Eagleplayer

Eagleeye & DEFECT

	COLLABORATORS		
	TITLE : Eagleplayer		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Eagleeye & DEFECT	February 9, 2022	

	REVISION HISTORY		
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Eagleplayer

1.1   Eagleplayer 2.00 Documentation

Documentation Guide

for

Eagleplayer

Version 2.00

© 1993-97 Defect Software Productions

Table of contents:

1.2   Introduction & Reliability

Introduction & Reliability

1.3 Tips & Tricks

Eagleplayer on own Pubscreen

Eagleplayer currently doesn`t offer the option to open an own screen.

If you on the other hand want to use the program this way the MUI

package offers a suitable tool, "PSI" (public screen inspector).

Just define a screen fitting your needs (refer to MUI`s documentation),

give it a name (e.g. "EagleplayerScreen") and switch the option

"autoclose" on (that makes the screen automatically close when

quitting Eagleplayer)

Then create a script for starting Eagleplayer in the following way

using a text editor of your choice:

MUI:PSI EagleplayerScreen open

Eagleplayer2:Eagleplayer Pubscreen=EagleplayerScreen

Don`t forget to set the Protection bits to "script" (e.g. "protect StartEP +s").

This script may be called now from Shell or using ICONX also from WB.

It is possible to call both command lines from a toolstarter like

Toolmanager, Barnone, Screentab etc. directly after each other.

Eagleplayer & Aminet CD`s

Eagleplayer may be used directly in conjunction with the AminetCD-Guidebrowser

as moduleplayer. More to the topic can be found here.

Eagleplayer & DirectoryOpus

Hints for working with DOPUS V4.x can be found here.

DOPUS5...

playing a module by double click from DOpus5

Just copy the soundmodule filetype from Eagleplayer2:Goodies/Dopus5/Filetypes

to Dopus5:Filetypes. Then you should be able to play your modules simply

be double click onto their names in the DOpus listers. Maybe you have

to add your own name prefixes or suffixes in the file class. Please

refer to DOpus5`s manual for more instructions about fileclasses

handling.

starting Eagleplayer from DOpus (or other programs) invisible

There are several ways to achive this effect. First it is possible to set

the "Iconify" Option in the command line.

advantage: when deiconifying Eagleplayer you`ve got your whole preferred

environment

disadvantage: all Engines specified in your configuration will be loaded upon

startup.

Another method is to disable all visible Engines (scopes, Moduleinfo or

whatever) by specific command line options and start Eagleplayer in

NoGUI mode.

advantage: saves memory and needed time to load Eagleplayer

disadvantage: Eagleplayer can only be handled using Arexx in NoGUI mode

starting to play with a specified subsong

Some sound systems produce multi-modules containing more than only one

jingle within a module. Sometimes it happens that the main melody is

not within the first subsong. (e.g. the tunes from Beast-I)

To start playing automatically with the desired subsong, just add

a comment to the file starting with SS=xxx where xxx is the number

of the subsong to play.

example(type in the shell): Filenote Modules:Whittaker/WH.Beast-2 SS:2

1.4   Erste Schritte in der Benutzung

1.5   Probleme / Fehler im Programm

Executive

Bei der Verwendung von Executive kann es vorkommen, daß der Eagleplayer

beim Abspielen "hoppelt" oder gar nichts passiert. Dies kann daran liegen,

daß sowohl Eagleplayer als auch Executive denselben CIA-Interrupt

verwenden wollen. Ändern Sie dies bitte in den Einstellungen von entweder

Eagleplayer und/oder Executive am sinnvollsten auf

"Timer.device". Damit sollte dieses Problem behoben sein.

Weiterhin wird empfohlen, in den Executive Prefs den Task "Eagleplayer"

auf "noschedule" und "above" zu setzen.

Screennotify.library

So nützlich die Screennotify.library sein mag beim Umstellen bestimmter

Einstellungen der Workbench, so lästig können auch deren prinzipbedingte

Fehler und Nebenwirkungen sein. Sollte es auf Ihrem Rechner verstärkt

zu System-Lockups kommen (nicht nur in Verbindung mit dem Eagleplayer),

probieren Sie es bitte aus, ob sich diese Sorgen nach dem Löschen oder

Umbenennen der Library abstellen. Der Eagleplayer ist für diese Art von

entstehenden Problemen mit dieser Library aufgrund seiner - je nach

Konfiguration - vieler Fenster leider prädestiniert (obwohl ich nochmals

betonen möchte, daß dies auf die Lib zurückzuführen ist und nicht den EP).

A1200 & externe PCMCIA HD/CD-Rom Controller

Die Software einiger über PCMCIA angeschlossene Festplatten & CD-Roms für

den A1200 verwendet den CIA-B Interrupt ohne sich um dessen momentane

Benutzung durch andere Programme zu scheren. Dadurch kann es zu

erheblichen Problemen führen, wenn diese Option im Timingmodus

eingestellt ist.

Bitte ändern Sie in diesem Falle die Einstellung auf "Timer.device".

1.6 Copyright and Conditions on Distribution

Copyright and Conditions on Distribution

The Eagleplayer, external programs, documentation, all interfaces and the

developer examples are ©1993-1997 Henryk Richter and Jan Blumenthal. Due to

our right we will take legal proceedings especially in cases, where parts

of the Eagleplayer package are distributed within other packages, software

products and the like without our permission. In the present version

Eagleplayer is to be taken as Shareware. Therefore we expect a shareware

fee of 20 DM (or 20 US$ when registering outside Germany) if the program is

regularly used, where the fee has to be discharged in BAR, as Eurocheque or

by money-order (unfortunately we cannot accept other currencies or

cheques). You will get the newest version of Eagleplayer in turn, including

functions (e.g. Save, EPDir, all engines), which are disabled in the

sharewareversion.

Please post the outfilled registration-card (Orderform) to:

Buggs Eagleeye

Henryk Richter Jan Blumenthal

Stephan-Jantzen-Ring 47 or Erich-Mühsam-Straße 34

18106 Rostock 18069 Rostock

Deutschland Deutschland

Telephone: 0381 / 1208867 Telephone/Fax: 0381 / 83979

E-Mail: henryk.richter@stud.uni-rostock.de

tfa652@cks1.rz.uni-rostock.de

jan.blumenthal@stud.uni-rostock.de

jb001@cks1.rz.uni-rostock.de

If you have any comments, suggestions, bugreports, new soundformats,

threatening letters, amigas to give away, ... or problems with the usage,

then you can also turn to the adresses above. E-mail is preferred in any

case, because by this method we can answer you the quickest way.

It is allowed to distribute Eagleplayer in public-domain series,

sharewareseries and networks under these conditions:

- the Eagleplayer - package must be complete, the files of this

package are listed in `Distribution`

- it is not allowed to change neither the program nor the documen-

tation, except packing it - but only, if the original state can

be reproduced 100% (e.g. LHA,LZX,ZIP)

- Distribution- costs for Eagleplayer itself or the price for disks

containing Eagleplayer may not exceed 5 DM or the matching amount

in any other currency, the same goes for transfer-fees in

networks. Compilations on CDRom do not fall under this condition.

- We forbid Mallander Software to distribute the player !!!

- Commercial spreading of Eagleplayer is strictly prohibited with-

out written approval from the authors.

We, the authors of the program, do NOT take any responsibility for eventual

damage of hard- or software caused by this program. Use it at your own

risk.

WE, THE AUTHORS, TAKE THE RIGHT TO FORBID SEVERAL FIRMS THE DISTRIBUTION OF

EAGLEPLAYER, IF THEY DON'T FOLLOW OUR COPYRIGHT-CONDITIONS. Eagleplayer was

not written for some people to make money out of it. It should expand the

world of Amiga by a useful music-replayprogram for (nearly) no money. And

these, who don't think about registering, are neither banned nor sent to

Coventry. Eagleplayer is NOT an SASG product and therefore doesn't cost

DM30! Updates are in any case free for registered users!

xfdmaster.library (C) Georg Hoermann

req.library (C) Colin Fox & Bruce Dawson

reqtools.library,powerpacker.library (C) Nico Francois

lh.library (C) Stephan Furmann

LHA,LHX (C) Stefan Boberg

Installer Program (C) Commodore Bussiness Machines

Installer and Installer project icon

(c) Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved.

Reproduced and distributed under license from Commodore.

1.7 Distribution

Distribution

The Eagleplayer is a program-package, which is spread on 2 disks or as LHA archive through Aminet.

About the 2-Disk-Version: If possible, leave the files on both disks where they are to prevent problems during booting or installing. Please keep in mind, that if you relabel the disks e.g. for a pd-serie, you have to change the install-script and "Deutsche_Anleitung" + "English_Documentation" respectively.

Internet Release: (packed as LHA-file)

1.8 What is Eagleplayer all about ?

What is Eagleplayer all about ?

The Eagleplayer is a program package for replaying music of different soundformats in a very comfortable way. It CANNOT be used for creating music, therefore it should not be mistaken for a musiceditor, sequencer, tracker and so on. Internally, it consists of replayers for Sound/Noise/Protracker, Startrekker(AM) & Audiosculpture(AM), other Protracker-Packers and a special Customplayer-Port, yet. But it can also reload other needed external players. The advantage of this method is clear: Only the needed players have to be loaded, which saves loading time and memory. Besides, the length of the mainprogram doesn't reach infinity.

All the internal SoundTrackerRePlayers support Song-Loading (except Protracker-Packers). Songs and samples can be in crunched state. The instruments-path can be changed with the configuration-file. Packed samples are allowed. They will be loaded as RAW-data.

Some features of Eagleplayer:

- several different user-interfaces, free adjustable, free choice of menufonts, partly free choice of backgroundpictures and resizable !

- hardware-independent replay system, where the sound output can be re-directed to different amplifiers to get e.g. digital delayed Surround or 14 Bit Output with Echo-Effects, modules can be played from fast-ram, 16 bit samples can be played with maximum quality directly and without 8-bit-precalculation in 14 bit

- modular structure, with over 150! supported moduleformats, external engines and low-memory system, where the external plug-ins are only

loaded when needed, which saves up to 1 MB memory.

- Up to 64 external loadable engines (e.g. amplifiers, user-interfaces,

ext. depacker, analyzer, patternscrollers known from Protracker, mod-

infos, module-selector, ...)

- Easy and comfortable configuration, all settings can be saved with

one action (inclusive actual music, state and position of engines,

current directory, packmode for saving etc.)

- Online Help per Amigaguide

- can be opened on any pubscreen

- supports AppWindow

- very comfortable handling: e.g. module selecting directly by the

pulldown-menu

- changing directory in the pulldown-menu

- the number of reloadable players is now only limited by your RAM.

- alphabetic ordering of players and engines in the manager, the entire

directory will be displayed, manual reloading of players and engines

by simply doubleclicking on the entry

- Speed control by sliders, partly even with Pitch up/down

- separately switching on/off all voices

- scrolling status-information

- carefully and extensively adapted playerroutines, nearly all of them

allow at minimum volume- and balance-setting,

a big amount of players already supports the Amplifier system

- can load nearly all kinds of packed files, e.g. XPK, Powerpacker,

FileImploder, Crunchmania, lh.library, Stonecracker and supports

archives (LHA,LZX,ZIP,RAR etc.) in a unique way: archives are

handled like normal directories. Besides, the new XFDMaster-

library from Georg Hoermann is supported, which offers a huge amount

of other depackers, XPK-packed files can be depacked during

loading, even with status display

- extensive module infos, if the replayer was adapted for that,

inclusive playtime calculation for different soundformats

(Protracker, Oktalyzer, SonicArranger, Jamcracker etc.)

- mighty ARexx-Port

- DoubleBuffering-function for continuous replay of modules by

immediatly loading the next module

- different filerequesters, depending on personal taste (ASL, Req,

ReqTools)

- runs without problems in multitasking, audio-channels are reserved

properly, all cpu-intensive parts (scopes) are running at low priority

- CyberGraphX is direclty supported by some scopes. (FFT Analyzer)

- it's a commodity (hotkey-activation, control via Exchange)

- Music- Fadein- and Fadeout possible

- replay of a randomly selected module from the current directory

immediatly after program start !

- can be configured using the configfile, by Tooltypes and with

Shell-arguments

- the instruments of a soundmodule can replayed and saved.

- even plays RAW-files e.g. CDDA

- able to use CIA-Timers as well as VBlank and Timer.device for synchronizing

of the replays

- supports Screennotify.library

- is able to find modules in files and executable programs

(Eagleexotic)

- loads modules lists of nearly all music players

- offers a powerful module database

- lists created & saved by Pysion can be handled like normal subdirectories

.

.

.

.

The modular structure allows an optimal setup for each system and the

special likings of its user.

1.9   Development of Eagleplayer

Development of Eagleplayer

The whole thing started way back with the Phenomena-demo "Enigma".

NoisePacker was used there. We ripped the music, but there was no replayer.

Now, a tool was coded in a hurry for playing back this really good module.

Later, other playroutines were implemented (NoiseTracker...). So the

Megaplayer 1.0 was born, which was released on 04/16/1992. At that time it

could replay 16 soundformats. It went up to 25 and the mainprogram got

everytime more obscure. In June 1992, after others had slandered about his

Noiseplayer-styled Megaplayer, Jan decided to write an new

Soundplayerprogram, which should set up new standards. He coded diligently,

created a nice user interface, tried to get even more playroutines and to

implement them into the program. A concept for external players was

designed. Because the concept from Delirium was better that time, this port was built in. Later, our port was extended and improved so much, that our players could only been used with Eagleplayer. We didn't want to live without graphic gimmics, so we developed a system for external, asynchron running analyzertools, which we started to program with the intention to document this new port officially. During development this intention got a dimension an outstanding person cannot understand. Hence this strange name. This anaylzerport has changed to an universal interface for all kind of applications, e.g. converter (Noiseconverter), list managment, module infos, pubscreenselector, patternscroller and many more.

1.10   Future

What is missing in this version yet?

There are many ideas left to be implemented. The thing is, they will need a while to be included.

some examples:

- full 16 Bit samplesupport, probably only for 68040/060 processors due to the extreme expense of calculation times

- ...

- ...

1.11   Updates

Where and when can i get new versions of Eagleplayer ?

New versions can be downloaded in any case from Aminet 'mus/play'. There the new versions will be available at first. Also a homepage is under construction, where you will even find betaversions.

1.12   Update-Service

Update-Service

Every registered user can order the newest version from us without paying anything. We only expect from him to send us 2 disks and an envelope with enough postage for our reply. One of these disks must contain the keyfile of the registered user. Furthermore, we ask you to fill out the updatecard together with your keyfilenumber (you can find it by selecting "About" in the "Project Menu" and then clicking on "Key").

If you are in the mood, you can answer perhaps some questions:

- Are you satisfied with the delivering of Eagleplayer?

- Are there problems with special system-configurations?

- Did you expect more or less from your registered Eagleplayer?

- Is the documentation sufficient for you?

- Which missing function should be included by all means?

- Does Eagleplayer work together with all programs?

- Do the playroutines do their job correctly?

- Are your friends also using the Eagleplayer? If not, why not?

We will copy the actual version of Eagleplayer onto these disks and send

them back to you. If the actual Eagleplayer should be the same yet, we will

wait for the next version.

1.13 Error Description

Error Description

If you want to report an error in Eagleplayer to us, you should pay

attention to some aspects and try to get the error being as reconstructable

as possible. A note like "Eagleplayer crashes sometimes during loading"

doesn't help us at all. May be the module is defect or the decruncher fails

or the memory is corrupted by something. There are many reasons. Here are

some points you should answer in any case to prevent such uncorrect

error-reports:

* Are you a registered user?

* Which Eagleplayer version do you use?

* Add your hardware-configuration!

* Which patches do you have installed?

* Free memory at the moment of crashing?

* Which players/engines were loaded? Is it obviously equal, wether engines

or players are loaded or not?

* Does the errror occurs on other system-configurations?

* Add your configuration-file!

* Are there problems with a specific module? Please always post it !!!

* Does the error only occur on machines with cache?

* How were the internal settings of Eagleplayer? Is it obviously equal,

how they are?

* Was the module recognized correctly?

* Were the gadgets already released (at moduleloading)?

* Was the main window in opened state (yet)?

* Under which kickstart version the error occured? Is it obviously equal?

* Which engines were active?

1.14   Using the Keyfile

Using the Keyfile

Registered users will get a keyfile from us. This file must be copied into

the actual directory of Eagleplayer or to Devs: (this also can be done by

the install script). It allows you accessing the (normally) locked

functions.

The keyfile is person-related. It contains the full useraddress, the

registrationnumber and some other data. We will take legal proceedings in

cases, where the keyfile was given to a third person. In the next version

of Eagleplayer this usernumber will be locked, of course.

All keyfiles can be used by following Eagleplayer versions.

1.15   System Requirements

System Requirements

Minimal Configuration:

- an Amiga with 68020 CPU

- OS 2.04

- 0.5 MB ChipRam and 1 MB Fastram

- harddsik

- monitor with speakers (which must be properly connected, of course)

Possibly Eagleplayer runs even with 68000 based Amigas

but we don`t guarantee for that.

suggested Configuration:

- 68030/50 MHz or better

- OS 3.0 / 3.1

- 4 MB or more Fastram

- HiFi-equipment

Please keep in mind that many functions are disabled under Kickstart 2.0.

From now on we will always refer to OS 3.0 in this guide. Please take care

about not runnig another program which uses the sound-channels. In this

case you will get an error-message after loading a module, see also at

"Error Messages".

Beside the standard-system libraries the player also tries to open the

XPKMaster.library, PowerPacker.library, CRM.library, XFDMaster.library and

LH.library in order to load compressed files. These libraries are not

really needed for starting the program. For the optional file-requesters

beside ASL, the "req.library" or "reqtools.library" are needed.

1.16 Registering

Registering Eagleplayer

The Shareware-fee for the Eagleplayer is DM20 for registrations from within

germany (US$ 20 or £10 outwards of germany) which we expect as chash,

Eurocheque or postal money order. (other currencies or cheques cannot be

changed currently so we can`t accept them!!) You`ll recieve the

latest registered version of the Eagleplayer containing the keyfile and all

Engines. The functions disabled in the standard version will be enabled.

Later versions of the Eagleplayer will accept your keyfile so that you`ll

able to use the all extended functions coming up in further versions of

Eagleplayer. ONLY PRIVATE ADDRESSES !! We won`t write to PLK`s !!

You also can transfer the fee to the following account:

Name of the bank: Ostseesparkasse Rostock

Account number: 1450017491

Bankindex number: 130 500 00

(account owner: Henryk Richter)

Please note in the text your name and that this is the registry for

Eagleplayer. Also in this case the registration-card is needed. You can

post it per letter or better (because faster) per e-mail.

Please send the filled registration form to one of

these addresses:

Buggs Eagleeye

Henryk Richter Jan Blumenthal

Stephan-Jantzen-Ring 47 or Erich-Mühsam-Straße 34

18106 Rostock 18069 Rostock

Germany Germany

Phone: +49 / 0381 / 1208867 Phone/Fax: +49 / 0381 / 83979

E-Mail: henryk.richter@stud.uni-rostock.de

tfa652@cks1.rz.uni-rostock.de

jan.blumenthal@stud.uni-rostock.de

jb001@cks1.rz.uni-rostock.de

In case you have any remarks,suggestions,Bugreports,new Soundformats,

threatening letters,Amigas to give away, ... for us or problems at using,

you can contact the addresses above. If you don`t write for registering

purposes remember: NO DISK, NO ANSWER !!! Send us a Disk and you will get

the latest generic version back, send us money and you get the latest

registered copy. easy, eh ?

If you want to describe us errors, please pay attention to the chapter

Error Description to make the errorfixing easier.

1.17   Installation auf der Festplatte,...

Installation

1.18   Installation on harddisk,...

Installation on harddisk

Eagleplayer is shipped with a Commodore Installerscript for a nearly

automatic installation. Please follow the instructions to avoid problems

(there are also help-texts for all the install steps available). A

directory for Eagleplayer will be created during installation, from where

the program can be started later.

ATTENTION: The structure of the Eagleplayer directory has fully changed in

version 2.0 ! Don't try to install Eagleplayer manually as in

version 1.54 - it would fail 100% .

1.19   Starting Eagleplayer

Starting Eagleplayer from CLI/SHELL

For running from Cli/Shell you only have to simply call the player. Don't

forget to change to the player-dir before starting it (with the DOScommand

'CD') or call it with the full path (e.g. Work:Eagleplayer/Eagleplayer).

Eagleplayer offers many arguments in shell/cli-mode. So you can specify a

module name which should be loaded automatically. You also can run the

Eagleplayer twice, evetually specified modules will be handed over to the

already running Eagleplayer. In any case entered Shell-arguments have

priority over the saved parameters in the config-file. The corresponding

options can be found in the ARexx-chapter and at all descriptions of the

menuoptions.

Attention! Since version 2.0 the behaviour at a shellstart has changed,

Eagleplayer doesn't disconnect itself from the shell anymore. It has to be

started with 'Run' for that case. This was necessary due to some new

features, like support for Aminet-CD's.

Example:

Eagleplayer loadmodule="dh2:Protracker/Industrial/Mod.Bifat_Bunker"

or

Eagleplayer loadmodule "dh2:Protracker/Ravy/Mod.Ani_Soundcompo"

1.20   Starting from Workbench

Starting from Workbench

You can start Eagleplayer from Workbench by simply doubleclicking its icon.

It will be loaded immediatly.

Eagleplayer supports tooltypes since version 2.0, which can be used

together with the configuration file. Please make sure, that the tooltypes

have got a higher priority than the settings in the configfile in any case

and that they are not reset during saving. In cleartext: The options which

are set manually by tooltypes can also only be changed this way. The

corresponding options can be found in the ARexx-chapter and at all

descriptions of the menuoptions.

Example for tooltypes:

config="work:eagleplayer2/my_second_configs/"

(would force Eagleplayer to search the configs in the alternative directory

"my_second_configs")

1.21   The configuration file

The configuration file

Since version 1.10 Eagleplayer loads a configfile. It consists of special

commandlines, which are typed in as strings. You can specify e.g. the

actual player dir or if fadein should be (de-)activated etc. This file is

usually located in the "configs/" dir.

You can specify an alternative config-path. this is done by setting the

environment variable "EP-Configpath". It allows you to use the player with

more than one configuration.

Example: setenv EP-Configpath Work:Music/Player/Eagleplayer/configs_2

copy env:EP-Configpath envarc:

Please keep in mind, that a saved config contains the complete state of

the whole Eagleplayer, that means modulename, actual engine settings, all

paths and all preference settings. The commands for the configfile are

listed in the config, which is shipped with the installation or in a

selfmade one. Not all commands must be specified in the configfile. They

are in nearly all cases compatible with the ones from the ARexxport.

1.22 Die PlayerBatch-Datei

The Playerbatch-file

The Playerbatchfile is an ASCII-File needed to load certain extern players

by recognizing certain file patterns. That means that every external player

only needs to be loaded when a module of this format is to be read into

memory. In the playerbatch File is the Name-recgognition string and the

path of every player written.

Example:

FC.*|FC13.* ="DH0:Eagleplayers/FC 1.3"

*.sdata ="DH0:Eagleplayers/MusicMaker8"

*.sdata ="DH0:Eagleplayers/MusicMaker4"

FC.*|Fc14.* ="DH0:Eagleplayers/FC 1.4"

Please remark that the pathname must be in " ". How you can see more than

one filepatterns are possible. It is even possible to load more than one

replayer for one Filepattern, e.g. when you have got MusicMaker or TFMX

modules where different module types have got the same prefixes. When

"Eject Players" is switched off every replayer will only be loaded once. A

complete Eagleplayer-Batch File you can find within the original

Eagleplayerpackage.

To use this file, the Playerbatch-function must be activated.

Since Version 2.0 there is the Playerloader, which replaces the

Playerbatch, so that the loaded files are analysed for its type and then

the appropriate player will be loaded automatically.

1.23 Usage, Program Control and Configuration

Using Eagleplayer

Introduction First Steps

Mainwindow Gadgets

Menus Pulldown-Menus

Appwindow   Drag & Drop

Pubscreens Where should it show up?

Arexx Introduction to the Arexx-Port

Commands   List of Arexx-Commands

DirOpus 4   How can I control Eagleplayer via Opus?

Aminet-CD`s Eagleplayer and Aminet CD`s

1.24   The Mainwindow

The Mainwindow

The Mainwindow is a graphic user interface (GUI), realised as an external

program (a so called engine) - so it can be exchanged depending on users

taste. It tried to redraw the look of the window with the help of

AmigaGuides features, so don't blame if it's unreadable due to your font

settings in multiview. If so, i suggest to check out the online help

instead.

Eagleplayer1.x-GUI

_____

|*| Eagleplayer V2.00 Generic/Registered | | |

|--|

|¯|1| | | Volume | |zz|

| |--|I |P |---| |

|_|2| | | Balance| |][|

||-|--|

|¯|3||<| < |<<||[]>||> |>> ||>>| > |>| |[] | ˆ |00|

| |--|---|

| |4| Welcome to Eagleplayer |
_____

BifatGUI

_____

|*| Eagleplayer V2.00 Generic/Registered |][|>[]|[.]| | |

|--|

|O|| |O|<<|

| -- |

|\| |/|

| -- |

| ||: |< |< << ||> >| >> >> >| >| ˆ 00|

| _____ _____ |

|[] / | Welcome to Eagleplayer | | | [] [] |

| -- -------- |
_____

1.25 Control by the pulldown menus

Control by the pulldown menus

Eagleplayer offers the following pulldown-menus, which can be selected by

the mouse in the usual manner.

Project

Load Module

Save Module

About Module

Load Player

Delete File

Help

Documentation

Iconify

About

Quit

Commands

Voices

Volume

Balance

Speed

Previous Module

Previous Subsong

Previous Pattern

Restart Song

Play

Fast Play

Next Pattern

Next Subsong

Next Module

Stop

Remove Module

Engines (see also here)

Load Engine

****-GUI Prefs <- Settings for the actual user interface

14 Bit Amplifier

8 Bit Amplifier

Amplifiermanager

BifatGUI

Dirlistviewer

Eagleexotic

Extractor

FFT Analyzer

Formatloader

Levelgraph

Levelmeter

Manager

Messagewindow

Moduleinfo

Noiseconverter

Patternscroll

Patternscroll++

Playerloader

Publicscreenselector

Pysion

Quadrascope

Samplesaver

SpaceScope+

StereoScope

Time

Configuration

Global

Scroll Information

Fade In

Fade Out

Filter

Mastervolume

Songname

Prefix

Load Into Fastram

Playerbatch

Modulesprefs

Rescan Directory

Load Directory

EPDir

Dos-Message

Always Load Dir

Hide All

Saving

Uncompressed

Powerpacker

LH.Library

XPK.Library

Crunchmania

Automatic Save

Overwrite

Safe Save

Samplemode

Save as Protracker

Protect Module

Protectionbits

Set Save Directory

Change XPK-Packer

Change Password

Programming

No New Module

Next Module

Previous Module

Random Module

Songend

Load next yet

Random Subsongs

Automatic Subsong

Load Always

Directory Jumps

Timemode

Calculate Playtime

Start with Random Mod

Immmediate Start

Minimum Playtime

Maximum Playtime

Timing

CIA-Timing

VBLANK-Timing

Timer.device

Automatic Wait

Rasterline Wait

DBF-Loop Wait

reserve Audiochannels

Softinterrupt

Set Rasterlines

Set DBF-Loops

Decrunching

Password automatically

Decrunch first Module

Decrunch files with XFD

Decrunch Executables with XFD

Warning

Small Module

Big Module

Defect Module

Filerequester

Iconify

Look

Choose Menufont

Choose Modulefont

Choose Scrollfont

Change PublicScreen

Filter

Save Configuration

Modules-Directory, *** Entrys

<<Parent Directory>>

Mod.A

Mod.B

OK.C

1.26 The AppWindow-Function

The AppWindow-Function

For the mainwindow Eagleplayer uses the appwindow-function, which is

available since Kickstart 2.0 . Appwindow is a kind of Drag & Drop - the

possibility to click on the workbench-icons of modules, dragging it over

the Eagleplayerwindow and release it there. The module will then be loaded

and played back. If you drop several icons via multiselect over the

Eagleplayerwindow, the corresponding modules will be overtaken

alphabetically into the pulldown-menu. The icons can be selected from

different directories!

Since Opus5 this function got even more important, because Opus lets you

Drag& Drop opuslister-entries into other windows, if they support it. So

you also can pick a file from a opus-lister and release it over the

Eagleplayer-mainwindow in order to load & play it back automatically.

1.27   pubscreens

Pubscreens

Since version 1.10 Eagleplayer is able to open itself on pubscreens. This

pubscreensupport affects to all windows of Eagleplayer and of the

subprograms, if not mentioned otherwise, so all engines open on the same

screen. Here a small list of publicscreen names:

CygnusEdScreen1 (under "Rendering choices" in the Environment-Menu

set "make screen public"to on)

BootX

PowerPacker

DOPUS.1

TERM

EDGE

Since Version 1.52 the engine called PubScreenSelector exists, with which

you can choose a pubscreen by simply clicking with the mouse on an entry,

so Eagleplayer will open itself an the choosen Screen. It can't be easier!

1.28   Commodity-funktions

Commodity-functions

Since version 1.10 Eagleplayer has a commodity-port, which is only

available under Kickstart 2.0 and upper versions. Eagleplayer can be called

up by selfdefined hotkeys. Default is the key-combination "CONTROL ALT E".

When activating this hotkey, the mainwindow with engines, config-window

etc. will be opened and put to front or it will be closed.

With the help of Commodities-Exchange furhter commands can be realised,

e.g. Quit.

The options "CX_Popup", "CX_Popkey" and "CX_Priority" can be specified in

the configuration file as well as via Shell and Tooltypes.

In opposite to the rules for small commodities, which close its

configurationwindow but stay in memory, the Eagleplayer quits, when

closing its window. An Iconify-function is extra available.

1.29   Internal DePackers

Internal DePackers

Eagleplayer can depack datafiles in the current version, if they are packed

with one of the following crunchers:

- CrunchMania ;only with Library, but all modi

- Powerpacker (if available, the library will be used)

- LH (PMC from Powerplayer- but where is the POWER

of the Powerplayer???)

- Fimp

- Pack (Soundtracker-Songpacker)

- XPK (with most packers (SQSH,NUKE) the file will

be decrunched during loading !!!)

- StoneCracker V4.0

The supported XFDMaster.library offers further packers, please refer to its

documentation, which formats are handled by this library.

Using Fimp, only one memoryblock will be allocated, into which the module

will be loaded and decrunched. This method works very well, but only with

Fimp. With other packers a memoryblock for loading and one for decrunching

must be allocated. This method needs ofcourse much memory and this is less

fine. After decrunching, the memory of the loaded and crunched file will be

freed.

Eagleplayer supports the XPKMaster.library, if the corresponding libs are

installed on harddisk. If a file needs a password, you can type it into a

stringgadget in the mainwindow. The advantage of XPK is, that the file can

be depacked during loading. Besides, this saves sometimes much memory,

depending on the used XPKLibrary.

For decrunching PCM-files, meaning files packed with the LH.library, this

library is needed. If it can't be found the module can't be depacked.

Pack is the SongPacker of the old trackers (NoiseTracker, ProTracker,

Startrekker).

Powerpacker-, Fimp-, Pack- and Stonecracker4.0-files can be decrunched

without any further tools. These decrunchroutines are built into the

Eagleplayer. If the Powerpacker.library is found, it will be used for

depacking PP-files.

Furthermore, Eagleplayer can also depack executables, if they are packed

with ImploderV4.0 (NormalMode without Library). This function was

implemented, because the program got bigger and bigger, especially the

players and engines. To make these usable for a disk-user they all can be

simply packed. CustomModules can also be imploded, RonKlaren-mods may not

be packed! In coming versions more file-decrunchers will be supported. I'am

sorry for chosing imploder for that, but better than not crunching, or?

Since version V1.52 the XFDMaster.Library is supported, that means that you

also can load PowerPacker-Executables if you own the appropriate slave. So

you can also pack players and engines with Powerpacker, Crunchmania or

Stonecracker, if your version of the XFDMaster.library supports this.

Attention: Before activating this, please read the documentation to the

corresponding function !.

You can get XFDMaster.library from

Georg Hörmann

Martinswinkelstraße 16c

82467 Garmisch-Partenkirchen

or on wellsorted PD-series / Aminet.

1.30   Introduction and Description of the Arexx-port

Introduction and Description of the Arexx-port

Finally, from Version 1.10 on the Eagleplayer supports ARexx. Therefore the

Eagleplayer offers a lot of commands which make the Eagleplayer able to be

practically remote controlled. All arexx-commands which belong to the

corresponding Menufunctions are also listed in these parts of the guide.

The ARexx-port is named rexx_EP.

An Adaption to DOPUS is therefore no problem at all.

Attention: Due to the new structure of the whole Eagleplayer 2.0 some

commands have changed, furhtermore, it is recommended in

some cases to use the shell-options instead.

So, an ARexx-script for loading a file via opus is no more

needed. The commandline Eagleplayer Loadmodule="{f}" toge-

ther with the command-type "AmigaDOS" and the option "run

asynchronously" does the same.

1.31   Arexx-command overview

Arexx-command overview

Following this text you will find all ARexx-commands supported by the

Eagleplayer. If no error occurs, the result returned by the Eagleplayer to

the calling Arexx-Script is the same which would be scrolled into the main

window .Of course not if the Eagleplayer expects exact arguments e.g. using

the status command. If an error occurs in RX_result the error-text will be
returned.

The User should of course test if the different possibilities can be used
in any case. It won't work e.g. if you wanna save a module but the current
replayer doesn`t support saving. If you tested it before, the problem is
solved. We offer a lot of status informations, make use of them ! If we
talk filenames from now on we mean FILE AND PATH ! (if no other case is
mentioned)

AboutEP

Aboutmodule

Addlistentry

Addlist

Allocchannels

ASCEngine

Automaticsave

Autopassword

AutosubSong

Balance

BigModule

CalcDuration

ChipRamAmplifier

CX_Popup

CX_Popkey

CX_Priority

Config

Crunchmode

DBFLoops

Defaultspeed

DeleteAllEngines

DeleteAllPlayer

DeleteEngine

DeleteFile

DeleteGui

DeletePlayer

Dirjump

Enable

Eject

EjectPlayers

EjectEngines

EPDir

ExtractFirstModule

FadeIn

FadeOut

Fasterspeed

FastRamAmplifier

Fastplay

Filerequestermode

Filter

Help

Helplink

Hide

HideAll

Iconify

Iconifymode

Instruments

Loadalways

Loadbefore

LoadConfig

LoadDir

LoadEngine

LoadEngineDir

LoadFast

LoadGui

LoadModule

LoadPlayer

LoadPlayerConfig

LoadPlayerDir

LockModule

Mastervolume

Menufont

Mintimeout

Module

ModuleFault

Modulesfont

NextModule

NextSong

NextPatt

NoEngine

NoGui

Notify

Overwrite

ParentDir

Password

Pause

PBits

Play

PlayerBatch

Playerconfig

PrevModule

PrevPatt

PrevSong

Prefix

Printtext

ProgramMode

Protect

PubScreen

Quit

QuitEagle

QuickStart

Randomsong

Randomstart

Rasterlines

ReplaySong

RescanDir

SafeSave

Samplemode

SaveAsProtracker

SaveConfig

SaveDir

SaveModule

SaveModulePrefs

SavePlayerConfig

SaveT

Scanalways

Scrollfont

ScrollInfos

Scrolltext

Slowerspeed

SoftInt

SongEnd

SongName

SmallModule

Speed

SSDir

Status

Stop

SubSong

Timemode

Timeout

Timingmode

ToggleFilter

Voice

Volume

Waitingmode

XFDDecrunch

XFDLoadSeg

XPKPackmethod

That's all yet! :-)

1.32   Arexx-Adaption to Directory Opus

Arexx-Adaption to Directory Opus

In this chapter is described how to adapt ARexx Scripts to control the

Eagleplayer via the Directory Opus. Therefore a lot of scripts are

supplied. Our ARexx-Scripts are adapted to DOpus4. (ARexx portname of the

first opened DOPUS: "DOPUS.1"). Please notice that the DirectoryOpus 3 uses

another Arexx portname ! To Adapt the Arexx-Scripts copy these to your

assigned "rexx:" directory. After this start the configuration program of

the DOpus. Choose the "BUTTON"-Gadget and click on one of the free Gadgets

of the gadget bank appearing at the top of the screen. It is best if you

open a new bank before (click on "NEW BANK") because of the many

Arexx-Scripts we wrote for this piece.Now appears the Button-Edit-Screen.

Enter the name of the button (e.g. "Load Mod"). Click on the gadget "NEW

ENTRY". Then some gadgets are freed and the cursor appears in the line

under the big listview gadget. Here you enter "rx rexx:EP_Loadmodules.dopus

{s}". Lefthand to this line you find a cycle button.If there doesn't stand

"AmigaDOS" click on this switch until "AmigaDOS" appears as type. Now you
still have to change the mode. This will be done by setting "run
asynchronously" in the "FLAGS ..." to on. Now select "OKAY" until you are
back to Opus (or save the configuration first). Select one or more modules
and then click onto the Loadmodule gadget.

This Arexx-Script takes all selected files and gives them to the
Eagleplayer which loads the first of them and puts the rest into the pull
down menu. If only one module is selected the whole directory will be
loaded. If the Eagleplayer wasn`t loaded at this time the Arexx-script does
it for you but make sure that the path for the Eagleplayer program is
correctly set in this script. If not so then load the script into a
texteditor and change the path of the Eagleplayer to the right one.

If you for example wanna load a module via double click you have to define
a filetype. Therefore click on the gadget "Filetypes". Now choose the item
"NEW" from the Pulldown Menu. There where the cursor now appears enter
"Soundmodule" and in the next stringgadget "Music". These inputs are only
for Orientation and have no special meaning. Now choose "New Entry". Click
on the Lens-symbol and then in the appearing requester "Matchname". There
where the cursor now appears enter "(MOD|CU).*". Then press "OKAY"

Now you are back to the window where you entered the "NEW" menuitem. There
you click on the "NEW" gadget. Now a requester appears where you can find
the word "Soundmodule" anywhere. Click there and a new window will open. At
the top in the middle of the window there you click onto the event "Double
Click". As event description you enter a text of your choice e.g. "Loading
soundmodule". Click on "New Entry", enter "rx rexx:EP_loadmodule.dopus",
and click "OKAY".

Now you have created a fileclass that would DOPUS make to give
automatically the double-clicked musicfile to the Eagleplayer which loads
and plays it when the name of the musicfile starts with "MOD." or "CU." But
don`t forget to set the path for the Eagleplayer correctly in the
scriptfile.

1.33 aminetcd

Using Aminet-CD`s with Eagleplayer

What is the Aminet at all? Aminet is a huge shared archive, where authors
can release their PD- and FD-programs. Aminet is built up by ca. 15
computers worldwide - connected by Internet, a computer network between
most of the universities and many firms over the world. Also at many other

places - e.g. at over 50 mailboxes - you can get regularly updated Aminet

files. It doesn't matter where a software is uploaded to Aminet, it will be

available worldwide within 24 hours. A speed, no other media can offer. The

archive consists of more than 9 gigabytes of software, therefore it is the

biggest network-based software archive of the Internet at this moment. So

we take now the chance to thank Urban Mueller a lot not only for XPK :-)

but also for administrating Aminet.

Ok, back to the topic: On the Aminet CD's, consisting of the new stuff from

the archive at the time, a huge number of music modules can be found. They

can be directly selected in the AmigaGuide-Index and played back with an

external modplayer. To use the Eagleplayer for it you have to start the

"Prefs"-program on the CD and type in the following line under the label

"Mod Player": "<Eagleplayer-path>/Eagleplayer loadmodule" The path is the

position of the Eagleplayer-drawer on your harddsik (you can use the

filerequester instead by clicking on the "Set"-gadget) and after the path

and name the word loadmodule must follow.

Example1: work:music/players/Eagleplayer loadmodule

Example2: work:music/players/Eagleplayer iconify=on loadmodule

1.34   AboutEP

AboutEP

This function shows some information about Eagleplayer, wether the program

is registered or not, and if yes the name and address plus the

keyfilenumber of the user.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

aboutEP

1.35   AboutModule

About Module

Scrolls an information text through the mainwindow, containing infos about

the music and replayer. The amount of the info depends on the replayer. In

the worst case only the name and length of the module will be shown. If no

module is loaded, the Aboutwindow of Eagleplayer will be opened.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

Aboutmodule - These two function return the same text, which is also

status m abo shown in the scroller.

1.36 autosubsong

Automatic subsong

Using soundsystems, which contain several musicparts, all subsongs will be
replayed before a new file is loaded - depending on the settings for
Programmode. Supposition is either the support of Songend detection by
the player or Timeout, respectively Randomsong set by the user.
Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Autosubsong state - state: [yes,on or 1]=on, [no,off or 0]=off
Arexx:

status g ass - returns actual state (yes/no)
Tooltype:

Autosubsong=state - state: see above

1.37 automaticsave

Automatic saving

If activated, the file will be saved without further name-requests under
the actual filename (or the songname + filename extension, depending on
settings) to the preset directory.
Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

AutomaticSave state - state: [yes,on or 1]=on, [no,off or 0]=off
Arexx:

status g ats - returns actual state (yes/no)
Tooltype:

AutomaticSave=state - state: see above

1.38   allocchannels

Allocate Audiochannels

The audio-hardware and its software-based interface of the Amiga are not
designed in the way, that several programs can use them simultaneous (or
only with enormous restrictions). Therefore the Eagleplayer has to use them
alone and without any influences from other programs. To avoid such
problems, Eagleplayer allocates (if activated) the audiochannels for sole
usage for the time in which the replay is done. It's recommended, to
not switch off this option.

Since version 2.0 the audiochannels are also given back when the Pause-
function is used and only allocated during replay, so the actual musicfile
hasn't to be removed from memory.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Allocchannels state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g ach - returns actual state (yes/no)

Tooltype:

Allocchannels=state - state: see above

1.39   autopassword

Automatic password

For unpacking/decoding of files, protected with passwords, you can allow by
this option an automatic decoding with the password, which is set in the
change passwordfunction. This makes only sense, if the files have got the
same password.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Autopassword state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g aps - returns actual state (yes/no)

Tooltype:

Autopassword=state - state: see above

1.40 Add list entry

Add list entry

Not documented!

1.41 ASCEngine

ASCEngine

Not documented!

1.42 Add list

Add list

Not documented!

1.43 balance

Balance

This regulator can be used to change the volume-ratio between the left and

right channel. If the slider is dragged to the left, the volume for the

right channel decreases and vice versa. A balance value of "0" means, that

both sides work with the same volume. If a replayer doesn't support

balance, this slider (and the corresponding menu entries) will be disabled.

The BifatGUI owns 2 extra buttons for the balance. If clicked once, the

left button sets the balance to the maximum left, and if selected again, to

the middle position. The left button works analogue.

The possibilities offered by the menu, allow the shifting of the

volume-ratio by one step to the left/right and a reset to the middle

position.

Keyboard:

-

Menu-Shortcut:

A-9 = Balance further to the left

A-0 = Balance further to the right

A-8 = reset to the middle, same amount on both sides

Arexx/Shellstart:

Balance number - number: -255...0...255 (-255=outer left; 0=middle,

255=outer right)

Remark: Eagleplayer calculates internally with

256*2 balance steps, but offers the user

actually a range from -64..64 as value,

to go conform with the Amiga-hardware-

restrictions)

Arexx:

status g bal - returns actual value (-255...255)

Remark: Because Arexx has got problems with

negative numbers, the returned numbers

for values <0 go from 65535 (=-1) to

65281 (=-255), which have to be re-

calculated by the Arexx-program.

Tooltype:

Balance=number - number: see above

1.44 bigmodule

Warning on big modules

If the option is activated and a too long module should be played next,

Eagleplayer opens a requester, which informs about that. Too big modules

are (normally) no problem, but so you can find out, where to save some

harddisk space.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Bigmodule state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g bmo - returns actual state (yes/no)

Tooltype:

Bigmodule=state - state: see above

1.45 ChipRAMAmplifier

ChipRamAmplifier

Not documented!

1.46 Popup

PopUp

This function opens/closes Eagleplayers mainwindow, depending on the

specified arguments.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

CX_Popup yes - mainwindow will be opened

CX_Popup no - mainwindow will be closed

status g mwi - yes/no as result

Tooltype:

CX_Popup=yes - opens the mainwindow on startup

CX_Popup=no - the mainwindow will not be shown on startup

1.47 crunchmode

Crunchmode

The options set here change the behaviour of Eagleplayer when saving the

actual file(s). You have the choice to save the file(s) either unpacked or

packed with the compressors mentioned later. Some of these methods

(Powerpacker, Crunchmania and LH.library) can activate a

soundmodule-specific, a socalled delta-mode, in order to gain much more

compression in most cases, than a normal packmode.

( ) unpacked - the file(s) will not be packed

( ) Powerpacker - powerpacker.library from Nico Francois is used

( ) LH.Library - lh.library from Stefan Fuhrmann is used

( ) XPK.Library - the actual set XPK-Packer is used

for saving

( ) Crunchmania - crm.library from Thomas Schwarz is used for

packing

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Crunchmode number - number: 0 = unpacked

1 = Powerpacker

2 = LH.Library

3 = XPK.Library

4 = Crunchmania

Arexx:

status g cmo - returns actual state (0...4)

Tooltype:

Crunchmode=number - number: see above

1.48   calcduration

Calculate duration

If activated and supported by the replayer the duration of the actual

musicfile will be calculated and shown in the moduleinfo. You can refer to

the Manager if you want to know which replayer supports this duration

calculation (e.g. Protracker+Clones, SonicArranger, Jamcracker, Oktalyzer,

AIFF, ADPCM)

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

CalcDuration state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g cad - returns actual state (yes/no)

Tooltype:

CalcDuration=state - state: see above

1.49   CX_Popkey

PopKey

This function sets the hotkey, with which the entire Eagleplayer can be

hidden or opened again.

Tastatur:

-

Menü-Shortcut:

-

Arexx:

CX_Popkey <Hotkey> - the hotkey will be set

status g pke - hotkeystringyes as result

Tooltype:

CX_Popkey=<Hotkey> - the hotkey will be set

1.50   CX_Priority

CX_Priority

this function sets the commoditiy priority.

Tastatur:

-

Menü-Shortcut:

-

Arexx:

CX_Pri <number> - Priority

Tooltype:

CX_Pri= <number> - Priority

1.51   config

Config

Not documented!

1.52   defaultspeed

Default Speed

The Replay Speed will be reset to the normal value.

Keyboard:

-

Menu-Shortcut:

A-#

Arexx:

Defaultspeed

1.53   documentation

Documentation

This command forces Eagleplayer to open and view (via amigaguide.library)

this file you're reading now (documentation or online-help respectively).

Remark: When you close the amigaguide-window, the file of the Doc will not

be automatically freed. To achieve this choose the menu item again.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

Documentation

1.54 dbfloops

set DBF-Loops

When the waitingmode is set to "wait with DBF-loops", several

waiting-loops are run by the CPU in order to play the musicfiles correctly

timed. This mode however strongly depends on the CPU, so it's recomended to

use the other timing methods instead, which are offered. A useful value for

the 7.09 MHz 68000-processor is ca. 300. If somebody has boredom and

desire, he can perhaps check out, which value works fine on 68060, i really

would like to see this number...

You can recognize a too small waiting-loop value if some notes are clipped

during replay. Increase in this case the value or switch to

"automatic wait".

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

DBFLoops number - number: amount of cycles to be waited

Arexx:

status g dbl - returns actual value

Tooltype:

DBFLoops=number - number: see above

1.55 dirjump

Directory jumps

If activated: During automatic loading of a next file, Eagleplayer jumps

into the subdirs of the last user-specified directory and searches there

for the next file to be played. This function is very useful, especially if

you e.g. sorted your musicfiles by the composers into different directorys.

This function makes only sense in connection with LoadDir and "random

module" (set in programmode.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Dirjump state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g dij - returns actual state (yes/no)

Tooltype:

Dirjump=state - state: see above

Example:

Assume, your directorys on your harddisk are organised in this way:

Modules:

|

Protracker

| |

| Audiomonster

| |

| Dreamer

| |

| Folxmusix

| |

| Gabba

| |

| Hardcore

.

.

.

If Dirjump is activated, then, if you select the protracker-dir by a

filerequester, Arexx etc and activate NextModule, Eagleplayer will take the

next file either from "Audiomonster", "Dreamer" or "Folxmusix" etc. and

return to the protrackerdir, waiting for a next file-request.

1.56 DeleteFile

Delete file

This function opens a filerequester for selecting a file to be deleted.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

deletefile [path+name] - if a file is specified, no requester pops up

and the file will be deleted immediatly

1.57 extractfirstmodule

Extract first module

This option changes the behaviour of Eagleplayer regarding to archives.

(LHA, LZX,ZIP etc.). Normally, the Extractor behaves in such a manner,

that archives are handled like normal directories, so the archived modules

can be selected separatly. This however causes some trouble when accessing

the modulearchives on the Aminet-CD's, because normally, these archives

contain one module only - The expense for going through the archives would

raise unnecessarily.

In short words: If the option is activated, the first file found in a

selected archive will be decrunched and played back, if it is a module. It

is highly recommended to activate the Hideall-function when using

"extractfirstmodule", because otherwise it can easily happen, that

Eagleplayer tries to decrunch and play back the "readme"-file in the

archive.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Extractfirstmodule state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g efm - returns actual state (yes/no)

Tooltype:

Extractfirstmodule=state - state: see above

1.58 epdir

EPDir

EPDir is a function for speeding up the directory-scan. The whole contents

of the directory will be written to a small file named ".epdir". When that

directory should be scanned afterwards, this file will be used instead and

the entries will be shown in the menu, if the contens has not changed

during saving the ".epdir"-file. The resulting speedup is especially

remarkable when using large directorys, fragmented harddisks which run with

the normal FFS filesystem. This function shows only effect in connection

with LoadDir.

Attention: In rare cases the Notify-function can collidate with the

EPDir-function, where the directory will always be scanned directly from

disk. In this case you should switch off "EPDir" or Notify.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

EPDir State - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g epd - returns actual state (yes/no)

Tooltype:

EPDir=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.59 Enable

Enable

This function allows you to switch on/off a replayer. The first argument is

the number of the replayer, the second one the wanted player-state.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

enable num state - "num"=number of replayer, "state" = 1 for enabled,

0 for disabled

status p ena - returns, wether the corresponding replayer is enabled

or not

1.60 Eject

Eject Module

Stops the actual module (if one is replayed) and removes it from memory.

Then the audio-channels are released, so other programs kann access the

audiofunctions of your Amiga.

Keyboard:

-

Menu-Shortcut:

A-e

Arexx:

Eject

1.61 FadeIn

Fade in

If Fadein is set, all modules will be slowly faded in upto the set volume.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Fadein state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g fai - returns the actual state (yes/no)

Tooltype:

Fadein=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.62 Filter

Filter

Switches on/off the LowPass-filter of the Amiga. This doesn't work for the

Amiga 1000 and 2000A. If activated, the output sounds very dull, because

all frequencies above 7 KHz are cut by the LowPass. So this function is

rather meant for setting the filter on/off during the start of Eagleplayer,

see also the comment about ToggleFilter.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

filter state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g led - returns the actual state (yes/no)

Tooltype:

filter=yes/no - state: [yes,on or 1]=on, [no,off or 0]=off

1.63 filerequestermode

Filerequester

With these 3 options you can choose your preferred filerequester, which

will then be used by the Eagleplayer. You can choose between ASL (system

standard), reqtoolslibrary (Nico Francois) an the older req.library (Cygnus
Software).

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Filerequestermode number - number: 0=req.library,1=reqtools.library,

2=asl.library

Arexx:

status g frm - returns actual state (0...2)

Tooltype:

Filerequestermode=number - number: see above

1.64 fasterspeed

Faster speed

The replay speed will be incremented by one step.

Keyboard:

-

Menu-Shortcut:

A-+

Arexx:

Fasterspeed

1.65 FadeOut

Fade out

If Fadeout is set, all modules will be slowly faded out and stopped at the
end of replaying.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

FadeOut state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g fao - returns actual state (yes/no)

Tooltype:

FadeOut=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.66 FastPlay

Fast Play

"Fast play" makes it easier to find a special passage in the title.

Keyboard:

-

Menu-Shortcut:

A-o

1.67 hideall

HideAll

HideAll takes care about not showing files from the actual directory in the

menu or dirlistviewer, which names are known as non-music-files.

List of hidden file suffixes:

".info",".s",".m",".i",".o",".h",".c",".a",".z",".hlp",".nfo",".lib"

".bat",".bak",".dmp",".dms",".doc",".bbs",".msd",".dok",".asc",".diz"

".txt",".ins",".asm",".rs",".nt",".fd",".ss",".as",".nt.pp",".catalog"

".guide",".hyper",".pic",".gui",".offs",".help",".rexx",".prefs",".instr"

".config",".device",".library",".font",".samp",".ilbm",".fastdir"

".backdrop",".speeddir",".isdir",".sadir",".okdir",".epdir",".display"

".displayme",".readme","readme","liesmich",".ip",".i.n",".ip.n",".ip.l"

"-handler"

List of hidden file prefixes:

"info.","smpl.","rsmp.","smp.","jpns.","index",

This function makes only sense in connection with LoadDir.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

HideAll state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g hia - returns actual state (yes/no)

Tooltype:

HideAll=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.68 Hide

Hide

This option closes all windows, which belong to Eagleplayer. In

opposite to the "iconify"-function all visual parts of eagleplayer will be

hidden. You can activate the mainwindow again by different possibilities.

Firstly by using the hotkey (normal: CTRL-LALT-E), open it via "Exchange"

or by relaunching Eagleplayer (this sends an "open"-message to the already

running Eagleplayer).

1.69 Help

Help

This function enables Eagleplayers online-help. After activating it, all

gadgets and menus are available. Selecting one, results in opening an

Amigaguide-window where the corresponding item is described using this

documentation.

Keyboard:

HELP

Menu-Shortcut:

A-h

Arexx:

help yes/no - switch on/off

status g hel - returns the state of the help-function as result

(yes/no)

1.70 iconifymode

Iconifymode

Here you have the choice between the 2 options, how the Eagleplayer should

react on the Iconify-command. On the one hand a appropiate menu-entry in

the tools-menu of the workbench will be created, on the other hand an icon

will be created on the workbench, with which the windows of Eagleplayer can

be reactivated.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Iconifymode number - number: 0=Appicon,1=Appitem

Arexx:

status g icm - returns actual state (0...1)

Tooltype:

Iconifymode=number - number: see above

1.71   Iconify

Iconify

Closes the mainwindow and all other opened windows of Eagleplayer and

creates a so called "AppIcon" on the Workbench or adds an entry to the

"tools"-menu of the Workbench. To re-open Eagleplayers windows simply

doubleclick on the appicon or select the menu item.

Keyboard:

i

Menu-Shortcut:

A-i

Arexx:

iconify yes

iconify no

status g ico - yes/no as result

Tooltypes:

iconify=yes

iconify=no

1.72   LoadEngine

Engine laden

Opens a filerequester for adding/removing one/several engine(s) to/from the

available ones.

Remark: All engines located in the "engines"-dir are already shown in the

pulldown-menu + manager. They can be (de)avtivated by simply doubleclicking

it in the manager, so this function is normally not necessary.

Keyboard:

e

Menu-Shortcut:

A-u

Arexx:

LoadEngine [name] - if a engine-name is specified, it will be loaded imme-

diatly, otherwise a filerequester pops up (analog to the

menu-function)

1.73   LoadPlayerDir

LoadPlayerDir

This function allows to include many Abspieler at once manually. That is

normally not necessary in cases where Playerloader and Playerbatch are

activated. All replayers from the chosen directory will be included into

the playerlist.

Remark: The Manager already lists all replayers from the

"Eagleplayers"-dir, which can be loaded/removed by simply doubleclicking on

the specific list entry.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

LoadPlayerDir [path] - when specifying a path, the replayers will be loaded

immediatly, otherwise a filerequester pops up.

status g dir - returns the actual replayer-path

see also: LoadPlayer

1.74   LoadPlayer

LoadPlayer / AddPlayer

This function allows to include a replayer manually. That is normally not

necessary in cases where Playerloader and Playerbatch are activated. When

calling it from the menu a filerqeuster pops up, where one or more

replayers can be selected, which will be attached to the list.

Remark: The Manager already lists all replayers from the

"Eagleplayers"-dir, which can be loaded/removed by simply doubleclicking on

the specific list entry.

Keyboard:

a

Menu-Shortcut:

-

Arexx:

LoadPlayer [path and name] - If a path is specified, the replayer will be

immediatly loaded, otherwise a filerequester

pops up. (analog to the menu-function)

see also: LoadPlayerDir

1.75 loadbefore

Load next module yet

This option forces Eagleplayer to immediatly load the next module from the
actual list /actual dir after loading a new musicfile. The choose of this
next module depends on the settings in Programmode. This allows a
continuous play of musicfiles without break. This function is also known as
"Doublebuffering". Especially megamixes, which are spread over several
musicfiles, can be replayed correctly (e.g. "RAF Megamix2", "LSD-Total
Kaos" or "Max Mix"[Flash Prod.]).
This function makes only sense in connection with LoadDir.
Keyboard:
-
Menu-Shortcut:
-
Arexx/Shellstart:
Loadbefore state - state: [yes,on or 1]=on, [no,off or 0]=off
Arexx:
status g lbf - returns actual state (yes/no)
rmi
Tooltype:
LoadBefore=state - state: see above
Small note: In the doc from the APlayer was stated, that the APlayer is the
first Player with such a feature. Very curious, because this option was
built into Eagleplayer far before the APlayer itself got released...

1.76 loaddir

Load Directory

When Loading a musicfile you have the choice by that option, wether the
whole directory should be scanned and viewed in the Pulldown-Menu and the
Dirlistviewer, or not.
Keyboard:
-
Menu-Shortcut:
-
Arexx/Shellstart:
LoadDir state - state: [yes,on or 1]=on, [no,off or 0]=off
Arexx:
status g ldd - returns the actual state (yes/no)
Tooltype:
LoadDir=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.77   loadfast

Load into Fastram

You can choose, wether the musikfiles should be loaded into chipram or into

the speedy fastram (if activated).

It is strongly recommended to activate this option, especially when using

8 Bit or 14 Bit-Amplifier, because the chipram is up to ten times slower

than fastram, especially on fast machines. If a soundformat is active,

which doesn't work together with the Amplifiers the sounddatas will be

automatically redirected to chipram, if "LoadFast" is active. The only

reason for deactivating this option, which comes to my mind, is an

unexpanded Amiga500/2000 with 68000-CPU, which heavily slows down when

modules are played from fastram. In this case you should set the

ChipRam-Amplifier as the first entry in the amplifier-list by using the

Amplifiermanager, so the calc-time will be minimated during replay from

chipram, as known from the 1.xx-versions of Eagleplayer.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

LoadFast state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g lfa - returns actual state (yes/no)

Tooltype:

LoadFast=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.78   lockmodule

Lock module

This switch takes care about not automatically stop or replace the actual

module by the next one from the list or removing it from memory,

independent from the settings for TimeOut, Songend etc. This function

will be deactivated if the button is selected again, or if the user selects

the next musicfile.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

LockModule state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g mlo - returns actual state (yes/no)

Tooltype:

LockModule=state - state: see above

1.79 loadalways

Load always

In this case, after Songend or Timeout, Eagleplayer will search through

the list of available files for the first file it knows it is a musicfile.

This is a help in cases, where the directory also contains e.g. textfiles.

If the loadalways-function is deactivated, Eagleplayer would stop after the

end of the actual module with the message "unknown format", if the next

file is a non-musicfile.

This function makes only sense in connection with LoadDir and a

Programmode exclusive "no new module".

Remark: if Eagleplayer doesn't find any playable file, the directory would

be completely scanned again and again. To prevent and stop this, switch off

this function in this case by the corresponding menu-item.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

LoadAlways state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g law - returns actual state (yes/no)

Tooltype:

Loadalways=state - state: see above

1.80 LoadModule

Load Module

Opens the filerequester for selecting a module to be played, it is also

allowed to select several modules at once (holding shift-key down while

selecting); in this case the list of selected files will be remembered and

used by the functions "next module" and "previous module"; Furthermore the

modules can be directly selected from the pulldown-menu (modulesdirectory).

If only one module was selected, the whole directory will be read; all

files, which are per definition no musicfiles (e.g. '.readme', '.txt', '.c'

or '.asm') will be hidden automatically (HideAll). They will not be

displayed in the menu.

Keyboard:

-

Menu-Shortcut:

A-l

Arexx/Shellstart:

loadmodule name1 name2 ... - Eagleplayer loads the first specified module

and reads the whole dir, if only one argument

was typed. Otherwise only the specified modules

will be included into the modules-menu

Example:

loadmodule "dh2:Protracker/Mod.Bifat_deadlocked EBM"

Tooltype:

loadmodule=name - as above, but connected with a equal-sign

1.81 Menufont

choosing a menufont

Opens a font-requester for choosing a font for the pulldown-menus except

the modules-menu.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

Menufont [name.size] - if a name is specified, the corresponding font will

be automatically set, otherwise the font-requester

pops up

status g mfo - returns the actual font

1.82 mintimeout

min. playtime

If you specify a minimal playtime, the replay will be stopped after this

time, independent on Songend and/or max. playtime. This function is very

useful for very short musicfiles, which nevertheless should be played some

minutes. The value is specified in seconds and typed in a requester, a

value of "0" deactivates the function.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

MinTimeout number - number: max. playtime in seconds (0=off)

Arexx:

status g ti2 - returns actual value (in seconds)

Tooltype:

MinTimeout=number - number: see above

1.83 Mastervolume

Mastervolume

Is Mastervolume set, the volume values for the scopes are changed, that

means the swing of FFT-Analyzer, Levelmeter, Levelgraph, Quadrascope

depends on the volume settings

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

mastervolume state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g msv - returns actual state (yes/no)

Tooltype:

mastervolume=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.84 modulefault

warn on corrupt modules

If activated, Eagleplayer opens an Info-Requester, when a module fault

occurs e.g. through a harddiskcrash or during net-transfer.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

modulefault state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g mfa - returns actual state (yes/no)

Tooltype:

modulefault=state - state: see above

1.85 Modulesfont

choosing a modulesfont

Opens the fontrequester for setting the font for the modules-menu.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

modulesfont [name.size] - if a name is specified the corresponding font will

be automatically set, otherwise a font-requester

pops up.

status g mof - returns the actual font

Example:

/* small ARexx-Example */

address 'rexx_EP'

modulesfont 'Helvetica.18'

1.86 NextPatt

next Pattern

You can compare a pattern with a notesheet, some soundsystems are based

upon such lists containing commands, which are executed successively. These

commandlists (or notesheets = patterns) can mostly be freely combined. This

function is able to skip to the next pattern in this order within the

module.

Calling it, will result in playing back the next pattern relative to the

current played one. If the last pattern is reached, the first one will be

replayed. Attention: If Songend is activated, a new module will be loaded

(set in Program Mode) or the replay will stop, if Lock Module is

activated.

Keyboard:

-

Menu-Shortcut:

A-x

Arexx:

NextPatt

1.87 NextSong

next Song

There exist musicformats containing more than one musicpart per file (e.g.

Hippel, TFMX, OctaMED) These parts are called "Subsongs", songs within the

soundmodule. This function jumps to the next subsong relative to the

currently played one. If the played subsong is the last, the function has

no effect. If the module only consists of one subsong, the button and the

corresponding menu entry will be disabled.

Keyboard:

-

Menu-Shortcut:

A-v

Arexx:

NextSong

1.88 NoEngine

NoEngine

This Tooltype/Shellstart option disables specified engines from beeing loaded

upon startup. It is intended to be used for temporary disabling visible Engines

in certain situations. Normally you shouldn`t need this option since all states

of windows belonging to Eagleplayer are saved automatically within the main

configuration file.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

-

Shellstart:

NoEngine=names - names: name(s) of the engine(s) not to load, if wanted

to disable more than 1 engine, all names have

to be separated by SPACE and the whole string

has to be quoted

Example: Eagleplayer noengine="FFT-Analyzer Moduleinfo DirListViewer"

Tooltype:

Autosubsong=names - names: see above

1.89 nogui

Using Eagleplayer without user interface

This function, which can only be called via ARexx or Shell/Workbenchoptions

(or by a manul entry in the config), is meant for running Eagleplayer in

the background with a minimum of memory consumption. Please read this

chapter carefully before you try it out, to understand it and to prevent

unnecessary frustration!

If NoGUI is activated, Eagleplayer will not load any user interface in the

future, so the program will be controllable via ARexx only. You can use

this e.g. for custom buttons in Directory Opus or for ARexx-scripts

realised as Toolmanagerdocks.

For preparation you should close all windows except the mainwindow. Then

you should save the configuration, quit Eagleplayer and relaunch it with

the Shelloption or with the Tooltype "nogui=yes". Alternatively you can

do this via Shell by typing rx "address 'rexx_EP';nogui yes;saveconfig". In

this case the nogui-command will be overtaken into the configuration. If

you call the hotkey (normal: CTRL-ALT-E) or the "Show interface"-button in

the Commodities-Exchange a filerequester pop up, where you can select one

or more modules to be played. To quit Eagleplayer in the nogui-mode use

Commodities-Exchange or the ARexx-command Quit.

To switch off the NoGUI-Option, the following ARexx-sequence can be simply

typed into the Shell: rx "address 'rexx_EP';nogui no". Then the BifatGUI

will be opened and you can work with Eagleplayer in the usual manner.

If this sounds too complicated to you, but you want to start Eagleplayer in

the background, then launch it with the Iconify=yes-Shelloption /

tooltype.

example: run >nil: Eagleplayer Iconify=yes

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

NoGUI state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g ngu - returns actual state (yes/no)

Tooltype:

NoGUI=state - state: see above

1.90 notify

Dos-Message

If activated, Eagleplayer watches over the actual directory for changes and reloads in this case the directory automatically.

This function makes only sense in connection with LoadDir.

Attention: In rare cases the Notify-function can collidate with the EPDir-function, where the directory will always be scanned directly from disk. In this case you should switch off EPDir or Notify.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Notify state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g not - returns actual state (yes/no)

Tooltype:

Notify=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.91 NextModule

Next Module

This function forces Eagleplayer to load the next module from the actual modulelist.If the acutally played module is already the last, then the first one will be loaded. This function is disabled in cases where the list consists only of one entry or the 'Load Dir' function is set off.

Attention: If "Random Module" is activated under Program Mode, then the next module will be chosen randomly, independent from the order in the list.

Keyboard:

-

Menu-Shortcut:

A-n

Arexx:

NextModule

1.92 overwrite

Overwrite

If activated, a requester pops up when saving the actual musicfile over an

existing one, so you can cancel this action in order not to overwrite the

other file.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Overwrite state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g owv - returns actual state (yes/no)

Tooltype:

Overwrite=state - state: see above

1.93 previous module

previous module

This function forces Eagleplayer to load the previous module from the

actual modulelist.If the acutally played module is the first one then the

last one will be loaded. This function is disabled in cases where the list

consists only of one entry or the 'Load Dir' function is set off.

Attention: If "Random Module" is activated under Program Mode, then the next

module will be chosen randomly, independent from the order in the list.

Keyboard:

-

Menu-Shortcut:

A-b

Arexx:

PrevModule

1.94 PrevSong

previous Song

There exist musicformats containing more than one musicpart per file (e.g.

Hippel,TFMX,OctaMED). These parts are called "Subsongs", songs within

the soundmodule. This function switches back to the previous subsong

relative to the currently played one. If the played subsong is the first,

the function has no effect. If the module only consists of one subsong, the

button will be disabled.

Keyboard:

-

Menu-Shortcut:

A-c

Arexx:

PrevSong

1.95 password

Change password

If an encrypted file is loaded, Eagleplayer tries at first to decipher it

with the password which is specified here in the requester. If the

deciphering fails, a requester pops up asking for the appropiate passoword.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Password String - String: the new password as string

Arexx:

status g paw - returns the actual password as string or "RESULT" for

no password

Tooltype:

Password=string - string: see above

1.96 Play

Play / Pause

Eagleplayer stops or starts the actual music, dependent on wether he was

already playing or not. Remark: If the player his mode from "pause" to

"play" the music will be replayed from the actual position, not from start,

see also Replaysong. The corresponding button for the play function looks

in the BifatGUI and Eagleplayer1.x-GUI like a tapedeck- or

CD-Player-button and is located nearly in the middle of the control window.

Keyboard:

-

Menu-Shortcut:

A-p

Arexx:

Play

1.97 PrevPatt

Previous pattern

You can compare a pattern with a notesheet, some soundsystems are based

upon such lists containing commands, which are executed successively. These

commandlists (or notesheets = patterns) can mostly be freely combined. This

function is able to skip a pattern in order to go back within the module.

Calling it, will result in playing back the previous pattern relative to

the current played one. If the first pattern is reached, the last one will

be replayed.

Keyboard:

-

Menu-Shortcut:

A-y

Arexx:

PrevPatt

1.98 pubscreen

Pubscreen

This function allows it to change the screen, on which Eagleplayer should

show up. If called from the pulldown-menu a stringrequester opens, where

the complete pubscreenname must be entered, check for Lower- and Uppercase

- its important. Alternatively, the Pubscreenselector can be used instead.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

pubscreen name - Eagleplayer will be opened on the pubscreen named 'name'

Example: pubscreen "DOPUS.1"

Arexx:

status g pub - returns the name of the acutal pubscreen

Tooltype:

pubscreen=name - as described above, but connected with equal-sign

1.99 Playerbatch

Playerbatch

The Playerbatch-function allows it to load the replayers only, if they are

really needed - resulting in reduction of loading time and

memory-consumption of Eagleplayer. More info you can find there.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Playerbatch state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g eab - returns actual state (yes/no)

Tooltype:

Playerbatch=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.100 parentdir

ParentDir

If called, the parent directory relative to the actual one will be read in.

This function makes only sense in connection with LoadDir.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

ParentDir

1.101 protect

Protect module

Some Compression-routines (Powerpacker and Crunchmania), which are offered

by Eagleplayer for saving the actual file, allow you to protect the saved

file with a password. this option activates the passwordrequest during

file-saving.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Protect state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g prt - returns actual state (yes/no)

Tooltype:

Protect=state - state: see above

1.102 protectionbits

Protectionbits

When saving musicfiles, Eagleplayer sets these bits to "read only", so

they can't be overwritten erroneously by the user.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Protectionbits state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g pbi - returns actual state (yes/no)

Tooltype:

Protectionbits=state - state: see above

1.103 programmode

Programming

The settings made here change the behaviour of Eagleplayer at the end of

replaying the actual musicfile. The following 4 settings are possible:

no new module - When a songend is automatically detected,

no new module will be loaded and the replay stops.

previous module - If Songend or PlayTime is activated, the

previous module will be loaded after replaying the

actual one.

next module - If Songend or PlayTime is activated, the next module

will be loaded.

RandomPlay - If Songend or PlayTime is activated, a randomly chosen

module will be loaded. When selecting Next- or Prev-

Module also a random module will be replayed.

At first, all modules from the scanned directory will

be played before loading a module twice. But it's

possible to select a module from the Pull-Down Menu -

the internal counter doesn't recognize this, so that

module can also be replayed again per RandomPlay. The

function Directoryjumps offers another way of random-

play.

This function makes only sense in connection with LoadDir.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

programmode number - number: 0 = no next module

1 = next module

2 = previous module

3 = random module

Arexx:

status g pmo - returns actual state (0...3)

Tooltype:

programmode=number - number: see above

1.104   printtext

Printtext

This ARexx-command forces Eaglepplayer to show a text in the statusline of

the mainwindow (Bifat- or Eagleplayer1.x-GUI).

Arexx:

Printtext string - string: any text

Example:

/* simple Example */

address 'rexx_EP'

printtext 'I l*ve Eagleplayer'

1.105   prefix

Prefix

When Saving the actual musicfile, a prefix will be added in front of the

modulename, if a) the module has an own name (located in the modfile

itself) and b) if the replayer supports this and supplies such a prefix for

the soundformat.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Prefix state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g upr - returns actual state (yes/no)

Tooltype:

Prefix=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.106 QuickStart

Immediate start

The music will be replayed immediatly after loading it, if the menu entry

is activated, otherwise the Play-function has to be used.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Quickstart state - [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g qst - returns actual state (yes/no)

Tooltype:

Quickstart=state - [yes,on or 1]=on, [no,off or 0]=off

1.107 quit

Quit

Do you really think about using this (totally senseless :-) function? It

simply quits Eagleplayer.

Attention: In opposite to old programversions Eagleplayer 2.0 will be

automatically closed when clicking on the close-gadget of the mainwindow.

That's because at least the BifatGUI has a seperate Iconify-Button.

Keyboard:

-

Menu-Shortcut:

A-q

Arexx:

quit

1.108 randomsong

Random Subsong

When playing a musicfile of a soundsystem, which supports several subsongs,

the first played subsong will be choosen randomly. See also here.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Randomsong state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g rns - returns actual state (yes/no)

Tooltype:

Randomsong=state - state: see above

1.109 rescandir

Rescan Directory

By activating this function the actual directory will be completely

rescanned, noticing the state of the Hideall-switch. The settings from

EPDir will be ignored.

This function makes only sense in connection with LoadDir.

Keyboard:

r

Menu-Shortcut:

-

Arexx:

RescanDir

1.110 rasterlines

set rasterlines

In the settings "wait for rasterlines" within the Waitingmode an amount of

rasterlines are count for a timed replay to get the same waitingtime with

on CPUs. The suggested value, which should be entered in the requester, is

8. But remember that the time used for one rasterline is not normed and

depends on the actual screenmode of the Amiga!

A too small value of rasterlines to be waited for can be recognized, if

some notes of the played musicfile are clipped. In this case you should

increase the value or switch to "Automatic wait".

usual values for some screenmodes:

PAL/NTSC/EURO36 - 8...10

DBLPAL/DBLNTSC/MULTISCAN/EURO72 - 18...30

When using graphiccards, the last active amiga-screen before opening the

Cybergraphics-screen is relevant (in most cases PAL or NTSC), see also

here.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Rasterlines number - number: amount of rasterlines to be waited

Arexx:

status g ral - returns actual value as number

Tooltype:

Rasterlines=number - number: see above

1.111 randomstart

Start with random module

If activated, Eagleplayer (when started) doesn't search for the file

specified in the configuration, but chooses another file from the

directory. The advantage is, that you haven't to search a file always by

yourself when starting Eagleplayer and that you will not hear always the

same music on start.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

RandomStart state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g rst - returns actual state (yes/no)

Tooltype:

Randomstart=state - state: see above

1.112 ReplaySong

Replay Song

The loaded music will be replayed from start.

Keyboard:

-

Menu-Shortcut:

A-r

Arexx:

ReplaySong

1.113 savemodule

Save module

Opens the filerequester for specifying a filename for saving the actual

music into. If the music consists of several files (e.g. TFMX) they will be

automatically saved into the same directory. Eagleplayer can save the files

also in compressed form.

This function works only in the registered version of Eagleplayer.

Keyboard:

-

Menu-Shortcut:

A-w

Arexx:

SaveModule [name] - Eagleplayer saves the file under the name 'name' or

opens a filerequester if no name was specified

Example: savemodule "ram:Mod.DMR_folx-vagen"

1.114 Stop

Stop

This function stops the replay of the actual module. If the

Play/Pause-button is then pressed at any time, the music starts from the

beginning.

Keyboard:

-

Menu-Shortcut:

A-s

Arexx:

Stop

1.115 Subsong

Subsong

If the module has several Subsongs, you can specify, which one should be

replayed. A number is expected as argument.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

Subsong Number - Specifying subsong as number

status g snd - returns actual subsong

1.116 Scrollfont

choosing a Srollfont

Opens the font-requester for setting the font for the GUI-scroller of

BifatGUI and Eagleplayer1.x-GUI.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

Scrollfont [name.size] - if a name is specified the corresponding font will

be automatically set, otherwise a font-requester

pops up.

status g scf - returns the actual font

Example:

/* small ARexx-Example */

address 'rexx_EP'

scrollfont 'Helvetica.18'

1.117 ScrollInfos

Scroll information

If this menu-entry is activated, all information texts will be scrolled

into the mainwindow, otherwise simply print into the window.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

scrollinfo state - [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g sci - returns actual state (yes/no)

Tooltype:

scrollinfo=state - [yes,on or 1]=on, [no,off or 0]=off

1.118 Songname

Songname

If this menu-entry is activated, not the name of the loaded file will be

shown in the titlebar / scroller, but the internal name of the music found

in the file. This is only possible if the musicformat supports that (e.g.

Protracker, (OctaMED, Screamtracker).

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

songname state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g sna - returns actual state (yes/no)

Tooltype:

songname=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.119 scanalways

Always Scan Directory

If activated, everytime when loading a musicfile via filerequester, Arexx

or an external engine the entire directory will be rescanned. Normally,

this function should stay disabled. But when you activate it, you can at

the very least switch off EPDir.

This function makes only sense in connection with LoadDir.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

ScanAlways state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g alw - returns actual state (yes/no)

Tooltype:

ScanAlways=state - state: [yes,on or 1]=on, [no,off or 0]=off

1.120 songend

Songend

If a songend is detected by the player, then the replay stops (if set so)
and, depending on the programmode-settings, a new file will be loaded or it
will be waited for user input. You can consult the Manager for the
information, which players support this songend-function.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Songend state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g end - returns actual state (yes/no)

Tooltype:

Songend=state - state: see above

1.121 savedir

set directory for saving

With this function you can set the directory, which will be automatically
preset in the filerequester used by Eagleplayer for saving the actual file
into. However, the main reason for this function is due to the existence of
the Automatic saving - option.

Remark: If the user sets another path in the filerequester during a manual
save, this path will be used from then on.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

SaveDir [path] - path: either a directorypath or a requester pops up

Arexx:

status g sap - returns actual path

Tooltype:

SaveDir=path - path: see above

1.122   safesave

Safe saving

If activated, Eagleplayer at first saves the actual file under a different

name and renames it only then, if the file was really completly written to

the medium. This avoids an overwriting of an existing file especially in

cases, where the writing process fails due to media-errors, which normally

result in data loss.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

SafeSave state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g sfs - returns actual state (yes/no)

Tooltype:

SafeSave=state - state: see above

1.123   samplemode

Samplemode

Some of the compression-methods (Powerpacker, Crunchmania and LH.library),

offered by Eagleplayer for saving the actual file, can activate a

soundmodule-specific, a so called delta-mode, in order to gain much more

compression in most cases, than a normal packmode. And exactly this feature

this switch activates.

Remark: We have inofficially introduced the samplemode for Powerpacker, so

Eagleplayer is actually the only program, which can load such packed files.

A decrunch-slave for the xfd.library from Georg Hoermann is in work.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Samplemode state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g smo - returns actual state (yes/no)

Tooltype:

Samplemode=state - state: see above

1.124 saveasprotracker

save as Protracker

Mainly, this option was included for the internal Eagleplayer-replayers
(Player5.x/ 6.x, Trackerpacker, Noisepacker etc). If activated, these
modules will be saved in the normal Protracker-format, which can be loaded
and manipulated by the wellknown trackers like Protracker or OctaMED.
Programs like Noiseconverter or Prowizard aren't needed anymore, especially
in connection with the Eagleexotic.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

SaveasProtracker state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g spt - returns actual state (yes/no)

Tooltype:

SaveasProtracker=state - state: see above

1.125 softint

Softinterrupt

This option was implemented to avoid negative interactions with other
timecritical programs (e.g. Modem-Usage). Because the CIA-Interrupt has
got a higher priority then the one for the modem-usage, transfer errors can
occur. To minimalize this danger a so called soft-interrupt will be started
right after the CIA-interrupt call. This softinterrupt will only be run if
requests from the serial interface occur. If there still should be problems
(e.g. with slow CPUs) then you may reduce the serial baudrate in your
terminalprogam.

When using "timer.device" for timing purposes this switch has got no

effect.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Softint state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g sit - returns actual state (yes/no)

Tooltype:

Softint=state - state: see above

1.126 speed

Speed

This slider stands for the replay-speed of the played musicfile. If it is

in middle position, the normal speed is set, moving it up increases the

speed, moving it down decreases it. If the actual replayer should not

support this speedcontrol, the corresponding sliders and menu-items will be

disabled.

Some replayers additionally offer a pitch up/down (changing the

tone-heigth). The resulting effect sounds like in/decreasing the speed of a

record player. (e.g. Protracker, AIFF, Oktalyzer, David Whittaker).

Using the menu, the speed can be in/decreased by one step and reset to the

normal position.

Keyboard:

-

Menu-Shortcut:

A-+ - increasing speed by one step

A-- - decreasing speed by one step

A-# - normal speed

Arexx/Shellstart:

Speed number - number: -25...0...25 (-25=slow;0=normal, 255=fast)

Arexx:

status g spe - returns actual value (-255...255)

Remark: Because Arexx has got problems with

negative numbers, the returned numbers

for values <0 go from 65535 (=-1) to

65511 (=-25), which have to be re-

calculated by the Arexx-program.

Tooltype:

Speed=number - number: see above

1.127   slowerspeed

Slower speed

The Replay speed will be decremented by one step.

Keyboard:

-

Menu-Shortcut:

A--

Arexx:

Slowerspeed

1.128   Status

Status

With the help of the status-command you can eaxamine Eagleplayers extensive

settings- possibilities. Following an overview over the corresponding

arguments. They can also be found in the docs of the commands itself.

Arexx/Shellstart:

Status <group> <dependent>

Returns the actual status. You have to differ between the following groups:

- Globale settings (g),

- Modul-dependent settings (m)

- Player-dependent settings (p)

- SampleInfo (s) (not yet implemented !!!)

- UserPrograms (u)

Depending on the group different arguments are expected. Below you will

find all status commands of the Eagleplayer

The global settings

----

status <group> <dependent> ;<group> = g

;<dependent> = <cmd>

<cmd> Meaning and Return value

----

ver Eagleplayer Version Zahl

dve DeliTracker Version Number

dir Playerdirectory String

udi Userprogramdirectory String

fon actual Font String

hep Help Path String

ins Instruments directory String

sap Save Path String

pbp PlBatchPath String

ubp UPrgBatch Path String

pke HotKey String

num Number of players Number

ico Iconify on <Yes|no>

mwi Main-Window opened <yes|no>

pli Playerwindow opened <yes|no>

vol actual volume Number

bal actual balance Number

spe actual Speed Number

ply Song is playing <yes|no>

hel HelpMode <yes|no>

pub actual public screen String

mau Max. Userprograms Number

usp loaded Userprograms Number

snd actual SongNumber Number

mil Number of Modules in List Number

epb Eagleplayer Globals Address

vo1 Voice1-state <yes|no>

vo2 Voice2-state <yes|no>

vo3 Voice3-state <yes|no>

vo4 Voice4-state <yes|no>

eab EP-Batch-State <yes|no>

ejp EjectPlayer-state <yes|no>

flp FlashPointer <yes|no>

msv MasterVolume <yes|no>

sna Use Songname <yes|no>

qsf Quit Surface <yes|no>

que Quit Eagle <yes|no>

sat SaveT <yes|no>

sci Scroll-Infos <yes|no>

led Filter-state <yes|no>

fid Fadein-state <yes|no>

fad Fadeout-state <yes|no>

aps AutoPasswort-state <yes|no>

ldd Load Dir-state <yes|no>

aud Auto Dir-state <yes|no>

sdn Show Dirnames-state <yes|no>

hia HideAll-state <yes|no>

uep Use EPDir-state <yes|no>

not Notify-state <yes|no>

alw Scan Always-state <yes|no>

mme Size of MenuMem Number

end Songend-state <yes|no>

lbf Load Before-state <yes|no>

law Load Always-state <yes|no>

qst Quickstart-state <yes|no>

rns RandomSong-state <yes|no>

ass AutoSubSong <yes|no>

tim TimeOut Number

ovw Overwrite-state <yes|no>

ats AutomaticSave <yes|no>

ssd SS Dir <yes|no>

prt Protect Module <yes|no>

pbi Protection Bits <yes|no>

smo SampleMode <yes|no>

spt Save As Protracker <yes|no>

smo Small Module <yes|no>

bmo Big Module <yes|no>

mof Module fault <yes|no>

icm actual IconifyMode String

frm actual Filerequester String

Module dependend settings

----

status <group> <dependent> ;<group> = m

;<dependent> = <cmd>

<cmd> Meaning and Return value

----

pat ModulePath String

dir Module directory String

fil Modulename String

pnr Number of active Player Number

num Position in Module-list Number

min mix. Number of SubSongs Number

max max. Number of SubSongs Number

aut Authorname String

son Songname String

len PatternLength Number

dpt different Patterns Number

stp Steps Number

sam Samples Number

chk CheckMod (mod ok?) <yes|no>

pon PositionNr Number

icm actual Playernumber Number

dur calculated playtime Number

siz entire module length Number

Player dependend settings

----

status <group> <dependent> ;<group> = p

;<dependent> = <playernum> <cmd>

<cmd> Meaning and Return value

----

nam Playername String

cre Creatorname String

pin PlayerInfo String

ver Versionnumber Number

vol Volumecontrol available <yes|no>

bal Balance control available <yes|no>

jmp Patternjump available <yes|no>

pos GetPositionsNr <yes|no>

spd Playspeed changeable <yes|no>

sub Subsong available <yes|no>

cfg Player can be configurated <yes|no>

pck Supports Packed Mods <yes|no>

upr Userprogram support <yes|no>

csa Config saveable <yes|no>

clo Config loadable <yes|no>

msa Module saveable <yes|no>

mii ModuleInfo possible <yes|no>

sai SampleInfo possible <yes|no>

msm Max. Samples Number

msy Max. SynthSamples Number

mle Max. Length Number

mpa Max. Pattern Number

mst Max. Steps Number

mso Max. SubSongs Number

rst Player restartable <yes|no>

sge Player supports Songend <yes|no>

voi Voicescontrol available <yes|no>

plf Player supports PlayFaster <yes|no>

mch Module change <yes|no>

mrs Module restore <yes|no>

dat Creator-Date String

tgl Taglist of Player Number

the engine dependend settings

----

status <group> <dependent> ;<group> = u

;<dependent> = <UserNr> <cmd>

<cmd> Meaning and Return value

----

nam Name of UserPrg String

abo About Userprogram String

cre Creator of Userprg String

ver Version of Userprg Number

rev Revision of Userprg Number

eve min. EP-Version Number

dat Creation-Date String

kis Kickstart UserPrg needs Number

prz Processname of Userprg String

win Window opened <yes|no>

con Config of User-Prg Number

kox X-coordinate of Window Number

koy Y-coordinate of Window Number

1.129 smallmodule

Warn on too short modules

If the option is activated and a too short module should be played next,

Eagleplayer opens a requester, which informs about that. Some soundsystems

(e.g. Deltamusic2) can produce heavy problems with too short modules.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Smallmodule state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g smo - returns actual state (yes/no)

Tooltype:

Smallmodule=state - state: see above

1.130 scrolltext

Scrolltext

This ARexx-command forces Eagleplayer to show a text in the statusline of

the mainwindow (Bifat-or Eagleplayer1.x-GUI). However, the

Scrollinfos-option must activated for this.

Arexx:

Scrolltext String - String: any text

Example:

/* simple Example */

address 'rexx_EP'

scrollinfos yes

scrolltext 'I l*ve Eagleplayer'

1.131 Save Configuration

Save Configuration

This function opens the filerequester for specifying the configfile-name

and saves all settings of the player into that file. The settings contain

all positions, sizes and states of the mainwindow, engines, replayers,

special settings of replayers/engines, the actual module-dir and which

music is momentally played..

Hint: If you don't want a module to be replayed automatically on startup

but a specific module directory being available after startup, then change

into that directory, eject the replayed module, set Randomstart to off,

and save the configuration.

Keyboard:

-

Menu-Shortcut:

-

Arexx:

SaveConfig [path+name] - if a path/name is specified, the config will be imme-

diatly saved without filerequester under that name

1.132 togglefilter

Toggle Filter

Switches on/off Amiga's low-pass-filter. That means if the filter was off

before, it will be activated and vice versa.

Keyboard:

-

Menu-Shortcut:

A-f

Arexx:

togglefilter

1.133 timemode

Timemode

This function overruns the settings for the maximal Playtime at

soundsystems, which automatically detect the end of a musicfile. Result:

One the one hand some musicfiles won't run endlessly, on the other hand the

remaining musicfiles won't be interrupted in the middle of the replay.

This function makes only sense in connection with LoadDir and with an

activated Songend.

see also min. Playtime

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Timemode state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g tmo - returns actual state (yes/no)

Tooltype:

Timemode=state - state: see above

1.134 timeout

max. Playtime

If you specify a maximum playtime, the replay will be stopped exactly after

this timespan and the prog will continue depending on the settings for

Programmode. The value has to be given in seconds via a requester. A value

of "0" disables this function!.

see also min. playtime and Timemode

Keyboard:

-

Menu-Shortcut:

A-z

Arexx/Shellstart:

Timeout number - number: max. playtime in seconds (0=off)

Arexx:

status g tim - returns actual state (in seconds)

Tooltype:

Timeout=number - number: see above

1.135 Voice

Voices

This function allows a seperate switching on/off of all voices of the

musicfile. Standard-Amigamodules consist of 4 voices, multichannel-modules

can have up to 32 (or more) voices, which however are not switchable (yet).

If the button looks "pressed" (so it differs in the borders from the

others) the corresponding voice is off, otherwise on; a checkmark in the

menus means an active voice.

Keyboard:

-

Menu-Shortcut:

A-1

A-2 Switch on/off voices 1 - 4

A-3

A-4

A-d normal setting, all voices are on with full volume

Arexx:

voice x y - x=voicenumber (1...4),y=number for volumevalue from 0(=off)

to 255(=max.)

status g voi x - x=voicenumber (1...4), result is a value between 0 = off

and 255=max for the volume

1.136   timingmode

Timing

The timingmode defines, how the replay of the musicfile will be timed. At
the moment there are two main timing-possibilities: CIA and VBlank. The
CIA-Timer of the Amiga works much more exactly and in all machines with the
same speed. The VBlank however (=Vertical Blanking Interrupt) depends on
the actual scanrate of the OCS/ECS/AGA-Amigas. Normally, CIA-Timing should
be set. There are only a few (old) modules, which require VBlank
(mod.FaceAnotherDay, mod.Techdust). At this point we mention the
Noiseconverter (a commercial for us), which can fix such modules with the
"correct tempo" - function.
If the CIA-Timer should be used by another program yet and is therefore not
available, an error message pops up.
From V2.0 of Eagleplayer on we added the option of using timer.device.
(default setting) Due to the fact that the program no longer uses the
hardware interrupts directly some problems in conjunction with other
programs and hardware extensions should be solved. On the other hand
this mode needs a little bit more CPU time caused by the timer.device
overhead.
If you even now don't understand the thing behind these options, please do
not touch them. With the preset options no problems raised until now since
Eagleplayer 1.53 . (A500-A4000/60)
see also Waitingmode
Keyboard:
-
Menu-Shortcut:
-
Arexx/Shellstart:
Timingmode number - number: the mode to be used
0 = CIA
1 = VBlank
Arexx:
status g tmd - returns actual state as number,
0 = CIA
1 = VBlank
Tooltype:
Timingmode=number - number: see above

1.137 volume

Volume

Controlling the volume of the replayed file. If the actual replayer should

not support volume-control then this slider and the corresponding

menu-items will be disabled.

The BifatGUI has 2 extra buttons for the volume. After the first click the

left one sets the volume to 0 (mute-function) and when selected again, to

the last value. The right one does this analogue, but after the first click

the volume will be set to the maximum value.

Via the menu you can decrease/increase the volume by one step and the

resetting to the maximum volume.

Keyboard:

-

Menu-Shortcut:

A-6 = decrease volume by 1

A-7 = increase volume by 1

A-5 = set volume to the preset value (=Maximum)

Arexx/Shellstart:

Volume Number - Number: 0...255 (0=off, 255=maximum)

Remark: Eagleplayer calculates internally with

256 volume steps, but offers the user

actually a range from 0..64 as value,

to go conform with the Amiga-hardware-

restrictions)

Arexx:

status g vol - returns actual value (0...255)

Tooltype:

Volume=number - number: see above

1.138 waitingmode

Waitingmode

Due to the work of the Amiga-hardware it is neccessary to wait a specified

timespan, when starting a new playback of datas by the soundchip paula.

Eagleplayer offers therefore some possibilities. Normally, the option

"automatic wait" is suggested, which uses the best method when needed. When

using some of the Amplifiers (8 Bit, 14 Bit Amplifier), these settings are

relevant for the replay and will be ignored. The waiting for the

audiohardware is not needed in that case.

Here are the options

() Automatic wait - uses CIA-timer, when possible, if not, a specified

amount of rasterlines wil be waited

() wait for rasterlines - dependend on settings, several rasterlines

will count, before the replay continues

() wait with DBF-loops - this function was only implemented

for 68000er-CPU-Amigas. By this setting simply a

processor-waitingloop will be run, which we cannot

recommend anymore for CPUs with cache.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

Waitingmode number - number: Mode to be used

0 = automatic

1 = Rasterlines

2 = DBF-Loops

Arexx:

status g wam - returns actual state as number,

0 = automatic

1 = Rasterlines

2 = DBF-Loops

Tooltype:

Waitingmode=number - number: see above

1.139 **xfddecrunch**

Depack files with XFD

XFD is a program library from Georg Hoermann, which is able to detect most

of the existing amiga-crunchers and its data in order to depack it.

Although Eagleplayer is able to detect and operate the most used packers,

the option was built in to use XFD for furhter crunchmodes.

(very practicable when using the engine Eagleexotic.)

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

XFDDecrunch state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g xfd - returns actual state (yes/no)

Tooltype:

XFDDecrunch=state - state: see above

1.140 xfdloadseg

Depack executables with XFD

If activated, Eagleplayer also tries to depack executables via XFD and to

use them. So you can pack all players and engines onto the harddisk (e.g.

with Powerpacker), if there shouldn't be enough space left. This ofcourse,

will increase Eagleplayers loadingtime.

Attention! If you want to activate this function, then make sure that the

XFD-slaves `XFD_FIMP_LHLib_XPK.Slave` and `XFD_ATN!_LHLib_XPK.Slave` are

NOT! placed in your Libs:xfd/ - directory. These XFD- slaves were

testversions from us, which were spread because of a misunderstanding. They

caused big problems when using this XFD-option for executables. (Massive

enforcer-hits -> loading Eagleplayer1.54 on some machines took several

minutes). The reason, because this problem occured only when using

Eagleplayer, is, that Eagleplayer isobviously the only program which uses

this XFD-feature.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

XFDLoadseg state - state: [yes,on or 1]=on, [no,off or 0]=off

Arexx:

status g xfl - returns actual state (yes/no)

Tooltype:

XFDLoadSeg=state - state: see above

1.141 xpkpackmethod

Change XPK Packer

When calling this function, a requester pops up asking you for a

4-letter-index, which stands for the xpk-library, which should be used for

saving the actual musicmoule.

Examples: SQSH, CRMS, SMPL, GZIP, MASH, NUKE

For packing musicfiles with samples (e.g. Protracker, OctaMED) we suggest

you to use SQSH or CRMS, which offer brilliant packrates. For us, SQSH is

faster and works better.

Keyboard:

-

Menu-Shortcut:

-

Arexx/Shellstart:

XPKPackmethod Name - Name: 4-lettered-name of the XPK-packer

Arexx:

status g xpk - returns actual packername, a String with 4 letters

Tooltype:

XPKPackmethod=Name - Name: see above