLame . . . . . 16

1.32 RunLameInfiltrator history . . . . . 17

1.33 First public release . . . . . 17

1.34 Second public release . . . . . 18

1.35 Third public release . . . . . 19

1.36 Fourth public release . . . . . 19

1.37 Thanks and greetings . . . . . 19

1.38 How to contact the author of RunLame . . . . . 20

# Chapter 1

## RunLameInfiltrator

### 1.1 RunLameInfiltrator documentation

---

RunLameInfiltrator

Version 1.10

Copyright 1992-1993 by Bilbo the first

---

Table of contents

Prologue    Disclaimer  
            PrayWare

Introduction    The aim of the game  
                Features  
                The bad news

Installation    Installing RunLameInfiltrator

Operation    How to use RunLameInfiltrator  
              Configuring RunLameInfiltrator  
              How to use File(Un)Linker

Hints        Problems - solutions - corner  
              Known bugs

History       History  
              Future plans  
              Features not to come

Epilogue    Thanks, greetings  
              Contact

### 1.2 Index table

---

## Index

AGA  
API  
Beta testing  
Configuring RunLameInfiltrator  
Contact the author of RunLame  
CPU  
Disadvantages of RunLameInfiltrator  
Disclaimer  
ECS  
Enforcer  
Features of RunLameInfiltrator  
Features that won't be implemented  
File(Un)Linker general notes  
First public release  
Future plans  
GUI  
History  
Index table  
Installation of RunLameInfiltrator  
Introducing RunLameInfiltrator  
Known bugs  
Lame  
MMU  
MUI  
OCS  
Operation description of FileLinker  
Operation description of FileUnlinker  
Operation description of RunLameInfiltrator  
Other Degraders  
PrayWare concept explanation  
Problems - solutions - corner  
Second public release  
Table of contents  
Thanks and greetings  
Third public release  
TLA  
VBR

## 1.3 Disclaimer

LICENSE and TERMS for software originating from Bilbo the first

Copyright © 1993 Bilbo the first

This software is NOT in the Public Domain, NOT Freeware and NOT Shareware.

This document overrides any former documents included with preceding versions of this software package, and is to be considered valid for those, as well as for the package with which it is included.

There are NO warranties, either expressed or implied, with respect to the software, its quality, performance, merchantability, or fitness for any

---

particular purpose. This software is provided "as is" and the entire risk as to its quality and performance is with the user.

The author of this software shall NOT be liable for errors contained in the software or the software's documentation, or for incidental or consequential damages of any kind in connection with the performance, use or possession of this software. The use of undocumented features/commands is not permitted except with permission from the author, and the author is in NO way responsible for consequences thereof.

Anyone may use it for private purposes, and it may be given away for free, as long as it is NOT part of any deal involving purchases; and fees for copying and/or distribution, and/or fees for disks exceeding the actual market price for an empty disk of the same brand/quality is to be considered as commercial exploitation of this software, and an infringement of the Copyrights (©) of the author. The right to use this software and/or its documentation for private purposes may or may not be restricted by additional copyrights and/or registration conditions.

If given away under above (or below) conditions, the receiver in NO way receives the ownership of the program itself, but may only be licenced to use it if above and below conditions are met.

If these rules infringes the local law and/or rights of any area, political or geographical, this software is NOT to be used and/or distributed in these areas.

There may or may not be specific charges for the use and/or spreading of this(these) program(s), and if existant, these charges are to be honored following a free test period of either 3 days or 10 executions of the program, whatever comes first. The payment of these charges do NOT in any way make the user/distributor entitled to any form of support and/or upgrades and/or does NOT give him/her/them/it any rights regarding the software, or the use/distribution thereof, and the Author assumes NO extra responsibility and will retain superior ownership and copyright of the code, parts of code, and/or source of code of all software.

The executionable and/or non-executionable parts of this program or any package consisting of any or all of these parts are NOT to be altered by any means (including editing, reprogramming, crunching, and recompiling), except de-archiving. Note that by changing any part of the software, you will resume all the responsibilities, but none of the rights of the author (Bilbo 1st). The author (Bilbo 1st) is in NO way responsible for any changes made to any part of the package, or consequences thereof.

The name of the executable file(s) may be changed at convenience, as long as this does NOT infringe any copyright of any other author, and by changing the name(s), observe that all rights and/or restrictions given by this text, and/or by the software itself still applies.

The documents contained in the software package contain proprietary information which is protected by copyright (©). All rights are reserved. No part of these documents may be photocopied, reproduced, electronically copied, or translated to another language without the inclusion of the software and other included files as part of a package. The executable files accompagnyng this package may NOT be reproduced or transferred outside a one-machine-environment without all these accompagnyng files including this file. Any translation of any part of the package, to whatever language, has to be included as an additional text file, and should NOT contain the signature of the Author, but MUST be signed by the person(s) doing the translation.

These documents are NOT to be changed by any means, excepting Archiving, if the documents are extractable, and readable after extracting, and the name of the archive is directly spellable using letters "A".."Z", numbers "0".."9" plus punctuation ".", and does NOT contain more than eight (8)

characters plus punctuation(".") plus three (3) characters. This is to ensure compatibility with the MS-DOS(TM) and PC-DOS(TM) file systems.

## 1.4 PrayWare concept explanation

This package is

'PrayWare'

Don't PAY, just PRAY!

This license is an addition to the disclaimer. All restrictions pointed out in the disclaimer keep their validity.

The author (Bilbo the first) doesn't intend to gain any profit from this production. Therefore the user does not need to pay any fees to the author. Though the author constantly suffers from lack of money (most people do, don't we?), but there is a much more valuable thing you can do for him. Something more valuable than cash money? Must be very expensive! Must be very difficult! Right?

Wrong! The author just asks you for something very easy, something which doesn't cost you any money but some minutes of your time. Interested?

Just get in contact with the Lord Jesus Christ and pray. Simply try it, talk to him like you would talk to a very private and good friend. Tell him what happened today, what you did feel like today. Tell him you received a software of the author which you use and like very much (of course only if you really do so :-). And tell him that the author is in need of help for he is often very weak and often fails to love his Lord and Lord's creatures the right way. Just talk with him about everything you like to.

Thanks, that's the best thing you could do for me.

Anyone feeling personally bothered by the concept of PrayWare may just ignore it. Nobody forces you to do things you don't want to do. Of course you though might use the software under the limitations expressed in the disclaimer.

## 1.5 RunLameInfiltrator introduction

The aim of the game

Do you run your system using another screenmode than PAL or NTSC, for example EURO72, an AGA specific screenmode or even a screenmode produced by a graphics expansion board? Have you ever tried to start a game or demo from within this screenmode? Have you ever been angry about those programmers who do not stick to the guidelines of programming thus producing programs which are not aware of the screenmode they were started out of?

Do you have an Amiga with 68010, 68020, 68030 or 68040 processor? Did you ever curse those so-called coders whose programs crash your computer when you

---



didn't disable the processor's caches and forgot to move the vector base register back to \$0.1?

lame programming is the disease, RunLameInfiltrator is the cure!

NEVER start RunLameInfiltrator as stand-alone-program, it definitely WILL crash your system. RunLameInfiltrator MUST be linked to another program to work correctly. This process is called 'infiltration' in the following, and the resulting file of linking RunLameInfiltrator to a program is called 'infiltrated program'.

RunLameInfiltrator is a Kick1.3+ tool, it may work under Kick1.0/1.1/1.2, but I cannot give any warranty for this. RunLameInfiltrator is 100% pc-relative and reentrant Assembler code, so you can link it to pure programs and they will stay pure after linking.

Just after having degraded the display and the CPU RunLameInfiltrator starts the program it has been linked to. When the program it has been linked to has been finished the display and CPU state is restored to the same as before starting the infiltrated program.

Summarizing we can say RunLameInfiltrator is a multitasking friendly temporary system degrader which, linked permanently to files, may let some bad behaving programs run in your system's environment (but can't give any guarantee). If some of those lame programs don't run though RunLameInfiltrator has been used, they are on a higher lameness-level, which can't be fixed by RunLameInfiltrator (this is definitely NOT a bug of RunLameInfiltrator).

I do write demos myself and you can believe, i do know how to write a proper startup code. Consider RunLameInfiltrator beeing a startup code.

## 1.6 RunLameInfiltrator features

### Features

The {}'s inform you about the system configuration needed for a feature.

- Disabling of all processor caches, burst or copyback modes.  
{68020, 68030[EC], 68040[EC]}
  - Moving VBR to location \$0.  
{68010, 68020, 68030[EC], 68040[EC]}
  - Optionally degrading display's screenmode to simulate OCS. This can optionally preserve or hide the current display's contents. Display degradation even works if the current active screenmode is produced by a graphics board (fi. Merlin © X-Pert).  
{ECS, AGA}
  - Degrading sprite resolution to simulate OCS.  
{AGA}
  - Turning off sprites in a system friendly manner.  
{OCS, ECS, AGA}
-

- Permanent fix for lame programs. Fix can be removed.
- Infiltration of pure programs preserves the reentry ability.
- Infiltration of pc-relative programs preserves the location relativity.
- Works under Kickstarts 1.3, 1.4, 2.04, 3.0 and 3.1.
- Due to 100% system conform and system friendly algorithms, future system compatibility is as high as possible. Furthermore the multitasking environment is not hurt or damaged by the code. All degrading is done as smart as possible, and the code did never crash during beta testing.
- verified, tested and safe (100% assembler) code resulted in a short binary (so the infiltrated programs only become a little longer).

Sounds great? YAH – Yet another hype? Hyphenosis – Better believe the hype! But keep on reading, here come... the bad news.

## 1.7 RunLameInfiltrator disadvantages

The bad news

- Because Workbench programs do indirectly ignite an UnLoadSeg() onto themselves, the code could be flushed by this (f.i. Mungwall). Therefore restoring the system is not yet possible when run from Workbench. This includes a WANTED mem leak in the infiltrated program. You will lose around 7K bytes of memory.
  - The infiltrated programs' code is changed permanently (but not irreversibly). If you don't like this, use RunLame.
  - No GUI for FileLinker and FileUnlinker (the infiltration support tools).
  - Dangerous FileUnlinker tool (if you don't know what you are doing).
  - The updating of infiltrated programs is slightly uncomfortable (unlink, and link again).
  - Though the code worked very well during the beta testing phase, there might be nasty bugs.
  - RunLameInfiltrator can't be used to fix non-DOS trackloading demos, for I don't want to support this kind of loaders. I usually like to keep good demos on my harddisk, which is not directly possible with trackmos. So I don't keep lame coded trackmos.
  - The documentation often might use improper terms. Please excuse this, for I am not a native english speaker. If you want to improve the documentation, please contact me.
-

## 1.8 Installing RunLameInfiltrator

### Installing RunLame

Installing RunLameInfiltrator in your system is quite easy. There is no need for an Installer script.

At first enter the directory, which contains the package. The name of the directory is RunLame\_V#?.#? (the #?.#? is the current release number). Then enter the Bin directory and copy to the following files to a directory which is in your system path:

- RunLameInfiltrator
- FileLinker
- FileUnlinker

That's it. You can now infiltrate programs. Refer to the configuration instructions and the operation description.

## 1.9 RunLameInfiltrator operation description

### How to use RunLameInfiltrator

---

ATTENTION!! NEVER START RUNLAMEINFILTRATOR ALONE AS PROGRAM!! IT WILL CRASH YOUR SYSTEM!! RUNLAMEINFILTRATOR MUST BE LINKED TO VALID PROGRAMS.

---

In comparison to RunLame which is included in this package, RunLameInfiltrator performs the following actions (which are RunLame's default options):

S	=	SCREEN	:	degrade display
V	=	VISIBLE	:	the current display stays visible
NC	=	NOCACHE	:	disable instruction & data cache
NIC	=	NOINSTCACHE	:	disable instruction cache
NDC	=	NODATACACHE	:	disable data cache
NB	=	NOBURST	:	disable instruction & data burst mode
NIB	=	NOINSTBURST	:	disable instruction burst mode
NDB	=	NODATABURST	:	disable data burst mode
NCB	=	NOCOPYBACK	:	disable copyback mode
NEC	=	NOEXTERNALCACHE	:	disable external caches
NVBR	=	VBRT00	:	move VBR to \$0.L
CL	=	COMMANDLINE cmd	:	cmd is executed, where cmd is the program

The above explained options are active during the execution of cmd and the changed settings of your system are reset to the state active before. So if you know that for example the demo called ObsoleteFX will not run on your normal system configuration, infiltrate RunLameInfiltrator to ObsoleteFX and try if it will work. To link RunLameInfiltrator to ObsoleteFX, you can use the following procedure:

---

```
FileLinker RunLameInfiltrator name_of_lame_program name_of_output_program  
  
(name_of_output_program is called the infiltrated program)
```

In this special case use:

```
FileLinker RunLameInfiltrator ObsoleteFX ObsoleteFX.rli
```

Now try if the infiltrated program works on your system. When the programmer(s) of the un-infiltrated program were too lame it will of course not work :-). Assuming that ObsoleteFX.rli now works on your system, the following will happen when ObsoleteFX.rli is started:

```
assumed system settings before starting ObsoleteFX.rli:  
data cache active          | instruction cache active  
data burst mode active     | instruction burst mode inactive  
external caches inactive   | copyback mode active  
VBR in Fast RAM at position p
```

```
system settings during execution of ObsoleteFX:  
data cache inactive        | instruction cache inactive  
data burst mode inactive   | instruction burst mode inactive  
external caches inactive   | copyback mode inactive  
VBR at $0.1  
screenmode changed to PAL,  
sprite resolution changed to lores and sprites turned off
```

```
system settings after execution of ObsoleteFX.rli:  
data cache active          | instruction cache active  
data burst mode active     | instruction burst mode inactive  
external caches inactive   | copyback mode active  
VBR in Fast RAM at position p
```

## 1.10 File(Un)Linker general notes

### General notes on File(Un)Linker

At this point I give a short introduction to the programs FileLinker/FileUnlinker, which I have written a long time ago. FileLinker/FileUnlinker is not pc-relative, nor is it reentrant, so you can't make it resident. It can't be started from the Workbench because it doesn't include the Workbench startup code. FileLinker/FileUnlinker works on Kick 1.3+, it may work on earlier Kickstart versions, but I cannot give any warranty for this. Both programs may contain bugs and do not represent my current programming skills (in fact when I read the source lately it gave me nightmares). But because in all the time I was using both programs no bugs have been found, I just release both programs unchanged with this package. An easier to use and similar tool I have written when I finished FileLinker/FileUnlinker. This advanced tool is called FileChainer and has been released to the public, so when you want to link normal programs, get FileChainer for this purpose. In the case of RunLameInfiltrator it is not possible to use FileChainer, so we have to use the both older and much more uglier programs. Of course you can use any other known program that is doing the same, but, I don't know of any other released program that could have been used instead.

---

```
FileLinker
```

```
FileUnlinker
```

## 1.11 FileLinker operation description

How to use FileLinker

What is FileLinker doing exactly? FileLinker is able to insert a ONE HUNK executable file as first hunk into the hunk collection of another executable file. The inserted hunk will be started on execution of the resulting file. This first hunk must execute the other hunks itself, if this is desired. You see, that abilities of FileLinker are slightly restricted, but it is exactly what is needed to infiltrate RunLameInfiltrator into a program.

To infiltrate RunLameInfiltrator to a program use:

```
FileLinker RunLameInfiltrator program_name new_program_name
```

## 1.12 FileUnlinker operation description

How to use FileUnlinker

FileUnlinker irreversibly removes the first hunk of an executable file, so RunLameInfiltrator can be removed from an infiltrated program. ATTENTION FileUnlinker doesn't check if the first hunk of the program that shall be processed really is RunLameInfiltrator, so use of FileUnlinker on an un-infiltrated program results in data loss and the program is irretrievably destroyed. Be careful with FileUnlinker in your own interest. You do best if you check with the Version CLI command of C= if a program really is infiltrated, before you unlink any hunks of it.

To remove RunLameInfiltrator from an infiltrated program use:

```
FileUnlinker infiltrated_program_name removed_program_name
```

Example:

```
FileLinker RunLameInfiltrator FooBar FooBar.rli
```

```
FileUnlinker FooBar.rli FooBar2
```

FooBar2 should be exactly the same as FooBar now.

## 1.13 Configuring RunLameInfiltrator

---

## Configuring RunLameInfiltrator

### Configuring for shell:

RunLameInfiltrator itself is not configurable, but you can simplify the infiltration and dehunking. This can easily be done using some aliases. Replace [path] with the path to the appropriate file.

```
Alias infiltrate '%file [path]FileLinker [path]RunLameInfiltrator $file'
```

```
Alias dehunk      '%file [path]FileUnlinker $file'
```

```
Alias update      '%file [path]FileUnlinker $file;  
                  [path]FileLinker [path]RunLameInfiltrator $file'
```

### Configuring for Directory Opus:

Enter the configuration module and the Buttons section. Press one of the still unused buttons. Enter 'Infiltrate' for Name string gadget. Make a New entry which is AmigaDOS. Enter '[path]FileLinker [path]RunLameInfiltrator {f}' in the entry string gadget (replace [path] by the path where the appropriate file is located). Enter Stack size '4000', Priority '0', Close delay '-1'. Uncheck all entries in the Flags... listview gadget. Check Output window and Do all files in the listview gadget. Click Okay.

Click on another free button. Enter 'Unlink' for Name string gadget. Make a New entry which is AmigaDOS. Enter '[path]FileUnlinker {f}' in the entry string gadget (replace [path] by the path where the appropriate file is located). Enter Stack size '4000', Priority '0', Close delay '-1'. Uncheck all entries in the Flags... listview gadget. Check Output window and Do all files in the listview gadget. Click Okay.

Click on another free button. Enter 'Update' for Name string gadget. Make a New entry which is AmigaDOS. Enter '[path]FileUnlinker {fu}' in the entry string gadget (replace [path] by the path where the appropriate file is located). Make a New entry which is AmigaDOS. Enter '[path]FileLinker [path]RunLameInfiltrator {f}' in the entry string gadget (replace [path] by the path where the appropriate file is located). Enter Stack size '4000', Priority '0', Close delay '-1'. Uncheck all entries in the Flags... listview gadget. Check Output window and Do all files in the listview gadget. Click Okay. Click Okay. Click Save. Click Okay.

Now you should have 3 new buttons in the button bar. The Infiltrate button links RunLameInfiltrator to all selected files. Make sure the files are not write and delete protected. The Unlink button is somewhat DANGEROUS. It will strip the first hunk of each selected file. Make sure the files you want to dehunk are not write and delete protected. THE HUNK STRIPPED OFF THE FILES IS NOT RECOVERABLE, SO THINK TWICE BEFORE YOU PRESS THIS BUTTON!! The Update button will replace an old version of RunLameInfiltrator in an infiltrated file with the latest installed version. BE SURE NOT TO UPDATE NON-INFILTRATED FILES, THEY WILL BE CORRUPTED!!

---

## 1.14 RunLameInfiltrator `Problems - solutions - corner`

### Problems - solutions - corner

- If you got this error message:

```
Screenmode degrading failed!  
Forbid all promotion utilities to promote:  
"RunLameInfiltrator (© by Bilbo 1st of Hypenosis)"  
Display of this lame program might be trashed!
```

Then you have probably installed one of the screenmode promotion utilities (fi. The Promotor, NewMode, ForceMonitor, PKludge). This promotion tool will then promote the compatibility screen too, what is a really stupid idea. You should prevent the promotion utility from promoting all screens with the title "RunLameInfiltrator (© by Bilbo 1st of Hypenosis)".

- After having been degraded, the display snaps back to the normal previous mode when a window has been opened. This window might be owned by the lame program or any other task. The problem with this window is it's ability to bring it's underlying screen to the very front of all screens. The currently frontmost screen now forces the display to it's own screenmode, thus stopping the graphics hardware displaying a degraded display. This is multitasking, and this is Amiga - you cannot do anything against this without violating multitasking rules. Of course it is possible to avoid this problem by hardware hitting, but this works against the concept of multitasking and will therefore not be done. If you need this though, try another degrading tool.
  - The sprite DMA shows weird patterns, though you infiltrated the program (which should turn off the sprites, too). If this happens, one of the currently open windows became active. All windows have an own pointer image, which becomes active on window activation. You may have installed an AutoPointer/SunMouse utility which activated a window after the sprites have already been turned off. Avoid this by disabling the appropriate application (the window-activating one). Another possibility is, that the infiltrated program opened a window which became active. You can't easily work around the latter problem.
  - The program you want to infiltrate is self-detaching and the settings of RunLameInfiltrator are only valid until the program has detached itself. If the program has an option to turn off detaching, then use this option at starting the program to prevent detach. Otherwise, bad luck. Some self-detaching programs even refuse to work after having been infiltrated. Other programs work totally okay after being infiltrated. The behaviour depends on the detaching mechanism used by the program.
  - You have Enforcer running and your system always hangs when you start an infiltrated program. See the Known Bugs section for an explanation for this phenomenon. Quit Enforcer before you start an infiltrated program.
  - You tried everything, but you didn't manage to run a lame program properly. RunLameInfiltrator is not able to fix very lame programming bugs, so you can assume, the programmer of the program is lame himself, or didn't have enough time to do a proper code. But nobody is perfect :-)  
Your last chance might be to try other degraders available in the Amiga
-

world.

## 1.15 What or who is lame?

In terms of Amiga scene users the word lame is used upon persons and objects (programs, tunes, pictures,...) and can mean very different things.

Lame persons are people who have their Amiga for a certain time and cannot be called beginners any more. A lamer has not learned anything while the whole time he/she uses his/her Amiga. The thing a lamer can do best is using autobooting software.

Secondly some people who get a very high rate of new software call other users who don't have access to the latest programs lamers, too. But in this case the lamer is the one who is proud on getting the latest software (yOoOo ya'r so 3lIté :-), in my eyes.

Third possible approach for defining a lamer is the person's coding abilities. Some people name others lamers, due to the fact that their programming abilities are more advanced than the ones of the others. Very often do programmers call others lamers because they think they are themselves very kewl and unbeatably the best. In most cases the code of such superc00ld00dz turned out to crash on non-Amiga500-\$C00000 models. Decide yourself who is lame here.

The latter definition leads us to the subject of lame productions, of maybe lame people. In the case of RunLameInfiltrator only lame programs are of an interest, other kinds of lame objects are not considered here.

Some people call it lame to use system functions in programs and therefore program everything themselves. The makers of trackloading demos seem to think in this track. Poorly the people who think to be able to make it always better than the operating system fail to test their code thoroughly and often their programs don't work on advanced hardware or operating system improvements. Other people like their demos to be run out of the Amiga's multitasking environment and turn off the multitasking. Some of them are so smart to think after a single Disable() the multitasking is shut down properly and they might do what they want (fixed address code, assuming screenmode/resolution is set properly, assuming there are no caches, or VBR). The demo may run, and if, it might never return to multitasking. Others return to multitasking, which is totally trashed then. This seems to be worth called lame, doesn't it?

Most kinds of demo effects or game effects cannot be programmed satisfactory using the system functions, so shutting down the system is not a bad idea, but shutting down a multitasking system requires a high level of understanding how multitasking works on an Amiga. So if people think they can code like kings, make a lot demos that crash all sorts of Amigas, they ridicule themselves and just do show that they try to jump around before they have learned to walk stably. In my eyes this is really lame and childish.

It is simply annoying to have to reboot the machine, just because having watched a demo or having played a game. This must seem very ridiculous to an IBM-compatible PC user, whom you try to impress by showing demos that always force your machine down to it's knees. Seriously, we all laugh about

---



the lame PCs and PC programs, but most of the brill' Amiga demos are less compatible.

So if MSDOS/WINDOWS PCs are lame (and they are), don't be as same as lame!

Think twice, what is the difference between Amigas and PCs!

## 1.16 What a TLA is

A TLA is a TLA for the word Three Letter Acronym.

## 1.17 What a CPU is

CPU is a TLA for Central Processing Unit.

The CPU is the hardware which reads and executes programs instruction by instruction.

## 1.18 What the VBR is

VBR is a TLA for Vector Base Register.

The VBR points to the memory address, which is the base for several vector offsets (f.i. interrupt vector offsets).

## 1.19 What a MMU is

MMU is a TLA for Memory Management Unit.

A MMU is a very powerful memory mapping hardware. It allows f.i. mapping addresses generated by the CPU to another hardware address. If no hardware address can be associated, an interrupt can be triggered. This both allows implementation of virtual memory.

## 1.20 What OCS is

AGA is a TLA for Advanced Graphics Architecture. The graphics abilities available under AGA are a superset of the graphics abilities of OCS and ECS.

## 1.21 What ECS is

ECS is a TLA for Enhanced Chip Set. The graphics abilities available under ECS are a superset of the graphics abilities of OCS.

---

## 1.22 What AGA is

AGA is a TLA for Advanced Graphics Architecture. The graphics abilities available under AGA are a superset of the graphics abilities of OCS and ECS.

## 1.23 What a GUI is

GUI is a TLA for Graphical User Interface.

A GUI should even unexperienced users enable to access a utility without being confused by a variety of new objects and operation methods. Standardized GUIs like supported by gadtools.library give programs based on it a similar look. All operations are accessed by the same objects and their behaviour is standardized (objects are checkboxes, cycle gadgets, listviews, pull down menus, drop-in-icons, drop-in-windows,...).

## 1.24 What MUI is

MUI is a TLA for Magic User Interface.

MUI is a very powerful class oriented GUI. MUI is © by Stefan Stuntz.

## 1.25 What an API is

API is a TLA for Application Icon.

An API is an icon into which you can drop other icons. Normally application icons are located on the Workbench and icons of any file can be dropped into it.

## 1.26 What is Enforcer

Enforcer is a tool written by Michael Sinz. It helps developers to find some kind of bugs. The symptoms of such bugs are called Enforcer hits.

## 1.27 Other Degraders available

Other system degrading software

By time several other degraders with different facilities are available in the Amiga world. I will not give a rating for them, but you should beware which you use, some are in fact of minor quality and might crash your system (and crashing the system is indeed a poor idea while you are write-accessing any form of disk).

---

All degraders known to me are listed below and are available from AmiNet (lowest required Kickstart release in []s):

Degrader	V1.30	[V1.2 - V3.0, depends on used features]
EmbedderII	V1.0	[V2.04 - V3.0, depends on used features]
KillAGA	V2.0	[V3.0]
RunIt	V1.4	[V1.2]
RunShit	V1.06	[V1.2]
WBKillAGA	V1.0	[V3.0] (needs KillAGA)

I doubt that you will have more luck using those programs, but if so, contact me and let me know.

Ah, and let me point out, that Degrader was the first degrader published. Degrader just lacks the capability of restoring the system to it's previous state, so RunLameInfiltrator was born. The concept of RunLameInfiltrator is original and not yet imitated by other software.

Nevertheless RunLame is the only Degrader available to me, that was able to switch the display into PAL mode when a screenmode of a graphics board was active (tested with a Merlin graphics board © by X-Pert). All other Degraders just left the graphics board's screenmode active.

## 1.28 Beta testing of RunLameInfiltrator

How RunLameInfiltrator has been  $\beta$ -tested

At first RunLameInfiltrator is tested by the author who has access to several Amiga computer models (his own and computers of friends of him). So RunLameInfiltrator is instantly tested on the following configurations:

- Amiga 2000 (68000, ECS, Kickstart 2.04, Workbench 2.1)
- Amiga 3000 (68030, ECS, Kickstart 2.04, Workbench 2.0)
- Amiga 3000 (68030, ECS, latest available Kickstart and Workbench)

When RunLameInfiltrator works o.k. on above models, it is sent to several people ( $\beta$ -testers) on the internet computer network. Because the author has no quick access to an AGA model, he relies on bug reports from his  $\beta$ -testers. Many of the  $\beta$ -testers have an AGA machine, so RunLameInfiltrator will be tested under this environment.

Because the author often receives AGA bug reports some days after spreading the latest  $\beta$ -versions of RunLameInfiltrator, the turn-around times for bug fixing are enormous (no direct feedback if the bug is really gone after having modified RunLameInfiltrator). So after adding a new feature, the author has to wait for rather a long time after spreading the  $\beta$ -version to ensure no bugs were embedded. After this certain time interval, RunLameInfiltrator theoretically is ready to go to a public audience.

So, when the  $\beta$ -testers are lazy and don't send a bug report when they find a bug, a buggy version will be spread to a wide audience. The bugs in it will only affect AGA machines though. This problem has shown up on version 1.6 which used to crash on AGA models, but the author didn't recognize the bug, because RunLameInfiltrator worked very normal on the above mentioned machines.

The author asks the AGA users of RunLameInfiltrator for their

---

understanding and forgiving if a version of RunLameInfiltrator doesn't work on their configuration. All known bugs will of course be fixed as fast as possible. If someone desires to send me a bug report or wants to become  $\beta$ -tester, feel free to contact the author.

## 1.29 Known bugs in RunLame

### Known bugs

- If Enforcer is running while RunLameInfiltrator is trying to move the VBR to \$0.1, the machine hangs. Of course this should not happen - I guess this happens because some system configurations redirect the Enforcer output to a window, and when RunLameInfiltrator copies the \$0-\$400 page this causes a whole bunch of Enforcer hits. I guess this is a bit too much for the Enforcer window, but I am not sure about this. Normally there should be reported several accesses to the addresses \$0-\$400, because RunLameInfiltrator MUST save vectors that it is going to change, in order to restore them when leaving (should cause some more Enforcer hits). I can't work around these Enforcer hits, RunLameInfiltrator MUST access these vectors on low level, sorry if this is annoying you.

## 1.30 Future plans with RunLame

### Future plans

- Configurable RunLameInfiltrator better fitting each lame program's needs.
- Emulation of fake fast RAM at \$C00000 on systems which have a MMU installed. For this purpose I rely on information about MMU programming, which is not yet available to me. If you want to help me, send me information.
- NoFastMem, 1mb/.5mb chip mem memory reduction features.
- I still need a native english speaker to correct my lousy english in the documentations for RunLameInfiltrator.
- Do you have any suggestions (to improve RunLameInfiltrator or the documentation)?

## 1.31 Features that won't be implemented in RunLame

### Features not to come

- RunLameInfiltrator which is able to prepare a meal, fly and wash your clothes :-)

## 1.32 RunLameInfiltrator history

### History

Version 1.3

- First public release.

Version 1.6

- Second public release.
- WARNING!! This release crashes on AGA machines.  
Read about beta-testing to understand why.

Version 1.9

- Third public release.

Version 1.10

- Fourth public release.

## 1.33 First public release

V1.0β: •Not released.

V1.1β: •Not released.

- Due to improved system conformity the little PAL screen has a name now.

V1.2β: •Not released.

- The PAL screen name is now identical for all forthcoming versions, so it is not needed to update any databases where the screen name has been registered (NewMode, Domino promotion system, ...).
- Like RunLame now clears caches after having moved the VBR, I strongly recommend NOT TO USE earlier versions of RunLameInfiltrator.
- Funnier version string, making all authors of lame programs blush.

V1.3: •Released to the public.

- The internals of RunLameInfiltrator have been extensively improved, so I strongly recommend NOT TO USE earlier versions of RunLameInfiltrator, they may not work properly on some Amiga models!!
  - Bug fixed: The PAL screen tag list was not terminated with TAG\_DONE. I wonder why this worked:-)
-

- The PAL screen now is in PAL:Low Res (non interlaced).
- The gb\_MemType is set to BANDWIDTH\_1X now on AGA machines. This has been suggested by ID of Sphinx. The tip came originally from Randell Jesup.
- Changing and restoring gb\_MemType is now safer (no task switching in meanwhile). When another task changes gb\_MemType after we did, the old value is not going to be restored, so we don't kill the environment of the other task (BTW, nobody had problems, but I think it's just smarter this way).
- gb\_MemType is only changed during the opening of the little PAL screen. Both happens while only interrupts are allowed but no taskswitching, thus reducing the chance another task can open a screen with the wrong bandwidth. Furthermore all screens opened by the program that has been infiltrated will have the former system bandwidth, but these programs should be aware of that when they are systemfriendly (I know only 2 demos which do open a screen using `OpenScreen(TagList())`).
- Heavy internal changes to the processor degrading/restoring code. It is now much more safer and no inconsistencies in multitasking with other tasks changing the same registers should occur (though nobody reported any problems).

## 1.34 Second public release

V1.4ß: •Not released.

- Bug fixed: By accident the PAL screen has had no name in V1.3.
- The PAL screen name is now identical for all forthcoming versions, so it is not needed to update any databases where the screen name has been registered (NewMode, Domino promotion system, ...). This should have been happened since V1.2ß, dunno why not.
- Bug fixed: Due to usage of the function `CacheControl()` RunLameInfiltrator crashed under lower Exec versions than V36. I didn't have a V34 Kickstart available, so I didn't see that this function is not available under V34- (and the autodocs don't point out that it's a V36+ function).
- Access to `graphics.library` base smarter now.

V1.5ß: •Not released.

- The sprite resolution for the PAL screen is explicitly reset to low resolution. This might not be needed, because the sprite resolution for a that PAL screen should be lores anyway. In case of false promotion of that PAL screen, at least the sprites will be set to lores now.

V1.6: •Second public release.

---

- The sprite DMA of the mouse pointer will point to a NULL graphic, thus reducing the chance of sprite flicker if the lame program does just disable the sprite DMA in the wrong moment. This does only work if the user doesn't activate any window in the execution phase, because all windows keep their pointer for greater convenience.
- Checking if the PAL screen opened really in PAL lores. Results in an error message if a promotor did change the screen mode (only if you have OS2+ installed, but think twice, have you ever seen a promotion utility for OS1.3-?)

## 1.35 Third public release

V1.7β: •Not released.

- Bug fixed: The cleared mouse pointer was defined to be 0 pixels wide. Now the width of it is set to 16 pixels.

V1.8: •Not released.

- Bug fixed: When restoring the sprite resolution, a wrong address register has been referenced. This might have caused the system crashes on AGA models.

V1.9: •Third public release.

- Minor internal improvements.
- Bug fixed: all previous versions did crash when the infiltrated program was started from the Workbench. Now starting from Workbench works safe, but restoring the degraded system components is not done any more. Furthermore a mem leak now is needed if started from Workbench.

## 1.36 Fourth public release

V1.10: •Fourth public release.

- Now all caches are cleared, even if a Kickstart version lower than V36 is installed. If Kickstart V36+ is installed, system functions are used for this, otherwise direct CPU control.

## 1.37 Thanks and greetings

Thanks, greetings

- Thanks, Jesus Christ for being the god who cares about us all.
  - Thanks and love to my girlfriend for being such a patient person, may the
-

- lord bless thee and protect thee.
- Greets to Zaphod Beeblebrox of Hypenosis (letz do the hype :-).
  - HOARS to ALFred, a legend is back home @ IRC, letz have a nice time!
  - U4ia, may the Lord show you, that he'll never forget about you. This is no hype, you got the vibe, man. Good luck for MDMA!
  - Thanx to Delirium for their help. Without you it would have taken me much more longer to implement my first GUI. Keep up good work on DeliTracker!
  - Thanx and greets to the brave beta testers (did I forget someone?): Zap, ALFred, Savage, Dense, Sphinx, Milano, Slammer, Kryss, Andemar, Mnemo, Skull, Ikke, Trooper1, VAG, Mjoelnir, Gio, SteveVai, Starfox, Gimli, KicStart, Tirreg, Xed, Exolon, Martin, Dizolve. (list in no specific order)
  - Greets to all the nice pals on IRC (Netrunner, XS, Zop, Gucky, Outland, Gateway, Janne, Juggi, Advance, Goldrnr, Shocker, Mag, and all I know but forgot now (list in no specific order)
  - Thanx to Jean-Guy Speton and all contributors for the AmigaDemoList, which contains valuable compatibility information about various demos.
  - Last but not least, I send encouragement to all those who made these despicable lame programs, thus forcing me to spend lots of time writing RunLameInfiltrator. Try to do a better job about compatibility, next time. Using RunLameInfiltrator to make work your own written programs should be a shame for you! One time diskmag will contain compatibility ranking lists, I hope.

## 1.38 How to contact the author of RunLame

For bugreports, ideas, suggestions, improvements, questions and all other concerning RunLameInfiltrator meet me on IRC on channel #amiga or #amigager.

---