RunLame

Bilbo the first

COLLABORATORS			
	TITLE : RunLame		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Bilbo the first	October 9, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

RunLame

1.1 RunLame documentation

RunLame

Version 1.32

Table of contents

Prologue

1.2 Disclaimer

LICENSE and TERMS for software originating from 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 posession 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 accompagnying this package may NOT be reproduced or transferred outside a one-machine-environment without all these accompagnying 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.

1.3 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 .

1.4 RunLame introduction

The aim of the game

Do you run your system using another screenmode than PAL or NTSC, for example EURO72 or an AGA specific screenmode? 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 socalled 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.l?

lame programming is the disease, RunLame is the cure!

RunLame is a Kick2.04+ tool, it will NOT work under Kick1.0/1.1/1.2/1.3/1.4! RunLame is 100% reentrant Assembler code, so you can set the pure flag and make it resident. RunLame is startable from Workbench.

RunLame considers some adjustments and a command line and executes the given command line, just after having degraded the display (optional) and the CPU (optional). When the command line has been executed the display and CPU state is restored to the same as before executing the command line.

Summarizing we can say RunLame is a multitasking friendly temporary system degrader which 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 RunLame has been used, they are on a higher lameness-level, which can’t be fixed by RunLame (this is definitely not a bug of RunLame).

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

1.5 RunLame features

Features

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

- Optionally disabling of each processor cache, burst or copyback mode. {68020, 68030[EC], 68040[EC]}

- Optionally moving VBR to location $0 or to a FAST memory location. {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. {ECS, AGA}

- Degrading sprite resolution to simulate OCS if display degrading is activated. {AGA}

- Turning off sprites in a system friendly manner if display degrading is activated. {OCS, ECS, AGA}

- Flexible and highly configurable fix for a wide variety of lame programs.

- Works under Kickstarts 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.

- Impressive user friendlyness due to various possibilities of usage:

  - A CLI/Shell interface with online help and OS2 conform parsing for quick and effective usage directly, or indirectly (fi. aliases, shell scripts, DirectoryOpus calls).

  - Startable from Workbench, additionally a whole bunch of tooltypes is supported.

  - A GUI in gadtools style of OS2+ for maximum convenience. Supports pull down menus, keyboard shortcuts, ASL filerequester (which of course is the most often patched system part :-), an application window wich can be zoomed.

  - An application icon which can be placed on the Workbench for intuitive and specialized usage.

- Detailed messages inform the user in failure situations. Messages are automatically directed to Shell window or to requesters, depending on operation mode.

- Output of the commandline is directed to Shell window or to an automagically opening window , depending on operation mode.

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

1.6 RunLame disadvantages

The bad news

- RunLame cannot simulate a Workbench startup for lame programs. The execution of programs is equivalent to a call from a shell. Therefore some programs might not make use of their tooltypes. But be serious, have you ever seen a program which needs Workbench startup and RunLame support?

- The GUI is not font adaptive, nor is it localizable.

- Though the code worked very well during the beta testing phase, there might be nasty bugs.

- RunLame can't be used to run non-DOS trackloading demos, for I don't want to support this kind of loaders. I usally like to keep good demos on my harddisk, which is not directly possible with trackmos. So I don't keep lame coded trackmos.

- The icon of RunLame is not yet very original. If you would like to draw a better one, which I shall distribute in the next release, contact me.

- 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.

1.7 Installing RunLame

Installing RunLame

Installing RunLame 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:

- RunLame
- RunLame.info

That's it. You can now start RunLame from shell or from Workbench. For special operation topic refer the operation description .

1.8 RunLame operation description

How to use RunLame

Usage from Workbench

Usage from CLI

General notes

Both, Workbench tooltypes and CLI AmigaDOS template control the configuration of RunLame. Those keywords are called switches to simplify the terminology.

Of course some of the switches exclude each other, for example it is not possible to use SCREEN and NOSCREEN together, or DATABURST and NOBURST . Wrong usage of switches results in the appropriate error message. Though

COPYBACK only works on 68040 processors, no errormessage is prompted when this switch is used on another processor because this makes scriptfiles easier system portable. Same is to be said about EXTERNALCACHE when there is no external cache available, as same applies to VBRTOFAST when the processor has no vector base register (68010+ have a VBR).

1.9 Using RunLame from the Workbench

General notes on usage from Workbench

To use RunLame from Workbench, simply double click it’s icon. Depending on the supplied tooltypes in the icon, RunLame is going to be configured.

Furthermore it’s possible to supply a COMMANDLINE to RunLame without using the apropriate tooltype. If you want to do this, click first RunLame’s icon, and then as many icons as u desire in the order you want to. The last needed icon is to be double clicked.

If you want adapt RunLame’s configuration to your own needs, read the section about configuration topics.

Workbench users are advised to read about the possibilities given by the

application icon feature and the path setting .

1.10 RunLame configuration on Workbench using tooltypes

Workbench tooltypes

COMMANDLINE

GFXUSERINTERFACE

APPICON

LOOP

DEFAULTS

SCREEN

VISIBLE

CACHE

INSTCACHE

DATACACHE

BURST

INSTBURST

DATABURST

COPYBACK

EXTERNALCACHE

VBR

APPEND

PATH

1.11 Using RunLame from a shell/CLI

General notes on usage from a shell/CLI

It’s assumed, that shell/CLI users are advanced and more experienced users. If you don’t userstand the explanations in this section, use RunLame together with the Workbench.

You can simply call RunLame using the command RunLame. This opens the GUI and configures RunLame according to it’s defaults. If you don’t need the GUI, call RunLame together with the needed switches and the desired commandline. Since RunLame ignores the Workbench tooltypes when started from a shell/CLI, shell users can only configure RunLame using aliases or script files.

1.12 RunLame configuration on shell

AmigaDOS template

```
RunLame H    = HELP/S,
  GUI  = GFXUSERINTERFACE/S,
  API  = APPICON/S,
  L    = LOOP/S,
  D    = DEFAULTS/S, ND   = NODEFAULTS/S,
  S    = SCREEN
```

```
              /S, NS   =
               NOSCREEN
              /S,
V    =
               VISIBLE
              /S, NV   =
               INVISIBLE
              /S,
C    =
               CACHE
              /S, NC   =
               NOCACHE
              /S,
IC   =
               INSTCACHE
              /S, NIC  =
               NOINSTCACHE
              /S,
DC   =
               DATACACHE
              /S, NDC  =
               NODATACACHE
              /S,
B    =
               BURST
              /S, NB   =
               NOBURST
              /S,
IB   =
               INSTBURST
              /S, NIB  =
               NOINSTBURST
              /S,
DB   =
               DATABURST
              /S, NDB  =
               NODATABURST
              /S,
CB   =
               COPYBACK
              /S, NCB  =
               NOCOPYBACK
              /S,
EC   =
               EXTERNALCACHE
              /S, NEC  =
               NOEXTERNALCACHE
              /S,
VBR  =
               VBRTOFAST
              /S, NVBR =
               VBRTO0
              /S,
APN  =
               APPEND
              /S, NAPN =
               NOAPPEND
```

```
                /S,
  P    =
                 PATH
                /K,
  CL   =
                 COMMANDLINE
                /F cmd
```

1.13 HELP, H

```
CLI options                     Workbench tooltypes

HELP, H       not supported
```

Explanation:

Display a short summary of valid options in the current CLI window. If the window is too small to completely hold this information, you may redirect the output to a file (RunLame HELP >T:RunLame.olh), which can be read using a text editor.

1.14 GFXUSERINTERFACE, GUI

```
                CLI options                     Workbench tooltypes

GFXUSERINTERFACE | GUI    GFXUSERINTERFACE | GUI  = USE
```

Explanation:

The
GUI
is always invoked when the
COMMANDLINE
parameter is omissed. Use
the GUI option to force RunLame to open the GUI, even if
COMMANDLINE
has
been specified, whose argument in this case will appear in the command line string gadget. The GUI is going to appear centered to the mouse pointer on the default public screen, I just liked the behaviour of the ReqChange redirected requesters. The GUI is mainly divided in two parts: the menu, and the window. Both, menu and window gadgets correspond directly to each other. Both, menu items and gadgets support shortcuts. The shortcuts of the menu are the same as the shortcuts of the gadgets, but they need the left Amiga key qualifier pressed additionally. So both, 'I'(or i') and LAMIGA+'I'(or 'i') result in the same action as if the Instruction Cache gadget or menu item would have been clicked/selected. The shortcuts are not listed in this documentation for the user can directly determine them from the GUI. The menu shortcuts are shown right to the appropriate menu item if

it supports shortcuts. The gadget shortcuts are case insensitive (like the menu shortcuts). The underlined character of the gadget description text is the corresponing shortcut key, not to be pressed with additional qualifiers except for optionally SHIFT. If the launched command line would produce any outputs to the CLI window, this output is redirected to a automatically on the default public screen opening CON: window, which can be closed asynchroneously (meaning that RunLame is not running any more, but the window remains on the default public screen and can be closed when not needed any more). In addition to be able to input the command line using the string gadget or the filerequester, it is possible to drop (multiselected) icons from the Workbench into the GUI window. The name(s) of the corresponding object(s) appear(s) in the command line string gadget added to the end of the previous contents (assuming
NOAPPEND
has not been
supplied). Using the
PATH
parameter makes it possible to set an initial
drawer for the file requester (normally the file requester uses the current directory as initial drawer)

The pull down menu (shortcuts not documented):

```
  main menu strip:         sub menu strips:

 +======================+ +=============+ +=============+ +=============+
 |RunLame    | |VBR location | |Caches | |Burst modes  |
 +======================+ +=============+ +=============+ +=============+
 |
               About
                     | |
               $0
                   | |
               Instruction
                 | |
               Instruction
                 |
 |~~~~~~~~~~~~~~~~~~~~~| |
               FAST memory
                 | |
               Data
                 | |
               Data
                 |
 |VBR location     | +-------------+ |
               External
                 | +-------------+
 |Caches    |     +-------------+
 |Burst modes   |
 |
               Copyback mode
                 |
 |~~~~~~~~~~~~~~~~~~~~~|
 |
               Screen
                   |
 |
```

```
               Visible screen
                 |
|~~~~~~~~~~~~~~~~~~~~~~|
|
               Enter commandline
                 |
|
               Open file requester
                 |
|
               Launch commandline
                 |
|~~~~~~~~~~~~~~~~~~~~~~|
|
               Quit
                     |
+----------------------+


 application window:

      +-----------------------+
      |
               .
                RunLame           |
      +-----------------------+
      |+---------------------+|
      ||
                $0
                VBR location ||
      |+---------------------+|
      ||

                Instruction Cache  ||
      ||

                Data Cache ||
      ||

                External Caches  ||
      |+---------------------+|
      ||

                Instruction Burst  ||
      ||

                Data Burst ||
      |+---------------------+|
      ||

                Copyback   ||
      |+---------------------+|
      ||

                Screen

                Visible||
```

```
       |+---------------------+|
       |


                   |
       |
                 Launch commandline
                   |
       +-----------------------+
```

1.15 About menu item

Explanation:

A short copyright notice is shown in a requester.

1.16 Enter commandline menu item

Explanation:

Activate the commandline string gadget which enables you to specify a
COMMANDLINE
.

1.17 Open file requester

Explanation:

Open an ASL file requester. You then may select a file whose path/name will be appended to the commandline, depending on the
APPEND
switch.

1.18 Launch commandline

Explanation:

This action ignites the execution of the command line. The current configuration of RunLame is active during the execution of the command line defined in the command line string gadget. After execution 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 with active data cache, but an active instruction cache would speed it up, forthermore it won't run in burst mode and with VBR in another location than $0.l, use:

```
 RunLame GUI INSTCACHE NODATACACHE NOBURST INVISIBLE ObsoleteFX
```

You will see the correct setting reflected in the gadgets and the menu items. If you now launch the command line, the following will happen:

```
 assumed system settings before launching:
  data cache active        | instruction cache active
  data burst mode active   | instruction burst mode inactive
  VBR in Fast RAM at position p

 system settings during execution of ObsoleteFX (launched):
  data cache inactive      | instruction cache active
  data burst mode inactive | instruction burst mode inactive
  VBR at $0.l

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

1.19 Quit RunLame

Explanation:

The RunLame task/process will stop existing, all held resources are given back to the system. The looping mode (activated by
LOOP
) is stopped, too.

1.20 APPICON, API

CLI options	Workbench tooltypes
APPICON | API	APPICON | API = USE

Explanation:

The
API
supplies a comfortable method of immediate starting objects ↩
whose
icons are dropped into it. It is possible to multiselect icons, so the command line consists of all objects in the order they have been selected. To quit RunLame, double click the API. When a command line has been supplied at the startup of RunLame, all icons/objects dropped into the API are appended to this command line, then the command line is executed (assuming
NOAPPEND

has not been supplied). When too many icons are dropped into the API , the Workbench screen beeps before command line execution. Output produced by the command line goes to the CLI when RunLame was started not from the Workbench. The icon representing the API is the same as used for RunLame (if this is missing, the system default icon is used) , the name is the same as RunLame has on your disk.

1.21 LOOP, L

```
                CLI options                     Workbench tooltypes

LOOP | L      LOOP | L    = YES
```

Explanation:

The LOOP option can only be used together with API and/or GUI . When you want to start several command lines in a sequence, it is quite unconfortable to always start RunLame for each command line execution. LOOP works around this problem by forcing RunLame to pop up the API and/or GUI again after each command line execution, until it is quitted by double clicking the API or quitting the GUI. When the appropriate input object (API/GUI) pops up again, the initial command line given at startup of RunLame (COMMANDLINE) is always inserted at front of the command line created using the input object. Furthermore all settings of the former command line execution are preserved.

Example:

```
 RunLame APPICON LOOP dir
```

You can now drop icons of Workbech drawers into the application icon to list their contents. The dir command is always assumed to be the first part in the command line. Same applies to startup from Workbench via (multi-)selected icons.

1.22 DEFAULTS, NODEFAULTS, D, ND

```
                CLI options                     Workbench tooltypes
```

```
DEFAULTS | D      DEFAULTS | D    = USE
NODEFAULTS | ND     DEFAULTS | D    = NO
```

Explanation:

If you specify DEFAULTS at startup of RunLame, the following options are assumed:

SCREEN
,
VISIBLE
,
NOCACHE
,
NOINSTCACHE
,
NODATACACHE
,
NOBURST
,

NOINSTBURST
,
NODATABURST
,
NOCOPYBACK
,
NOEXTERNALCACHE
,
VBRTO0
,
APPEND

The DEFAULTS option is always assumed to be set. So if you want to disable all caches, the copyback mode and want to degrade the display, you need not to specify one of these options. Those options are really obsolete, but were implemented for completeness. Furthermore the specification of those options in script files (though leaving them out would have the same effect) increases the readability of the script file because every user not knowing which defaults are set by RunLame can see what really is going to happen. So the following usages of RunLame affect in the same:

```
 RunLame foobar

 RunLame D S NC NB NCB NEC NVBR CL foobar

 RunLame BURST NIC NOEXTERNALCACHE CL=foobar

 RunLame SCREEN NOINSTCACHE NODATACACHE NOINSTBURST NODATABURST NOCOPYBACK
         NOEXTERNALCACHE VBRTO0 DEFAULTS COMMANDLINE=foobar
```

The NODEFAULTS switch forces RunLame to configure itself to fit best the current system settings. So if you specify NODEFAULTS and don’t override any of the settings, RunLame will change nothing in the system while executing the commandline.

Furthermore specification of options overrides the DEFAULTS/NODEFAULTS settings of RunLame. So it’s possible to start programs in your normal environment, with some parameters just changed, not all possible. The parameters changed are set to the same value each time you use this feature, but the rest of the settings are the built in defaults/the current system setting.

Example:

```
 RunLame NODEFAULTS VBRTO0 SCREEN INVISIBLE ObsoleteFX

 assumed system settings before RunLame usage:
  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 active        | instruction cache active
  data burst mode active   | instruction burst mode inactive
  external caches inactive | copyback mode inactive
  VBR at $0.l
  screenmode changed to PAL, screen blanked,
  sprite resolution changed to lores and sprites turned off

 system settings after execution of ObsoleteFX:
  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
```

1.23 SCREEN, NOSCREEN, S, NS

```
                CLI options                     Workbench tooltypes

SCREEN | S      SCREEN | S    = ON
NOSCREEN | NS     SCREEN | S    = OFF
```

Explanation:

Non system conform programmed software might cause problems when started from within another screen mode than PAL. This might effect in weird running bitplanes or anything else, but when it happens, you will become aware of it, believe me. To avoid these annoying problems, the default option called SCREEN opens a 2x2 pixels wide screen with depth 1 (this should eat up not very much memory :-) before starting the problematic program. Due to the existance of several screen mode promotion tools I have to point out that it is not very clever to allow promotion of the screen opened by RunLame. Furthermore some programs use an incorrect method to turn off sprites which results in running pixel patterns (similar to the above decribed bitplane effect). RunLame tries to avoid this problem when the SCREEN option is active. In favour to RunLame’s multitasking abilities,

the sprite DMA is not shut down, but redirected to a NULL sprite (which is the most effective method to turn off sprites). This method cannot avoid, that the pointer is reactived by any window becoming active after turning off the sprites (refer to
Problems - solutions - corner
).

1.24 VISIBLE, INVISIBLE, V, NV

```
                CLI options                      Workbench tooltypes

VISIBLE | V     VISIBLE | V   = YES
INVISIBLE | NV      VISIBLE | V   = NO
```

Explanation:

The VISIBLE option keeps the previous frontmost screen visible, but displayed in PAL. This is handy when the program you start with RunLame outputs to any window opened/opening on the previous frontmost screen. For programs that don't output anything to the previous frontmost screen, use the INVISIBLE option, so the current display contents are hidden. This option only makes sense together with the
SCREEN
option.

1.25 CACHE, NOCACHE, C, NC

```
                CLI options                      Workbench tooltypes

CACHE | C     CACHE | C   = ON
NOCACHE | NC      CACHE | C   = OFF
```

Explanation:

This option controls all processor caches installed in the system. Refer to

INSTCACHE
and
DATACACHE
for more detailed description.

1.26 INSTCACHE, NOINSTCACHE, IC, NIC

```
                CLI options                     Workbench tooltypes

INSTCACHE | IC      INSTCACHE | IC    = ON
NOINSTCACHE | NIC   INSTCACHE | IC    = OFF
```

Explanation:

This option allows you to control the instruction cache of the CPU . Some lame programs use self modifying code which can lead to inconsistency between cache contents and memory contents. The result in most cases is a system crash. In this case turn off the instruction cache.

1.27 DATACACHE, NODATACACHE, DC, NDC

```
                CLI options                     Workbench tooltypes

DATACACHE | DC      DATACACHE | DC    = ON
NODATACACHE | NDC   DATACACHE | DC    = OFF
```

Explanation:

This option allows you to control the data cache of the CPU . Some lame programs modify memory data whilst it is read by a DMA channel, which can lead to inconsistency between cache contents and memory contents. The result in most cases is a weird display, a crash or other anomalies. In this case turn off the data cache.

1.28 BURST, NOBURST, B, NB

```
                CLI options                     Workbench tooltypes

BURST | B     BURST | B   = ON
NOBURST | NB      BURST | B   = OFF
```

Explanation:

This option lets you control the burst modes of the CPU caches all together.

Refer to

INSTBURST and DATABURST for more detailed description.

1.29 INSTBURST, NOINSTBURST, IB, NIB

```
                CLI options                     Workbench tooltypes

INSTBURST | IB      INSTBURST | IB    = ON
NOINSTBURST | NIB   INSTBURST | IB    = OFF
```

Explanation:

This option allows you to control the burst mode of the instruction cache of the CPU . Disabling might fix strange behaviour of programs.

1.30 DATABURST, NODATABURST, DB, NDB

```
                CLI options                     Workbench tooltypes

DATABURST | DB      DATABURST | DB    = ON
NODATABURST | NDB   DATABURST | DB    = OFF
```

Explanation:

This option allows you to control the burst mode of the data cache of the CPU . Disabling might fix strange behaviour of programs.

1.31 COPYBACK, NOCOPYBACK, CB, NCB

```
                CLI options                     Workbench tooltypes

COPYBACK | CB     COPYBACK | CB   = YES
NOCOPYBACK | NCB    COPYBACK | CB   = NO
```

Explanation:

If none of the other option helps improving the behaviour of a lame program, try changing the state of the copyback mode of the CPU .

1.32 EXTERNALCACHE, NOEXTERNALCACHE, EC, NEC

CLI options	Workbench tooltypes
EXTERNALCACHE | EC	EXTERNALCACHE | EC = ON
NOEXTERNALCACHE | NEC	EXTERNALCACHE | EC = OFF

Explanation:

This option should give you control over externally installed caches. Due to lack of such hardware, this could not be tested. Furthermore it seems not possible to check for existence of such caches in a system friendly manner. Therefore this option is not accessible.

1.33 VBRTOFAST, VBRTO0, VBRTO, VBR, NVBR

CLI options	Workbench tooltypes
VBRTOFAST | VBR	VBRTO | VBR = FAST
VBRTO0 | NVBR	VBRTO | VBR = 0

Explanation:

This option gives you control over the VBR of the CPU . Most of old programs using interrupts don’t consider the fact, that 68010+ CPUs have a VBR. Since Kickstart V36 the VBR is not granted to point to address $0 (which is located in CHIP memory). If such a lame program accesses CPU vectors which are offsets to the VBR, and the VBR doesn’t point to $0, the CPU jumps to an undefined memory location. Most of the crashes can be fixed by moving the VBR to location $0.

1.34 APPEND, NOAPPEND, APN, NAPN

```
                CLI options                        Workbench tooltypes

APPEND | APN      APPEND | APN    = YES
NOAPPEND | NAPN     APPEND | APN    = NO
```

Explanation:

The APPEND option is set by default, but can be overriden by NOAPPEND. APPEND is responsible for the appending behaviour of the file requester (or the application window feature and the APPICON feature). If you specify NOAPPEND, the command line will always be overwritten by any new given filenames. This works for the application icon, the application window feature and the file requester. You see, when you use APPICON and NOAPPEND together, it's senseless to supply a startup command line, because it's going to be overwritten by the objects you drop into the application icon.

1.35 PATH, P

```
                CLI options                        Workbench tooltypes

PATH | P    = path  PATH | P    = path
```

Explanation:

Using this option you may specify a default startup path for the file requester of the GUI. This is useful if the program you want to execute is not located in the current directory.

1.36 COMMANDLINE, CL

```
CLI options                        Workbench tooltypes

COMMANDLINE | CL  = cmd COMMANDLINE | CL  = cmd
```

Explanation:

The command line may consist of same complex format as could be used in the AmigaDOS shell. The only thing you must pay attention to is that the COMMANDLINE option is always specified as the last one in the whole parameter line, or if not explicitely specified, the name of the program that shall be executed must be the last word in the parameter line. In-/Output redirection is possible.

Examples of valid usage:

```
 RunLame NV VirtualIntelligence                  ;a demo

 RunLame NS C B CB EC VBR CL list >RAM:foo SYS:  ;senseless, but works

 RunLame NOSCREEN CACHE B CL= C:dir RAM:         ;senseless, but works

 RunLame SCREEN VISIBLE COMMANDLINE XCopyPro     ;a copy program

 RunLame NS IC DC IB DB NCB COMMANDLINE=Resident ;works
                                             ;(Resident is resident here :-)
```

NOTE: when you use filenames containing blank characters, you have to enclose them into triple quotation marks:

```
 for AmigaShell from C=: RunLame """foo bar"""
 for CShell (Unix port): RunLame \"\"\"foo bar\"\"\"
```

Otherwise the shell will strip away the quotation marks, and RunLame will never see them, thus executing the command line without quotation marks, which of course will interpret the blank characters as name delimiters.

1.37 Configuring RunLame

Configuring RunLame

Configuring for shell:

At first the configuring procedure for shell users is going to be described. As you know you have the possibility to use the built in defaults of RunLame or the current system settings for some of the configuration parameters (VBR , CACHE ,...). Those can be overridden by further specified options and parameters. So when you want always to use another configuration than the defaults, just make an alias for the desired configuration.

Example:

```
 Alias RunLame 'RunLame NODEFAULTS SCREEN INVISBLE VBR LOOP API'
```

So when you now type RunLame, it always opens the API, it loops and uses the current systems setting except for the display which is degraded and black, and the VBR is always moved to $0.l.

Configuring for Workbench:

Workbench users just can do the same, they need only to specify the appropriate tooltypes for the icon.

Now you can experiment a bit (follow the instructions below):

- MakeLink FROM RAM:DirLame TO RunLame
- Copy RunLame.info (which came with this distribution) TO RAM:DirLame.info
- Edit the tooltypes in this way:

```
 COMMANDLINE=dir
 APPICON=USE
 LOOP=YES
 DEFAULTS=NO
```

(the other tooltypes keep the way they are)

- Now double click DirLame.

Now an application icon named DirLame should appear on the Workbench. Drop some drawer icons into this and watch what's happening. If you need such gimmicks very often, you may draw a special icon for this purpose.

Using RunLame from Workbench often shows up the problem, that the default path of the filerequester (of the GUI) is pointing to the directory where RunLame is located. This is annoying in the most cases. So if you have a special directory from where you are used to start the lame programs, set the PATH tooltype to this directory.

Configuring for Directory Opus:

Directory Opus users may want to install RunLame as a button, which starts all highlighted files in the file listview.

Enter the configuration module and the Buttons section. Press one of the still unused buttons. Enter 'RunLame' for Name string gadget. Make a New entry which is AmigaDOS. Enter '[path]RunLame [options

] {f}' in the entry string gadget (replace [path] by the path where RunLame is located). Enter Stack size '4000', Priority '0', Close delay '2'. 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.

The new RunLame button is now fully functional. Highlight a lame file and press the new button. Voilà.

1.38 RunLame `Problems - solutions - corner'

Problems - solutions - corner

- If you got this error message:

```
          Screenmode degrading failed!
   Forbid all promotion utilities to promote:
    "RunLame (© by Bilbo 1st of Hypenosis)"
```

  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 "RunLame (© by Bilbo 1st of Hypenosis)".

- The sprite DMA shows weird patterns, though you made RunLame degrade the display (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 apropriate application (the window-activating one). Another possibility is, that the launched program opened a window which became active. You can't easily work around the latter problem.

- The program you want to start is self-detaching and the settings of RunLame 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.

- You have Enforcer running and your system always hangs when you start an Program using the VBRTOFAST|VBRTO0 option. See the Knows Bugs section for an explanantion for this phenomenon. Quit Enforcer before you use the VBRTOFAST|VBRTO0 option.

- When you use filenames containing blank characters, you have to enclose them into triple quotation marks:
  ```
   for  AmigaShell  from  C=:  RunLame """foo bar"""
   for  CShell (Unix port):  RunLame \"\"\"foo bar\"\"\"
  ```
  Otherwise the shell will strip away the quotation marks, and RunLame will never see them, thus sending the command line without quotation marks to System(), wich of course will think that the blank characters are name delimiters. Let me point out that this is NOT a bug of RunLame, learn to live with it.

- When starting RunLame from CShell, and you ask for the error cause using the why command, you will always get the message that no error occured, even if really an error occured. This is NOT a bug of RunLame, but of the CShell V5.31-. If you rely on those error codes, use the AmigaShell of C= instead of the CShell.

- You start RunLame with the intention to display the CLI template using RunLame ?. The template is displayed, but the shell does not allow any CLI input. Instead of waiting for input, RunLame immediately opens the GUI. It is probable that you are using CShell V5.20+ and RunLame resides in the script path defined in the _path variable of the shell. This is NOT a bug of RunLame. I reported this problem to the current developer of CShell, so this might be fixed in future. In the meantime fix the problem by placing RunLame somewhere in your DOS path wich is defined by the AmigaDOS command path (fi.: Path Work:BestToolEver ADD; Rename RunLame Work:BestToolEver/RunLame) or specify the full path of RunLame's location in the filesystem each time you want to use it (this might happen in an alias like this: alias runlame Work:BestToolEver/RunLame).

- You tried everything, but you didn't manage to run a lame program properly. RunLame 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. 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]
RunIt	V1.4	[V1.2]
RunShit	V1.03	[V1.2]
KillAGA	V2.0	[V3.0]
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.

1.39 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 RunLame 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!

1.40 What a TLA is

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

1.41 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.

1.42 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).

1.43 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.

1.44 What OCS is

OCS is a TLA for Old Chip Set. The graphics abilities available under OCS are the lowest possible on any Amiga Model.

1.45 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 .

1.46 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 .

1.47 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,...).

1.48 What MUI is

MUI is a TLA for Magic User Interface.

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

1.49 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.

1.50 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.

1.51 Beta testing of RunLame

How RunLame has been ß-tested

At first RunLame is tested by the author who has access to several Amiga computer models (his own and computers of friends of him). So RunLame 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 RunLame works o.k. on above models, it is sent to several people (ß-testers) on the internet computer network. Because the author has no quick access to an AGA model, he relies on bug reports from his ß-testers. Many of the ß-testers have an AGA machine, so RunLame will be tested under this environment.

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

So, when the ß-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 1.28 which used to crash on AGA models, but the author didn't recognize the bug, because RunLame worked very normal on the above mentioned machines.

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

1.52 Known bugs in RunLame

Known bugs

- If Enforcer is running while RunLame is trying to move the VBR to $0.l, 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 RunLame 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 RunLame 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, RunLame MUST access these vectors on low level, sorry if this is annoying you.

1.53 Future plans with RunLame

Future plans

- 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.

- Gio suggested that a config file parameter would be handy. I will do this someday (will I? :-)

- I would like to distribute a really funny icon together with RunLame. So if you want to draw one, feel free to send it to me.

- I still need a native english speaker to correct my lousy english in the documentations for RunLame.

- Shall RunLame's window become font adaptive in future?

- Does anybody need an ARexx port for RunLame?

- Is it useful to localize RunLame?

- Do you have any suggestions (to improve RunLame or the documentation)?

1.54 Features that won't be implemented in RunLame

Features not to come

- Someone asked for a bootblock intro removal feature. This has in no way anything to do with system degradation, so why the heck should it make it's way into a degrader?

- MUI instead of gadtools GUI . RunLame is intended to be a short program, to allow selfmade intro compilation disks without eating up too much disk space. MUI itself is a disk space killer, understand?

1.55 RunLame history

History

Version 1.1

- Public release named 'RunPal'. No features except for screen degrading.

Version 1.25

- First public release named 'RunLame'.

Version 1.28

```
                  ·Second public release.

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


                Version 1.32
                   ·Third public release.
```

1.56 First public release

```
V1.2ß:  ·Not released.

        ·Corrected the version string to be displayed OS3+ conform using the
         CLI "Version" command.

V1.3ß:  ·Not released.

        ·Disables caches, burst modes and copyback mode now.

V1.4ß:  ·Not released.

        ·Added the template switches for screen, caches, burst, copyback and
         VBR moving (VBR moving is not yet implemented).

        ·Online help available now via HELP.

V1.5ß:  ·Not released.

        ·RunPal  is  renamed  to  RunLame due to many enhancements that were
         added to the program, which now doesn’t only open a PAL screen, but
         is able to run many more lame programmed software.

        ·VBR is only moved  when  nescessary,  it  is  enclosed in Disable()
         Enable(),   Caches   will   be  flushed  and  overwritten  data  is
         saved before moving and restored after restoring VBR.

V1.6ß:  ·Not released.

        ·The screen has been minimized and can be in-/visible now.

  ·RunLameInfiltrator is now distributed with the package.

  ·FileLinker and FileUnlinker are now distributed with the package.

V1.7ß:  ·Not released.

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

V1.8ß:  ·Not released.
```

·Graphical user interface added.

V1.9ß: ·Not released.

·Performed some improvements of GUI.

·When having used GUI all possible CLI window output of the commandline executed is redirected to an automatically opening CON: window, which can be closed asynchronously to termination of RunLame.

V1.10ß: ·Not released.

·Added file requester gadget and ASL file requester to GUI.

·When a further window (any requester for example) is opened by the GUI, the menu is disabled totally.

·It is now impossible to launch an empty commandline from within GUI.

V1.11ß: ·Not released.

·When a further window (any requester for example) is opened by the GUI, the gadgets are all disabled (except for the GENERIC_KIND).

·Fixed a spelling error: ‘modi’ -> ‘modes’

V1.12ß: ·Not released.

·Window now has a zoom gadget and is aware of resizing (DragIt), try to resize the window with a resizing patch program, it will allways snap back to it’s old size. Nobody should try to resize a window without sizing gadget, Bilbo’s windows will strike back :-)

·The settings can be chosen now. You may use the default settings of RunLame, or you can force RunLame to configure to the current system settings.

·The above mentioned settings can be overridden by using options.

·The resulting configuration of RunLame is reflected by the GUI now.

V1.13ß: ·Not released.

·The RunLame GUI now is an application window when the Workbench is opened, hence (even multiselected) icons can be dropped in to fill the commandline string gadget.

·RunLame may now be started from Workbench. The icon may contain various tool types.

V1.14ß: ·Not released.

·If RunLame has to inform the user of an error, this will happen depending on where it has been started from:
Has RunLame been started from the CLI, all error messages go to the

default output of RunLame, hence it can be redirected (needed for script files, maybe). Has RunLame been started from the Workbench, all error messages appear in requesters, providing full comfort.

·Dropping icons into RunLame's window now adds the path/name of the appropriate object to the commandline string gadget (gadget is not cleared, as in previous version).

·Usage of the file requester now adds the path/name of the appropriate object to the commandline string gadget (gadget is not cleared, as in previous version).

·When launched from Workbench, all multiselected icons' (objects) path/name will appear added to the commandline string gadget.

V1.15ß: ·Not released.

·Bug fixed: Previous versions did try to get the path/name of RunLame into a 30 bytes buffer. When both together was bigger, the starting from Workbench would fail (only V1.14). The buffer is now 128 bytes bigger, and the path is not retrieved any more when started from Workbench, in no case the path will be prompted. Hoooray, this was the first bug found in RunLame!

V1.16ß: ·Not released.

·When any additional requester is opened by RunLame itself, in the meanwhile all input to RunLames's main window is blocked and the pointer changes to busy image.

V1.17ß: ·Not released.

·Bug fixed: Previos versions failed on pathes/filenames including space characters, because Execute() couldn't find them when they are not enclosed in "'s. Now a " is pre-/appended if pathes/filanames contain space characters.

·A seperating space is only inserted in the string gadget when nescessary.

·When commandline string gadget overflows while using the application window feature or the filerequester or the startup multiselect feature, a DisplayBeep() is generated to alarm that not all selected names are reflected in the string gadget.

·Application icon feature added. This is not yet working as stand alone, but in combination with the GUI it is fully functional.

V1.18ß: ·Not released.

·APPICON feature now fully functional. Double click the application icon to quit RunLame.

·LOOP option added. RunLame will pop up again after having executed a commandline, until it is definitely quitted by the user. This feature has been suggested by Milano.

- Bug fixed: Launch menu item is enabled now when a commandline is given at RunLame startup when using the GUI.

- Bug fixed: It was possible to launch an empty commandline from GUI when clearing the string gadget, not pressing Enter, but selecting the Launch menu item.

V1.19ß:
- Not released.

- 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).

- PATH parameter implemented. Supply a path with PATH to set the initial drawer of the file requester. This feature again has been suggested by Milano.

V1.20ß:
- Not released.

- 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.

- Now System() is used to execute the commandline, instead of Execute(). Thus now it's possible to break all breakable runlamed programs. Furthermore the error number generated by the commandline is returned by RunLame now.

V1.21ß:
- Not released.

- Bug fixed: Since V1.19ß there has been an Enforcer hit to $0 when PATH has not been specified. This didn't hurt, because at $0 normally a 0 byte is located, so the PATH was considered as empty string, and this is what we wanted it to be when we don't specify a PATH.

- The output window now will open with full width on all Workbenches (it did formerly open in maximal 720 pixels width).

- 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).

- Some internal improvements (now setting the secondary error code, too, though it seems this is not nessesary).

V1.22ß:
- Not released.

- This version is somewhat experimental, I have to wait for the results of my ß-testers, because I don't own an AGA model. 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. This would be the smartest way to do this all, but I don't know if this works. Furthermore all screens opened by programs in the execution commandline 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)()).

V1.23ß: ·Not released.

·Bug fixed: NOVBRTOFAST switch didn't work.

·Bug fixed: VBRTOFAST did hang the system when the VBR was originally located at $0.l (WHY do I always have to find these nasty bugs myself, I thought I had beta testers :-). This bug has been in RunLame for a rather long time, but I don't know exactly since which version.

·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).

·When LOOP and GUI is used, the GUI pops up first time centered under the mouse pointer, but the next times it pops up always where it was located before disappearing. This feature has been requested by Milano.

V1.24ß: ·Not released.

·The file requester now remembers the last entered directory when OKAY has been selected (not if CANCEL has been pressed). Again a suggestion by Milano.

·When using menu item 'Enter commandline' while window is zoomed, the window will be brought to full size.

·Bug fixed: When the height of font used for RunLame's GUI window drag bar differed from 8, the zoom gadget did still zoom the window to a height of 8 pixels, which resulted in an awful looking window drag bar. The zoomed height of the window now depends on the height of the font used for the window's drag bar. (Does none of my ß-testers use a font with another height than 8 ? Or do they never zoom the window ?)

V1.25: ·Released to the public for the first time.

·It's now possible to turn off the commandline appending feature with NOAPPEND. I really can't see why anybody could need this, but I added this for Milano. I might reference this mode by 'dumbnosemode' :-).

1.57 Second public release

V1.26ß: ·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, ...).

- The windows drag bar height in zoomed state is now calculated as proposed in the include files of C=.

V1.27ß:
- Not released.
- When screen degradation is activated (SCREEN option), the sprite resolution for that PAL screen is explicitely 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.

V1.28:
- Second public release.
- When screen degradation is activated (SCREEN option), 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 launch phase, because all windows keep their pointer for greater convenience.
- The CLI option NOVBRTOFAST has been renamed to VBRTO0. The Workbench tooltype VBRTOFAST has been renamed to VBR.
- Checking if the PAL screen opened really in PAL lores. Results in an error message if a promotor did change the screen mode.
- Bug fixed: mutual excluding Workbench tooltypes didn't produce an error message.

1.58 Third public release

V1.29ß:
- 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.

V1.30ß:
- 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.

V1.31ß:
- Not released.
- Changed the 'VBR to fast' gadget to a cycle gadget, which is more intuitive. Furthermore changed the apropriate menu item into two mutex submenu items.

V1.32:
- Third public release.
- Bug fixed: 'VBR location' submenus were incorrectly reflecting the

startup configuration of RunLame (VBRTOFAST, VBRTO0).

- Bug fixed: RunLame did crash on Kickstarts lower V2.04. CLI and Workbench start of RunLame under pre 2.04 Kickstarts is now possible without crash and memory loss. RunLame will then exit with an error beep but no error message.

- Some minor internal improvements.

- All new AmigaGuide documentation replaces plain ASCII docs.

1.59 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, tète de tomate, greets to Zenith members, too.

- U4ia, may the Lord show you, that he'll never forget about you. This is no hype, you got the vibe, man.

- Thanx to Delirium for their help. Without you it would have taken me much more longer to implement my first GUI.

- 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, Gio, SteveVai, Starfox, Gimli, KicStart, Tirreg, Xed, Exolon, Martin, Duggy. (list in no specific order).

- Greets to all the nice pals on irc (Zop, Gucky, Outland, Gateway, Janne, Juggi, Advance, Goldrnr, Shocker, Mag, VAG, and all I know but forgot now (list in no specific order).

- I can't resist: Comrade, if you really think you HAVE TO write a text about proper programming, then don't forget to tell the people about VBR. BTW, the stock LoadView(NULL) is not working on A4000, so forget about this (and it's not working for 1.3!). And those double CEND are pure stupidity, when C= doesn't say you need two of them for compatibility, then you definitely DON'T need two (even your startup code doesn't need that).

- 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 RunLame. Try to do a better job about compatibility, next time. Using RunLame to make work your own written programs should be a shame for you! One time diskmags will contain compatibility ranking lists, I hope.

1.60 How to contact the author of RunLame

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