

Eagleplayer

COLLABORATORS

	<i>TITLE :</i> Eagleplayer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 30, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Eagleplayer	1
1.1 Eagleplayer 1.52 Documentation	1
1.2 copyright und vertriebsbedingungen	4
1.3 distribution	5
1.4 was ist und kann der eagleplayer ?	6
1.5 entstehung des eagleplayers	8
1.6 was fehlt in dieser version noch ?	8
1.7 wann gibt es eine neue version des eagleplayers ?	9
1.8 update-service	9
1.9 fehlerbeschreibung	10
1.10 keyfile-benutzung	10
1.11 systemanforderungen	11
1.12 benötigte zusätzliche dateien	11
1.13 installation auf diskette,festplatte,...	12
1.14 start des eagleplayers von cli & shell	13
1.15 start von der workbench aus	13
1.16 das konfigurationsfile	13
1.17 das hauptfenster	14
1.18 steuerung des programms über die schalter (gadgets)	15
1.19 steuerung über die pulldown-menüs	17
1.20 tastaturkürzel	25
1.21 die appwindow-funktion	26
1.22 pubscreens	26
1.23 font-unterstützung	26
1.24 interne depacker	27
1.25 der lha-extraktor	27
1.26 öffnung und funktion des player-fensters	28
1.27 steuerung über die schalter und erklärung der funktionen	28
1.28 tastaturkürzel im playerwindow	28
1.29 interne player	29

1.30	einleitung zu den userprogrammen	31
1.31	steuerung der user-programme	31
1.32	zu den einzelnen bisher verfügbaren userprogrammen	32
1.33	keine programmierrichtlinien für userprogramme	33
1.34	einbindung des eagleplayers in andere programme	34
1.35	die playerbatch-datei	36
1.36	die userprogrammbatch-datei	37
1.37	fehlermeldungen	37
1.38	fehler des programms	38
1.39	problemecke	39
1.40	einleitung zu den externer playern	40
1.41	das externe player konzept	40
1.42	schematischer aufbau von externen playern	41
1.43	schematischer aufbau von custom modulen	41
1.44	anpassung von playern	41
1.45	playerheader	42
1.46	modulerkennung	42
1.47	interrupts	43
1.48	bedeutung der tags	43
1.49	delitracker support funktionen	47
1.50	tips zur anpassung	49
1.51	die besonderheiten und neuerungen des eagleplayers	50
1.52	die neuen tags	51
1.53	moduleinfo	52
1.54	analyzeransteuerung	54
1.55	die neuen eagleplayer-globals	56
1.56	einführung und beschreibung des arexx-portes	60
1.57	arexx-kommandoübersicht	60
1.58	anpassung an directory opus	68
1.59	danksagungen	69
1.60	registration	71
1.61	registrierkarte	71
1.62	updatekarte	72
1.63	biographie	73
1.64	externe player	77
1.65	fehler in den playroutinen	94

Chapter 1

Eagleplayer

1.1 Eagleplayer 1.52 Documentation

Manual
for the
EaglePlayer
Version 1.52
Manual \$VER: V1.08 (April-28-94)

© 1993 Defect Software Productions
external Playerconcept © Delirium, expanded by Defect

Table of contents:

1. Copyright
 - 1.1 Copyright and distribution-conditions
 - 1.2 Distribution
 2. Introduction
 - 2.1 What is the EaglePlayer (able to do) ?
 - 2.2 Formation of the EaglePlayer
 - 2.3 What is not yet implemented in this Version ?
 - 2.4 When will the next EaglePlayer come up ?
 - 2.5 How to become a registered User
 - 2.6 Orderform
 - 2.7 Update-Service
 - 2.8 Bug reports
-

-
- 2.9 Using the Keyfile
 - 3. Installation and starting
 - 3.1 System requirements
 - 3.2 Additionally needed files
 - 3.3 Installation on Disk,HardDisk,...
 - 3.4 Starting the EaglePlayers from CLI/SHELL
 - 3.5 Starting from the Workbench
 - 3.6 The Config-file
 - 4. controlling the Program
 - 4.1 The main window
 - 4.1.1 Controlling the program using the Gadgets
 - 4.1.2 Control via Pulldown Menus
 - 4.1.3 Shortcuts
 - 4.1.4 The Appwindow-function
 - 4.1.5 Pubscreens
 - 4.1.6 Font-support
 - 4.1.7 internal (Un)Packer
 - 4.1.8 The LHA-Extraktor
 - 4.2 The Player-window
 - 4.2.1 Opening and Functions of the Player-Window
 - 4.2.2 Controlling using the buttons and description of it's ↔ functions
 - 4.2.3 Shortcuts in the Playerwindow
 - 4.2.4 Internal Players
 - 4.2.5 External Players
 - 4.2.5 Bugs within the players
 - 4.3 The User-Programs
 - 4.3.1 Introduction
 - 4.3.2 Controlling the User-Programs
 - 4.3.3 to the Userprograms currently available
 - 4.3.4 No coding guides for User-Programs !
-

-
- 4.3.5 Binding the Eagleplayer into other programs
 - 4.4 The Batch-files
 - 4.4.1 The PlayerBatch-file
 - 4.4.2 The UserprogrammBatch-file
 - 5. Bugs
 - 5.1 Error Warnings
 - 5.2 Errors of this Program
 - 5.3 Problem-sheet
 - 6. EXTERNAL PLAYER PROGRAMMING INTERFACE
 - 6.1 Introduction to the external players
 - 6.1.1 The external Player concept
 - 6.1.2 The structure of external players
 - 6.1.3 The structure of custom modules
 - 6.2 How to build external players
 - 6.2.1 Playerheader
 - 6.2.2 Modulerecognition
 - 6.2.3 Interrupts
 - 6.3 Meanings of the Tags
 - 6.4 DeliTracker support functions
 - 6.5 Hints and Tips
 - 6.6 The special informations and news for the Eagleplayer
 - 6.6.1 The new tags
 - 6.6.2 Moduleinfo
 - 6.6.3 Analyzer control
 - 6.6.4 The new Eagleplayer Globals
 - 7. The Arexx-Interface
 - 7.1 Introduction and description of the Arexx-Port
 - 7.2 The commands
 - 7.3 Arexx-Adaption to DOPUS 4
 - 8. History
 - 9. Acknowledgements
-

9. Greetings and Acknowledgements

1.2 copyright und vertriebsbedingungen

1.1 Copyrights and distribuiton-conditions

~~~~~

The EaglePlayer is to be seen as Shareware in the current Version. That means, that we expect a ShareWare-fee of 30 US\$ outward of germany (else 20 DM), that has to be given CASH or as Eurocheque (other currencies or Cheques can NOT be accepted). In return for that you will get the latest version of the EaglePlayer, which contains features, that are disabled in the ShareWare version (e.g. Save, Tooltypes-Support). Besides that you will get the improved version including new UserPrograms for free, so that the Shareware-fee is quite justified considering the mailing costs.

Please send the supplied  
Orderform  
to:

(read chapter

Registration  
first)

Eagleeye

Buggs

-----

-----

|                        |                            |
|------------------------|----------------------------|
| Jan Blumenthal         | Henryk Richter             |
| Erich Mühsam Straße 33 | or Stephan-Jantzen-Ring 47 |
| 18069 Rostock          | 18106 Rostock              |
| Germany                | Germany                    |

Phone/FAX: +49 / 0381 / 83979

Phone: +49 / 0381 / 1208867

E-Mail: [dtc@informatik.uni-rostock.de](mailto:dtc@informatik.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. NO DISK, NO ANSWER !!! We just can't send away (nearly) thousands of Disks. Send us a Disk and you will get the latest Demo- or the according registered version back.

The EaglePlayer may be distributed on Public-Domain-Series, ShareWare-Series and Networks under the following conditions :

- The EaglePlayer-Package must remain complete, the files belonging to that package are listed in the 'Distribution' chapter.

The EaglePlayer is going to be delivered on 2 Disks (uncrunched).

- Neither the program, nor the documentation may be changed, except for packing them, as long as the original state will be 100% recoverable (e.g. LHA, ZOO, ARC, ZIP, PowerPacker).
- The distribution price for the EaglePlayer itself or for the Disks containing the EaglePlayer may not exceed the total of 5 DM or the



corresponding amount in other currencies. The same applies to transferring-fees in Networks.

- The firm "Mallander Software" may NOT distribute this software !!!
- The EaglePlayer may not be distributed in Software-Packages, excepted PD- or Shareware-Series (e.g. Fish, Time), without any WRITTEN permission of the authors.  
This permission has to be gathered BEFORE DISTRIBUTING it !!!!!!!!!!!
- That applies for commercial use or distribution as well, what is explicitly prohibited from the authors without having a WRITTEN permission.

We, the authors of this program, will not take any liability for damages on hard- and software that were eventually caused by the program. The use of this program is at your own risk.

WE, THE AUTHORS, PRESERVE TO THE RIGHT OF PROHIBITING THE DISTRIBUTION OF THE EAGLEPLAYER TO SINGLE VOLUNTEERS, IN CASE THEY DO NOT FOLLOW OUR COPYRIGHT-CONDITIONS. The EaglePlayer was not written for people, who wanna earn much money with it. It shall expand the Amiga-World with a senseful music playing program that only requires little money to get.

## 1.3 distribution

### 1.2 Distribution

-----

The Eagleplayer is a program package which is distributed on 2 disks. The further distribution of this package is only allowed when the following files and directories (with their contents) are supplied in unmodified form. When you change the Names of the Discs (e.g. for a Series like Time, Fish) you have to change the Paths in the "Install"-Script as well as in the files "Deutsche\_Anleitung" and "English\_Documentation" !

```
Eagleplayer_Disk_1:
  Locale (dir)
    Catalogs (dir)
      deutsch (dir)
        Noiseconverter.catalog
  Icons (dir)
    Eagleplayer.Doc.Guide.info      Eagleplayer.Dok.Guide.info
    Eagleplayer.info
  c (dir)
    Echo
    LhA
    MuchMore
  Prefs (dir)
    Env-Archive (dir)
      EaglePlayer (dir)
        Eagleplayer.Config          Eagleplayers.Batch
        Userprograms.Batch
  Eagleplayers (dir)
    FC 1.3                          FC 1.4
```

```

    Jamcracker
    SonicArranger
EP-UserPrograms (dir)
    DirListViewer
    Quadrascope
    Time
libs (dir)
    XFD (dir)
        XFD_FIMP_LHLib_XPK.Slave
    Compressors (dir)
        xpkFAST.library
        xpkLHLB.library
        xpkRDCN.library
        xpkSQSH.library
    amigaguide.library
    diskfont.library
    icon.library
    powerpacker.library
    reqtools.library
    xpkmaster.library
Utilities (dir)
    AmigaGuide
    Installer
Deutsch.info
Deutsche_Anleitung
Deutsche_Anleitung.info
Disk.info
Eagleplayer
Eagleplayer.info
English.info
English_Documentation
English_Documentation.info
Icons.info
Install
Liesmich_SOFORT.Dok
Liesmich_SOFORT.Dok.info
Noiseconverter
Noiseconverter.info
Readme_NOW.Doc
Readme_NOW.Doc.info

Eagleplayer_Disk_2:

    ArexxExamples.lha
    Disk.info
    Eagleplayers.lha
    Userprograms.lha

    Oktalyzer
    TFMX Pro

    Patternsroll
    Space-scope

    xpkIMPL.library
    xpkNUKE.library
    xpkSMPL.library

    commodities.library
    explode.library
    lh.library
    req.library
    xfdmaster.library

    AmigaGuide.info

```

## 1.4 was ist und kann der eagleplayer ?

### 2.1 What is EaglePlayer (able to do) ?

-----

The Eagleplayer is a program-package to comfortably play music of different sound-formats .It is NOT usable to create music-modules or songs,and should

---

therefore not be mixed up with a music-editor, sequencer, tracker etc. It actually includes the play-routines for Sound/Noise/Protracker, Startrekker (AM) & AudioSculpture (AM) as well as the internal customplay-port as already known from DeliTracker1.3 and is able to load furthermore needed players after. The advantage of this method is as the following:

You do only have to load the actually needed players, what saves loading-time and memory. Besides that, the length of the main program will not increase to infinite length. All internal SoundTrackerRePlayers support Song-Loading. The Songs and Samples may be packed. The Instrument-Path may be adjusted via Config (chap. 4.2.2). The Samples may be packed. They are all loaded as RAW data.

Following some features of the Eagleplayer

- built up modular
- the number of loadable players is only limited by the amount of free memory (chap. 3.4)
- supports AppWindow in all its intern windows (Kick2.0+)
- 100% Assembler coded
- alphabetical order of players in the Playerwindow
- a AddPlayerDir function
- a DeleteAll function
- players can be enabled/disabled
- possibility of loading up to 32 external
  - Userprograms
    - (e.g. Analyzerprograms)
- own graphical interfaces can be programmed and used as Userprograms
- the choice of modules to be played can be done directly via Pulldown-Menues (chap. 4.1.2)
- comprehensive possibility of configuration (FadeIn, LoadNext ...)
- speed-regulation via Proportional Gadgets (chap 4.1.1)
- all voices can be separately enabled/disabled
- status report via scroller
  - carefully and extensively adapted
    - playing-routines
    - , nearly all
- operate on an A4000 (see Player.Doc)
- 7 intern
  - Depackers
    - + XPK + XFDMaster
- is able to use the DeliTracker-Players, if the corresponding player adapted to the EaglePlayer is not present
- extensive ModuleInfos, if the Replayer has been adapted appropriately
- ONE version for all Kickstarts (from Kick1.2 upward)
- gadgets automatically accomodate to the color-palette when opening a window (3D-look)
- modern styling (we think)
- has an Iconify-Mode (AppItem, AppIcon, LittleWin)
- internal SoandTrackerPlayers are able to load songs
- a PlayFaster-Gadget
- more Arexx-commands (above all, status-commands)
- a Randomsong/AutoSubSong-function
- "load on demand system" for external replayers {"(see)" Link "Die PlayerBatch- ↔ Datei"}
- "load on demand system" for userprograms
  - (see)
  -
- LHA-Extractor

- different filerequesters (ASL,Req,ReqTools,None)
- Fadein/fadeout of the music
- complete configuration including open windows, their position, actual directory, actual Module, Packmode etc. as an ASCII-File
- able to play a random chosen module automatically after starting the program.
- duration calculator for certain soundsystems

A special advantage of the EaglePlayer in contrast to many rivals is, that it works with most of the

playing-routines

(those not

working, see

Problems

) and plays the modules always

at same speed, independent to the picture-rate, thus under PAL as fast as under NTSC and Productivity modes.

Due to the modular system the Eagleplayer may be adapted in the best way for every computer system and the user's wishes.

## 1.5 entstehung des eagleplayers

### 2.2 Formation the EaglePlayer

-----

It began with the Phenomena Demo "Enigma". There we found a module converted with the Noisepacker. We ripped the music but hadn't got any player for this format. A tool for playing this module was written. Some time later some other formats were included (Noisetracker...). This program was called "Megaplayer" and was released in April 1992. It was able to play 16 different formats.

This program grew up to about 25 and got more and more confused.

About one year ago Jan decided to write a new and competent Soundplayer-Program, because people were quite unhappy with his Noiseplayer-like Megaplayer. He industrially worked on it, created an attractive graphical interface, tried to fetch as many playing-routines as possible and to integrate them in the program. (we also developed a concept for external replayers)

Due to the fact that the Port developed by Delirium was better we included this port into our program. Later our port was extended so far that both programs are no more compatible to each other.

We still wanted more. So we started to code a port for extern Modules for Analyzing purposes. At this point we wanted to write a documentation for everyone to program his own Analyzer tool, so we called the proggies "Userprograms". Later this port grew up so much that we thought nobody could understand it fully and we gave up to write a documentation. Nowadays this port is a univereal one for Analyzers, Public Screen Managers, Pattern-scrollers, even extern Programs may call the Eagleplayer. (not only via Arexx)

## 1.6 was fehlt in dieser version noch ?

### 2.3 What is still missing in this version ?

---

A lot, till today the following functions are planned for the near future

- \* Sampleinfo (with saving the samples, playing them and so on)
- \* UserProgram Manager

Comment: The functions marked with a '\*' will probably only be implemented in the registered version ! Note that you will get the topic version of the EaglePlayer, if registered. That may mean under circumstances, that some features could be missing, but were planned to be implemented and could not get finished because of time problems. These parts of the program may then be used with a KeyFile in later versions.

## 1.7 wann gibt es eine neue version des eagleplayers ?

### 2.4 When will new versions come up ?

---

That's hard to say. It may take a little longer getting the new versions ready because we, Eagleeye and Buggs, are recently only at home on weekends. Users who want to be registered are herewith asked of being patient in getting an answer from us.

## 1.8 update-service

### 2.7 Update-Service

---

Every registered user may order the latest version without any fee. Just send 2 Disks with back porto to us. (Don't forget to send your keyfile on one of these disks) Please send also your Configuration file and the Eagleplayer/ Userprogram Batchfiles.

Furthermore we expect the

Update Form  
to get from

ya. Don't forget to PUT IN YOUR USERNAME - you find it when you select the "About"-Item in the Project Menu and then click on "Key".

Please answer the following questions:

- Are you satisfied with the delivery of your copy ?
  - Did you got problems with specific Machine configurations ?
  - Did you expect more or less from your registered Version ?
  - Is the Documentation sufficient ?
  - Which functions do you wanna to be included ?
  - Does the Eagleplayer work together with all of your programs ?
  - Do the playroutines work correctly ?
  - Are your friends using the Eagleplayer as well as you ? When not, why ?
  - Do you think the Eagleplayer needs to much memory ?
-

- (don't know the Batch functions, hey ?)
- Do you think the Eagleplayer needs to much CPU-Time ?  
(look at a certain opponent in Version 2.0, hehehe)

When you describe Bugs of this program, please look first into the chapter

Bug reports  
!

We will put the actual Eagleplayer onto these disks and send them back. If no new Version is available at this time you'll have to wait for your disks until the next version will be released. Ok ?

## 1.9 fehlerbeschreibung

### 2.8 Bug reports

-----

If you wanna describe an Error of the Eagleplayer you should try to report it as detailed as possible.  
A message "The Eagleplayer crashes whilst loading" doesn't help us at all. It can happen that the module is corrupt or the depacker doesn't work or the current replayer destroyed the memory. There are lots of opportunities.

Following we listed some things you should answer:

- \* Are you a registered User ?
- \* Which Eagleplayerversion are you using ?
- \* Hardwareconfiguration ! (Amiga Model, Memory, OS etc.)
- \* Which Patches did you install ?
- \* Free Memory when crashing ?
- \* Which Players/Userprograms were loaded ? Is it obviously unimportant that players/userprograms were loaded ?
- \* Does this Error happen using other Computer configuratuions ?
- \* Your Config file we need as well !
- \* Which Module makes problems happen ? SEND IT US PLEASE !!!
- \* Does this Error happen only on Machines with cache ?
- \* How were the current internal settings of the Eagleplayer ? Are them unimportant for this Error ?
- \* Was the module recognized correctly ?
- \* Was the Mainwindow opened ?
- \* Is the Error Kickstart (OS) dependend ?
- \* Which Userprograms were active ?

## 1.10 keyfile-benutzung

### 2.9 Using the Keyfile

-----

Registered Users get a Keyfile from us which has to be copied to "Devs:". This keyfile allows the Access to the functions disabled until this time.

---

The Keyfile is person depending. It contains the full Address of this User his Usernumber and other datas. If we should get to know that a keyfile was given away to other persons we take him to court !! (JUDEMENT DAY !!)  
In the next version of the Eagleplayer this keyfile won't be accepted.

All Keyfiles may be used in upcoming new versions of the Eagleplayer. It is possible that further DEFECT-productions work with this keyfile (more won't be said)

## 1.11 systemanforderungen

### 3.1 System requirements

-----

The EaglePlayer needs at least Kick 1.2 and 512 KB of Ram. But I have to mention that some functions of the EaglePlayer, like Appwindow (see there) will not run under 1.2/1.3. The same matters for a big part of User-Programs written by us. On machines that work with pseudo 2.0-versions (all 36-er Kickstart versions), we do not guarantee anything, because, because these Kickstart-versions were not finished at this time. When we talk about "2.0" from now on, we think of the Kickstart-versions from V 37.175 (OS2.04), which are regularly in ROM in A500+/A2000E. Included are the following versions 3.0 etc., which are used in A1200/A4000. Tests on this machines have been successful :) --==

There is also often the mark (r.v.). That means just that these functions will only be usable in the registered versions.

The Amiga should of course be connected to a monitor with speakers or to a stereo-tower/"ghetto-blaster", because otherwise you just won't hear anything (I think so).

You have to be patient that no other sound-program is running already, which uses the audio-channels. In this case the player will reply a message after loading a module (see "Errormessages").

Attention: Although we are saying that the EaglePlayer runs with 512KB MEM, it may be that still 1 MB is not enough when you have loaded all of our User-Programs and all of our External Players that are available now. So if you do not have such amounts of memory you should think about which players you need and which not! MAKE USE OF THE

EAGLEPLAYER-BATCH  
ABILITIES !!

## 1.12 benötigte zusätzliche dateien

### 3.2 Additional files needed

-----

The Eagleplayer doesn't depend on any library anymore !!

---

The EaglePlayer tries to open XPKMaster.Library, Crm.library, PowerPacker.Library such as LH.Library to load files packed with these packers. These Libraries are NOT NEEDED for starting, but are recommended, anyway if you want to load such packed modules.

To Use filerequesters you will need at least one of req.library, asl.library and retools.library.

The Eagleplayer tries to load a  
Configuration  
file

## 1.13 installation auf diskette, festplatte, ...

### 3.3 Installation on Disk, HardDisk, ...

-----

You are advised to use the automatic installation script basing on the commodore installer tool. (All actions are described there with a little help text) For installing using our script you should boot from your harddisk to make it easier for you since the script can find certain directories for libraries etc. automatically and you won't need to enter them by hand.

To install the EaglePlayer by hand, the following steps are necessary:

- at first you have to copy the Libraries mentioned above to "Libs:" ("Libs"-directory on your Boot-Disk/Partition)
- the second step is to copy the file "EaglePlayer" in your favourite directory (e.g. "DH0:Music/Players/" or "DF0:" ...)
- then you have to define the players that should be immediately loaded on program-startup and to copy them into one of the following directory-variants. (The EaglePlayer searches in the order listed below and then loads only that External Players that have been found in the first directory)

```
"EaglePlayers/"      (in your current directory)
"Sys:EaglePlayers/"
```

- at last the User-Programs have to be installed, just by creating either a directory called "EP-Userprograms" or "Sys:EP-Userprograms" and copying the wanted User-Programs into it.

Example:

```
copy EaglePlayer to dh0:Music/Players/
mkdir dh0:Music/Players/EaglePlayers
mkdir dh0:Music/Players/EP-Userprograms
copy EaglePlayers/ to dh0:Music/Players/EaglePlayers/ all
copy EP-Userprograms/ to dh0:Music/Players/EP-Userprograms/ all
```

ATTENTION!! From Version 1.52 on the Eagleplayers and Userprograms are

---



packed onto the second disk into an lha file. If you don't know how to handle LHA then use the installer script !!

## 1.14 start des eagleplayers von cli & shell

### 3.4 Starting the EaglePlayer from CLI/SHELL

---

To start the EaglePlayer from Cli/Shell you only have to run the Main Program. Before that please change the current directory to that one, where you have put the Player with its Players and User-Programs (DOS-Command 'CD'). This is important in case if the Player-Directory (Deli- or EaglePlayers) as well as the User-Programs are in the same directory as the EaglePlayer itself. If the EaglePlayers and Userprograms are located in "SYS:", The 'CD'-Command will not be necessary and you only have to run the Program. The 'Run'-Command is not necessary to detach the EaglePlayer from Cli/Shell, because it will link off after starting from Cli/Shell, so it will be further usable and may be closed, too. In the unregistered version you cannot deliver any arguments, neither from CLI/SHELL nor from Workbench.

## 1.15 start von der workbench aus

### 3.5 Starting from Workbench

---

To start the EaglePlayer from Workbench you simply have to double-click on its Icon, which will load it then.

Attention: Up to now there is NO Tool-Type support.

## 1.16 das konfigurationsfile

### 3.6 Das Konfigurationsfile

---

From version 1.10 upward, the Eagleplayer will load a Config-File, which consists of a couple of commands that are entered as strings. In the Config. you can, for example, specify the current Player-Directory or if Fadein shall be switched on or off etc. The file normally will be located in the "ENV:EaglePlayer/" directory. When running the Player it will at first try to load it from the current directory and then from the S: -directory. The Config-File is not actually needed for starting the Player. Please note that when you save a configuration the current settings are saved, such as Modulename, the User-Programs' settings as well as all paths and preferences-setttings. The commands for the Config-File can be found in the configuration on the Main-Disk or in other saved configurations. You don't have to use all commands in your configuration.

The commands of the config file are partially used in the

---

Arexx-Commands

.

## 1.17 das hauptfenster

### 4.1 The Main Window

After having started the Eagleplayer successfully you'll see the following window:

```

I
n
f
o
P
a l
b a
o y
u e
t r

M W I c
o i o
Q d n n
u u d i
i l o f
t e w y

| | | |
| | | |
v v v |

-----|
|*| Eagleplayer V1.52 Generic/Registered | | | |
|-----|
| |1| | | Volume | | | |zz | <--
| | | ? | P |-----|
|_|2| | | Balance| | | | |] | <- Filter
|_|-----| | | | | | | | | | | | | | | | | |
|_|3| |< | < |<<| |[]> | |> | >> | |>>| > | >| | [] | ^ | 00|
|_| | | | | | | | | | | | | - | 00|
|_|4|-----|
|_| Welcome to Eagleplayer |
|-----|

^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
| | | | | | | | | | | | | | |
S V p p p r P F n n n S e L
p o r r r e l a e e e t j o
e i e e e p a s x x x o e a
e c v v v l y t t t t p c d
d e i i i a t

```

s o o o y / P P S M a  
 u u u l a u o M  
 s s s S P a t b d o M  
 o a y t s u d o  
 M S P n u e o l u d  
 o u a g s r n e l u  
 d b t e n g e l  
 u s t e  
 l o e  
 e n r  
 g n

## 1.18 steuerung des programms über die schalter (gadgets)

### 4.1.1 Controlling the program using the Gadgets

After starting most the Gadgets of the Main Window should be disabled (they're "ghosted"). To load a music-file you simply have to click on the disk-symbol. After this a filerequester will appear where you choose the music-file(s) you wish to play. After clicking on "OK!" the Eagleplayer attempts to load and play the first music-file. While playing these following buttons may be enabled.

Speed - Easy way to change the speed of the played song. If the knob is in the middle, the normal speed is chosen. For faster playing, move it upward, moving the knob downwards decreases the speed.

- or Plus - and Minus Gadgets

To stay Delitracker-compatible we included the possibility to control speed by clicking on the [+]- or [-]- buttons.

Voices - to switch the 4 Voices of the standard-Amigas on/off, you have to "press down" the appropriate switch (off), else the voice is on.

previous

Module - makes the player load the previous module of the actual list. If the current module is already the first one, the player will load the last one. If "Load Dir" was disabled, the function will also not be available. Attention: If you switched "Random Module" on, this player will not load the previous songfile and the song will replay forever.

previous

(Sub)song - for soundsystems, where one module contains not only one song. This function plays the previous subsong. If the actual song is already the first one, nothing will happen. If the module contains only one song the gadget will be disabled ("ghosted").

previous

Pattern - plays the previous pattern, if the current one was already the first, the last one will be played.

Note for beginners: A pattern is comparable with a note-sheet. Some soundsystems use this method by buliding up a list of instructions executed after another. These lists, called "patterns" can be combined in any way you like.

Replay Song - the loaded songfile will be re-played up from the beginning

Play / Pause - switch between play and pause, if the module plays when you select this button, it will stop and when you click this button again, it will continue

Fast Play - "fast forward", gives you an easier possibility to search after a certain passage in the songfile.

next Pattern - plays the next pattern (see at "prev. pattern"). If the last one is reached, the first one will be played.

Next (Sub)song - plays the next subsong of a music-file with more than one song within it. If the module countains only one song this function will be disabled.

Next Module - loads the next musicfile of a list (see also at "prev. module").

Stop - stops the current song, sets the actual pointer where to play to the beginning of the song.

Eject Module - stopps the actual module (if this is playing or not) and throws it out of the memory.

Load Module - opens a filerequester for choosing a musicfile to be played. You may choose more than one by holding down the "SHIFT"-key and clicking the files. In this case the chosen files will be overtaken and you can choose these modules by selecting "prev. Module" or "next module" or you use the possibility selecting the modules directly via the pulldown-menu (Modules-Directory). If you choose only one songfile the whole directory will be loaded. (some files e.g. these ones ending with '.nt', '.nt.pp', '.txt' or '.s' will be hidden automatically).

Iconify - Changes the size of the main window to a minimum, closes it (when set) and puts in this case a so called "Appicon" onto the Workbench or adds a Menu-item to the tools-menu of the Workbench. (both functions require OS2.0) To open the main window of the Eagleplayer again, doubleclick on the Appicon or select the Menuitem (See also chapter 4.1.2 "Iconify Modes")

Switch Filter - Switches the state of the Low-Pass-Filter of the Amiga (no effect when you use the Amiga 1000 or a 2000A - better buy a 1200/4000)

Info About

Module - scrolls an information-text through the main window telling something about the current song in memory and the replayer

for this music. The size of this text depends on the player. If the player doesn't support moduleinfo, only modulename and size are given. If no module was loaded, an information-text about the player will be given out.

Quit - the Close-Gadget quits the Eagleplayer (Oh No!).  
 Notice: Under 2.04 the player won't be quitted by default but the windows and the Userprograms will be closed. To quit select the menuitem "Quit" (  
     Chapter 4.1.2  
     4.1.2 Control via Pulldown Menus  
   ) or set  
 "Quit Eagle" within the preferences (also (  
     Chapter 4.1.2  
     4.1.2 Control via Pulldown Menus  
   )  
 and then press on the "close-gadget".

Volume - changes the volume

Balance - changes balance

Playerwindow - opens/closes the Playerinfo-Window

## 1.19 steuerung über die pulldown-menüs

### 4.1.2 Control via Pulldown Menus

-----

The Eagleplayer offers the following pulldown-menus:

Project   Commands   Special   Modules-Directory

The functions in the "Commands"-menu are exactly the same as the gadgets in the main window, so please look there (  
     chapter 4.1.1  
   ) for explanations.

The Project Menu

~~~~~

Load Module - loads a module (look into (
 chapter 4.1.1
)
)

Save Module - only in the registered version. Opens the filerequester to select a file where the actual module should be saved to.

About

Module - scrolls an information-text through the main window telling something about the current song in memory and the replayer

for this music. The size of this text depends on the player. If the player doesn't support moduleinfo, only modulename and size are given. If no module was loaded, an information-text about the player will be given out.

- Add Player - Opens the filerequester for selecting an external player to be added.
- AddPlayerDir - Let's you select a whole directory of external players to be added.
- Show Players - opens the Playerinfowindow (see chapter 4.2.1)
- Load Config - Load a settings file by selecting it via the filerequester.
- Save Config - Save the current settings by selecting via filerequester (usually to "ENVARC:Eagleplayer/Eagleplayer.config")
- Delete File - Lets you delete a file
- Help - Online Help, only for registered users
- Iconify - Changes the size of the main window to a minimum, closes it (when set) and puts in this case a so called "Appicon" onto the Workbench or adds a Menu-item to the tools-menu of the Workbench. (both functions require OS2.0) To open the main window of the Eagleplayer again, doubleclick on the Appicon or select the Menuitem (See also chapter 4.1.2 "Iconify Modes")
- About - Opens the text requester containing infos about the Eagleplayer and their creators (that's us !)
- Hide - OS2.0+, closes the windows and the Userprograms. To get the Eagleplayer "back" there are 4 possibilities: First and easiest, use the "hotkey" (default: CTRL+ALT+E), second use "commodities exchange" (supplied with your workbench release 2.0 and above) to activate the Eagleplayer, third call the program via ARexx back and the fourth method is to start it again. The second Eagleplayer quits suddently after starting but signals the first one to appear. Hope you like it.
- Quit - quits the Eagleplayer
- Das Special-Menu
~~~~~
- Preferences:  
~~~~~
- ScrollInfos - The Informations like Filter on/off will be scrolled into the main window if enabled
-

- FlashPointer - Whilst depacking a module the mouse-pointer will flash if enabled
- Fadein - When start playing a module the volume will increase from 0 to the normal value
- FadeOut - When stopping playing a module the volume will decrease to zero before the replay stops
- Filter - Should the filter be set at the start of the module or not...
- MasterVolume - If this option is enabled, the "Analyzing" Userprograms like "Analyzer", "Levelmeter", "Levelgraph" react on the current volume by changing their output depending on the current volume settings, what means that on a small volume the output won't be high exact in the same way. (r.v.)
- Auto Password - If you are using XPK-encrypted modules you this option makes the eagleplayer use always the same password instead of asking every time you attempt to load a module
- Use Songname - display the modulename instead of the filename in the titlebar of the parent screen (if supported by the replayer)
- Use Prefix - When saving a module a prefix will automatically be added in front of the module, if available (useful in connection with "Use Songname") (r.V)
- Quit Surface - If set, the Eagleplayer will quit if a Surface Userprogram like "HIFI" or "Noiseeagle" is closed.
- Quit Eagle - (OS2.0) If this option is set, the Eagleplayer quits on clickin on the "Quit" button of the main window, else only close all windows. (See also "HIDE" in this chapter and chapter 4.1.1)
- Save t: - this function offers you the possibility to load packed modules even when the replayer doesn't support packed ones by depacking the module to "t:" and loading it again
- LHA-Extraktor - when enabled, LHA packed files will be depacked (LHA is looked for in "c:")
Note: Only the first file will be taken !
You should'nt use LHA for packing modules. It is too slow and the results of packers like QPK-SQSH, XPK-SMPL or LH.library with Samplemode are better.
-

LoadFast - This is a function you can load the module into fastram if supported (Oktalyzer 8Voices, all internal Trackers except Startrekker/Audio-Sculpture)
That means you can save your rare Chip memory. Useful when you use Modules >250 KB

~~~~~

Eject Players - All unused external replayers will be deleted. When removing a module from memory. Only useful in connection with "Use PlBatch". Lets you save lots of memory.

Eject UserPrg - If a userprogram is closed it will be removed from memory. Only useful in connection with "Use UPrgBatch". Let's you save memory as well as the "Eject Players" function

Use PlBatch - The Batch-Loading is switched on/off

Use UPrgBatch - The Batch-Loading is switched on/off

Load PlBatch - A  
EPBatch-File  
for players will be  
loaded

Load UPrgBatch - A  
UPrgBatch-File  
will be  
loaded

#### Modules-Prefs

~~~~~

Rescan Dir - The Directory will be reloaded

~~~~~

Loaddir - The Directory will be loaded and shown in the pulldown menu. If this option is disabled, "Nextmodule" and "Prevmodule" are disabled as well. The Loadnext, Load Previous and Random functions are not able to work. (should be clear, why)

Use EPDir - This is a kind of ".fastdir" offering the possibility to load a complete directory within a very short time. It works by saving the directory into a small file within it and reading it instead the whole dir. (r.V.)

Autodir - On changes within the actual directory or disk the directory will be read again

Notify - (OS 2.0) The notify function is the same like "Autodir" but much safer and future compatible, should be used always instead of "Autodir"

---



Scan Always - Should the directory be loaded at any time a new module is selected (using the filerequester) or sent from an external source (Noiseconverter, M.E.L.G.U)

~~~~~

Show DirNames - Shall the directories be taken over into the pulldown menu ? (r.V.) Using this function you may switch into subdirs or go to parent directories without using the filerequester. Very powerful and user-friendly function !!

HideAll - Using this option all as "non Module" known files ending e.g. with *.asm or *.nt will be hidden.

~~~~~

CheckDifference - disabled, not ready yet

DirMemsize - Enter the size (in Bytes) of the internal directory buffer. One file or directory needs 100 Bytes.

Entries Per Menu- Enter the number of max. entries for each modules menu. You may enter a value between 0 and 63 (limited by the operating system) Entering 0 the Eagleplayer will calculate the maximal value possible for every screen itself.

#### User-Programs

~~~~~

Add UserProgram - A Userprogram may be added via File-requester

Add UserDir - Add a whole directory of Userprograms.

~~~~~

(example) Analyzer \_/ enable/disable the Userprograms

#### Save-Mode:

~~~~~

Uncrunched - The Module will be saved unpacked.

PPcrunched - The Module will be packed before saving with the Powerpacker algorithms. You may between choose Normal, Protected and Samplemode or combine them.

!! The Powerpacker Samplemode is a creation of DEFECT. Powerpacker © Nico Francois. !!

LHCrunched - The Module will be packed using the "lh.library" when saving. You may choose between "normal" and "Samplemode".

XPKCrunched - Save the module using one of the XPK modes. (see below). For using a "Crypter" (see XPK documentation) select "protected" mode.

CRMCrunched - The module is packed with crunchmania modes. You may choose between Normal, Protected and Samplemode or combine them.

```

~~~~~
Automatic Save - if this option is set the Eagleplayer
                 saves the current module into the dir
                 set using "SetSaveDir" when selecting
                 "save module" without opening the
                 filerequester. Useful for getting lots
                 or new modules and checking whether
                 you wanna get them or not.
Overwrite       - If this option is not set a warning
                 will appear if you attempt to over-
                 write a file whilst saving. (r.v.)
Safe Save       - not implemented yet
SSdir          - try to save the module into the
                 directory named as the soundsystemname
                 (e.g. The Chiptrackermodule "Anarchy"
                 would be saved into the dir
                 "Chiptracker" under the name "Anarchy")
SampleMode     - pack via "Samplemode" for better
                 crunching results when available
Save As PT     - The actual Module will be saved as
                 Protracker by the internal Players
                 (except Customplay, Startrekker, Audio-
                 sculpture and Oldsoundtracker)
Protect        - For packed savemodes offering the
                 ability saving the modules in a
                 "crypted" way, see above
Protectionbits - sets the module to "read" only after
                 saving to prevent from deleting, writ-
                 ing or even attempts to execute it
~~~~~
Set Save Dir   - set the directory for the autosave
                 function
Change XPKname - Changes the XPK-Packer or Crypter for
                 saving the module and packing via XPK
Change Password - lets you set the default password

```

Programm-Mode:

```

~~~~~
No new Module - When a Songend is found no new module
                 is loaded and playing stops
LoadPrev      - If Songend or Playtime is set the
                 previous module of the list or directory
                 is loaded on Songend
Loadnext     - If Songend or Playtime is set the
                 next module of the list or directory
                 is loaded on Songend
RandomPlay   - If Songend or Playtime is set a random
                 chosen module of the list or directory
                 is loaded on Songend of the current
                 module. This routine is written in a
                 way that no module will be played twice
                 before every module was played. But it
                 is possible to select a module using
                 the pulldown menu. In this case the
                 module will not be remarked and possibly

```

played twice. If you select the "Nextmodule" and "Prevmodule" gadgets, the randomplay will be performed as well. (like CD-Players do)

~~~~~

Songend - If enabled the playing will stop if the end of song was reached

Load Before - The next module will suddently loaded after loading a new module so that 2 modules are at the same time in memory and the player can change to the second one megafast (very useful for megamixes like "RAF Mega Mix" or "LSD - Total Chaos")  
Must be used together with "Load next", "Quickstart" and "Songend" recommended!

Auto Subsong - before Loading the next module the Eagleplayer will play all available Subsongs ("Songend"/"Set Playtime" !)

Randomsong - Äquivalent zu RandomModule bloß mit Subsongs

LoadAlways - Voraussetzung: Songend=on oder Playtime gesetzt und LoadNext=on  
Es werden so lange Module geladen, bis eines abgespielt werden kann.

Random Song - Choose the subsongs in a random order

Load Always - Load one module after another until the player finds a known one (useful if you didn't load all replays but wanna hear music without a break)

Time Mode -

Calc Duration - Here you may set if you want the duration of the loaded modules is calculated when supported or not when you think it lasts too long

Quickstart - When module was loaded, it starts automatically playing or (if disabled) waits until you press the "Play-Pause" button

~~~~~

Set Playtime - asks you for entering the playing time in seconds until the module will stop (and the Player loads the next if preferred)

Warning:

~~~~~

small Module - let a warning requester appear if module too short

big Module - let a warning requester appear if module too long

Module fault - let a warning requester appear if errors in the module (not used currently !!!)

## File-Requester:

~~~~~

No Filerequester- No filerequester used, all inputs are done using a stringgadget in the main window (Situated where the Scrolltexts appear)

Req.Library - the req.library will be used

Reqtools.Library- the Reqtools.Library will be used

ASL.Library - (OS2.0) the ASL.Library will be used

Iconify:

~~~~~

little Window - on command "Iconify" (see above) change the window size to a minimum

App Icon - (OS2.0) Iconify to an APP Icon placed onto the Workbench  
you may also "drag and drop" icons of modulefiles onto this appicon to play these music-files

App Item - (OS2.0) Iconify to an App Item added to the "Tools" menu of your Workbench

Set Voices - lets you change the state of the 4 Voices provided by the standard Amiga Hardware between on/off

## Set Volume

Default Volume - set volume to max.

Higher Volume - make it louder

Lower Volume - silence is golden

## Set Balance

Default Balance - set balance to "middle"

Balance to left - a little more music from the left side

Balance to right- a little more music from the right

## Set Speed

Default Speed - Set speed back to 0=standard

Higher Speed - in a hurry eh ?

Slower Speed - got much time or what ?

Change Pubscreen - You're asked then for entering the name of the public screen the Eagle-player should appear on, see chapter 4.1.5

Toggle Filter - Toggles the current state of the audio Filter on/off

the Modules-Directory Menus

---

-----

Here you may choose the songfiles to play directly by selecting the items. You may also get into subdirs or choose the parent dir. (selecting the items) It is even possible to use multiple volumes. Therefore read the text to the Appwindow function.

## 1.20 tastaturkürzel

### 4.1.3 Shortcuts

-----

The most important functions of the pulldown menus support key-usage. Following there are the combinations listed which start the corresponding functions.

#### \* Project-Menu \*

Load Module           Right Amiga + l  
Save Module           Right Amiga + w  
About Module           Right Amiga + a  
Help           Help-Key  
Iconify       Right Amiga + i or "I"-Key  
Hide         Right Amiga + h or "ESC"  
Quit         Right Amiga + q

#### \* Command-Menu \*

Prev Module           Right Amiga + b  
Prev Song             Right Amiga + c  
Prev Pattern           Right Amiga + y  
Replay Song           Right Amiga + r  
Play Song             Right Amiga + p  
Player Faster         Right Amiga + o  
Next Pattern           Right Amiga + x  
Next Song             Right Amiga + v  
Next Module           Right Amiga + n  
Stop           Right Amiga + s  
Eject Module           Right Amiga + e

#### \* Special-Menu \*

Modules-Prefs  
  Rescan Dir   R-Key  
User Programs  
  Add UserProgram U-Key  
Programm-Mode  
  Set Playtime   Right Amiga + z  
Set Voices     Right Amiga + 1-4 (je nach Voice)  
Set Volume  
  Default Volume   Right Amiga + 5  
  Higher Volume   Right Amiga + 6  
  Lower Volume    Right Amiga + 7  
Set Balance  
  Default Balance   Right Amiga + 8  
  Balance to Left   Right Amiga + 9  
  Balance to Right   Right Amiga + 0

```
Set Speed
  Default Speed Right Amiga + #
  Higher Speed  Right Amiga + +
  Slower Speed  Right Amiga + -
Toggle Filter  Right Amiga + f
```

## 1.21 die appwindow-funktion

### 4.1.4 The AppWindow-Function

---

The Eagleplayer uses in all it's Windows (not the Userprograms, these are external Tasks) the Appwindow-function. If you e.g. have got an icon for a Soundmodule, drag it onto an Eagleplayer Window and release it, this module will be loaded and played. If you drag more than one icon into the window they'll be overtaken in an alphabetical order into the pulldown menu. The Files may exist in any directory you want !!

## 1.22 pubscreens

### 4.1.5 Pubscreens

---

The Eagleplayer may be opened on Public Screens. The Pubscreenssupport is available for all Windows connected to the Eagleplayer. That means that all Userprograms open also on this Screen. Now follows a short list of possible public screen names:

```
CygnusEdScreen1 (at Rendering choices in the Environment-Menu
  "make screen public" set to "on")
BootX
PowerPacker
DOPUS.1
TERM
Workbench
```

## 1.23 font-unterstützung

### 4.1.6 Font-Unterstützung

---

You may change the font of the Eagleplayerwindow in the configuration file. You simply need a texteditor (e.g. memacs,ed,CED or ASM-ONE hehe) and change the line 'Font "topaz.font"' to the font you like and save the file. WARNING: The Eagleplayer accepts only Fonts with the size 8\*8 unpacked and not proportional ! If other fonts are used, the Eagleplayer produces shit whilst scrolling. On the Main disk there is such a font as an example called

---

"eagleplayer.font"

## 1.24 interne depacker

### 4.1.7 Intern DePackers

---

In this version the Eagleplayer is able to unpack datafiles packed with one of the following crunchers:

- XPK
- Powerpacker (if exists, the library will be used)
- LH (PMC from Powerplayer- but what of the Power-  
player has to do with POWER ???) [LAME PLAYER]
- CrunchMania (only Library, but all modes)
- StoneCracker V4.0
- Fimp
- Pack (Soundtracker Songpacker)

Using Crunchmania and Fimp is only one Memoryblock needed into which the module will be loaded and unpacked. This method works fine but only when using CRM and FIMP. Using other packers there's needed to allocate another memory block. This method needs a lot of memory and that isn't good at all. After depacking the memory used for the packed file will be freed at once. The Eagleplayer supports the XPKMaster.library, when the needed Libraries can be found. If a file should need a password it can be given in via a String-gadget appearing in the Main Window.

To depack PMC-Files (with files packed with LH.library) this lib is needed. If it can't be found the module won't be depacked.

Pack is the packer of the Soundtrackers (Noisetracker/Protracker/Startrekker) These files will automatically be depacked when loading. That means that every external Player is able to load packed Songs of it's format.

Powerpacker-, Crunchmania-, Fimp-, Pack- and Stonecracker4.0-Files may be depacked without external libraries (they're built in). If the Eagleplayer finds the PowerPacker.library, this lib will be used to depack PP-files.

Out of this the Eagleplayer is able to load packed executable files, if they are packed using the Imploder 4.0 (Normal Mode without library). This function was implemented because the program got bigger and bigger, especially the UserPrograms and external Players. To make these usable for a disk-user they're packable. Custom Modules may also be packed via the Imploder.

For saving packed modules, see chapter 4.1.2 "Save Modes"

## 1.25 der lha-extraktor

### 4.1.8 The LHA-Extractor"

---

From Version 1.50 on the Eagleplayer offers the possibility depacking LHA-Archives. At this time only the first entry will be depacked and loaded.

---

For depacking the program "LHA" itself is used which has to be found in the "C:" directory.

The module will be depacked to "T:" and loaded from there.

For future releases we plan a full support.

## 1.26 öffnung und funktion des player-fensters

### 4.2.1 Opening and Functions of the Player-Window

---

The Playerwindow will be enabled/disabled using the Menu-Item "Show Players" of the "Project" Menu in the Main Window. On Iconify it will also be closed. If you set "little Win" as Iconify Mode, you are able to open it also when the window is in Iconify mode.

## 1.27 steuerung über die schalter und erklärung der funktionen

### 4.2.2 Controlling using the buttons and description of it's functions

In the recessed-bordered Listview-Gadget at the left side you choose the Replayers. If you selected one player you are able to do the following:

- read, what functions it supports, e.g. ModuleInfo, Save, Packable
- delete it (except internal players)
- disablen/enable it, that means you are able to switch it off without deleting
- change it's config/load it/save it/

the following actions will be performed in the Player-Window

- Add-Player - Loads a new player. If already a player exists with the same name it will only be loaded then the version number of the new one is equal or higher.
- Del-Player - deletes the selected player
- Config - calls an internal routine of the chosen player where you
  - e.g. change it's instruments directory.
  - (if possible)
- Add-PlayerDir - Here you can add a whole dir with replayers
- Load - calls an internal routine of the chosen player where you load its configuration-file (if possible)
- Save - calls an internal routine of the chosen player where you save its configuration-file (if possible)
- Enabled Yes/No - Here you enable/disable he player by clicking on that button. The player won't be deleted !

## 1.28 tastaturkürzel im playerwindow

---



### 4.2.3 Shortcuts in the Playerwindow

---

The most important functions of the playerwindow support key usage. Following these keycodes are listed:

```

Close Window      ESC-Key
Add Player        A-Key
Add PlDir         P-Key
Delete Player     D-Key
Delete All        E-Key
Enable/Disable   E-Key
Config           C-Key
Load             L-Key
Save             S-Key

```

Moving the entries of the Listview-Gadget is possible by using the Cursor-keys.

## 1.29 interne player

### 4.2.3 internal Players

---

The Eagleplayer owns in this version 18 internal players. Internal players can not be deleted but enabled/disabled. In the checking progress the external Players will be called first and the internal at the end of all. All internal Players added after Version 1.40 are formats converted back to the original Protracker format before playing for best compatibility. All these Modules may be saved in Protracker's format.

The following replays are Eagleplayer-internal:

```

Audio Sculpture -      See at Startrekker
CustomPlay      -      This is an interface for songs not belonging
                    to a certain sound-system or only working on
                    fixed addresses. Therefore these songs have to
                    be adapted and a playerheader like on the
                    "real" players has to be at the beginning.
Digital Illusions-      SoundPacker, used in the games "Pinball Dreams"
                    and "Pinball Fantasies", packs not bad, try to
                    check out the Noiseconverter results.
Game Music Creator-      Oldsoundtracker-like Soundformat. Nothing
                    special.
Kript           -      A Protrackercrypter, which uses a Longword added
                    to the Patterns. This player is normally dis-
                    abled. To use Kript Modules, enter the Correct
                    Code first (using the config ability of the
                    PlayerWindow) When this replayer is active it
                    recognizes every Protracker module.
Noise/ProTracker -      The most used Soundsystem ever on the Amiga !!!
                    We don't differ between Noise,Sound and Pro-
                    tracker (because we don't think this to be
                    useful - the Delitracker one shitted too much).

```

---

If you have got modules not working with this replayer, please send them us.

It is possible to load not only modules but songs. The default instruments path can be chosen via the configuration. This path will be used for all internal sample loaders. By the way: We use an original Protracker V1.0a Playroutine in which we solved lots of Bugs ! Probably the only Protracker replay correctly working on A4000 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Noiserunner - A Protrackerreplayer from CAOS of Sanity which was written for a maximum Speed. It changes the module only a little.

Old SoundTracker - Here we use the Soundtracker 2.0 Routine by Unknown of Doc. It seems to be quite compatible. Modules not working, please with Testmodule and - when available - Replay-Source into the next mailbox leading to DEFECT Headquarter Rostock. Songloading is ofcourse supported.

ProPacker 1.0 - A PT-Packer by Azatoth of Phenomena. Attention ! Too Short Modules will not be recognized because of the fact that the patterns are situated at the end of the Module and MUST BE COMPLETE. (crash danger) Some stupid people call this format "Hannitracker" or even "Strangeplayer".

ProPacker 2.0/3.0- PT-Packer by C. Estrup (Static Bytes).

ProRunner 1.0 - A Packer by Cosmos of Sanity. It's an Encoder for higher replay speed.

ProRunner 2.0 - Real PT-Packer by Cosmos. Packs quite good although it countains some bugs and packs really slow. Use the Noiseconverter instead.

Player 4.x - Finally the ultimate player for this confusing Format. The Modules of "P40A", "P40B" and "P41A" will be loaded and replayed by our Protracker routine.

P4.0/4.1 is a very efficient packer for Protracker Modules.

Packer von Jarno Paananen.

Startrekker - The replayer used is the one from Audiosculpture (commercial Successor of the legendary Startrekker) The differing between Startrekker and Audio sculpture is only in the NT-file. If no external file is loaded the module will be recognized as Startrekker. With the build in Routine you can play Startrekker Standard Modules, Startrekker AM Modules, Audio-sculpture Normalmodules and Audiosculpture AS Modules.

The .NT/.AS File isn't needed in all cases, except the modules uses AM Samples. Songloading is of course supported.

Unic-Tracker,

Unic-Tracker2 - A Packer by Laxity of Kefrens. Modules to be found in the most new Kefrens Productions.

WantonPacker - A tool by Wanton of Bloodsuckers. It's only

a Crypter for Modules, it's Playroutine doesn't handle all effects correctly.

## 1.30 einleitung zu den userprogrammen

### 4.3.1 Introduction to the Userprograms

---

The Userprograms are something really new on the market. They are externally loaded, self running programs, which work asynchronous. (every Userprogram is a separate process) Basically designed as simple analyzing programs there came then new possibilities which give them the power not only to be controlled but control the Eagleplayer itself. The Userprograms can access all important datas of the Eagleplayer so that this opens really new worlds. So it is absolutely no problem creating a completely different surface for the Eagleplayer, e.g. with own screen in a CD player look. Therefore it is only needed to write the surface with input-handling and to give the commands away to the Eagleplayer, as easy it is possible to change the surface within less than one second, e.g. use the Intuitracker-nostalgic look. The Eagleplayer is currently able to hold 15 Userprograms in memory, which all can be enabled/disabled the way you want. Their settings like on/off, position of the window, mode (Analyzer) can be saved. When needed you can load one Userprogram, twice, three times or more. You only need to add it via "Add Userprogram" or copy the file under different names into your "EP-Userprograms" dir. An information-window for the userprograms will be written in the near future.

## 1.31 steuerung der user-programme

### 4.3.2 Steuerung der User-Programme

---

#### 4.3.2 Controlling the User-Programs

The Userprograms can be switched on/off selecting the menu-item for each of them. If it is selected (little "hook" right of it), this Userprogram is enabled and active.

If the close-gadget of the Userprogram is clicked (or you choose the menu-item "quit") the Userprogram closes its window but won't be removed out of memory and is re-activateable in the way described above.

Every Userprogram has its own possibilities and hints, to describe their full functions would be a little bit too complicated.

A short tip to the Analyzer: Look into its pulldown-menu, there you will find an option to switch off the Scroller and the little picture, if you hate them. AND DO NOT FORGET THAT ALL GADGETS ARE AT LEAST USED TWICE ! Try all Modes or die die die ! (more than 20)

The following finished Userprograms may also be used under Kickstart 1.2/1.3 (the UPS marked with '\*' are only for registered users)

- Time
- Levelgraph
- Noiseseagle
- \* Levelgraph2
- \* Levelmeter
- \* Analyzer2 (Small Analyzer)

The others Userprograms need at least the Operating System 2.04:

- Analyzer
- Quadrascope
- HiFi (looks like Technics and is named "Kenwood")
- M.E.L.G.U. (Mega Eagleplayer List Generating Unit)
- \* Stereoscope
- \* Monoscope
- \* Voicemeter

## 1.32 zu den einzelnen bisher verfügbaren userprogrammen

### 4.3.3 to the Userprograms currently available

-----

#### Analyzer

- Quasi Spectrum Analyzer, which works using the usual method reacting on the samplerate which was used for playing the actual note. (like Sound/Noisetracker etc.) Although some people don't wanna believe it this piece supports 21 different modes (ok, some only differ in trifles).  
Who cares about the scroller or the picture may switch it off by selecting the corresponding pulldown-menu item. These settings will also be used in the configuration as like as the actual mode.

#### Quadrascope/Stereoscope/Monoscope

- shows the actual played sample of each voice, either separated (Quadrascope) or additive put together. (Monoscope, Stereoscope)  
This Output is done in realtime, that means that on an oscilloscope connected to the Amiga the same picture should be showed.  
-> NEW NEW !! 2 Modes added available using the pulldown menu !  
Solid Mode (like in Protracker 3.20) and Line Mode connecting every Pixel with a line for more authentic look.

#### Time

- shows the current play- and system time. The program was rewritten in a way that the processing time usage was decreased drastically.

#### Levelmeter/Levelgraph/Levelgraph2

- REAL Output analyzer which reacts on the volume of the sample currently played. It shows the value calculated additive in the same way the samples are displayed in the scopes.

#### Voicemeter

- This is the same gag like in the usual tracker-programs. (see also the Analyzer description). Here the different modes are available by pressing onto the key "1". A selection using gadgets is in preparation.

#### Analyzer2

- little, under Kickstart 1.3 available version of the analyzer, still not concluded, but the final version follows recently

(sometime at the end of '93)

#### Noiseeagle

- Who doesn't like the Eagleplayer surface at all (Hi CPT. BIFAT of TEK) may start this Userprogram that orders a file and plays it, in the same way as the good old Noiseplayerdoes it. The difference to the original is that "some" more replays will be supported and I didn't include the hardwarehacking mouse/keyboard interrogates.

#### Hifi System V1

- There was a time I checked out all my pictures and I found a pic of a HiFi tower drawn by a platin member which I adapted (after asking the creator) to my own ideas and present it now as the first complete new Eagleplayer surface. Because of the fact that our gfx men didn't like this surface at all this project wasn't finished and not released in a public version. Mainly it is made to directly control the Eagleplayer (I don't think that I must describe its functions) and the secondary reason is that the analyzers are inegrated. (in a little more colorful way). The Levelmeter (LED-chain) is samplecontrolled, see also above for informations.

#### M.E.L.G.U

- separate doc

#### Space-Scope

- Lice looking sample shower which I saw first on an Atari ST Machine. Needs a lot of CPU-time because of large Blitting Area and lots of Points to be displayed within a short time. Looks best with
  - "Mod.load master loading" by Rebels,
  - "Mod.Heart\_of\_Rotterdam" by Greenhouse rewind,
  - "Mod.Breeze-Diablo" by Diablo/BudBrain,
  - "Mod.Disco Groove" by RSI or
  - "Mod.Madness took me" by Scoopex

"Mod.Boesendorfer P.S.S" by TRSI looks not bad as well (but sounds much better therefore)

## 1.33 keine programmierrichtlinien für userprogramme

### 4.3.4 Why didn't we release programming guides for coding the Userprograms ?

---

The biggest problem is that the Userprograms were already very complex in version 1.0 and from verion 1.10 on a lot of new functions have been added. Just to mention the possibility to develop completely surfaces. Our second big problem is a time problem (where to take from ??) and even if we would try to do our best writing this doc I don't expect that anybody would understand that stuff...

( Even we sometimes don't seem to have the look anymore :)--== )  
 Let's see, when I'm ready with my year in the army, before this I don't think...

---

## 1.34 einbindung des eagleplayers in andere programme

### 4.3.5 Binding the Eagleplayer into other programs

For Programmers only !!

~~~~~

The Eagleplayer offers the possibility to be called via it's Userprogramport from an extern source. To do this, you have to send it a "UM_Message" structure filled with certain values. An example for these Opportunitites is the Noise-converter which is distributed with this package.

What to do exactly ?

~~~~~

First find the Port:

```

move.l 4.w,a6
lea Portname(pc),a1 ;"EAGLEPLAYERPORT",0
jsr _LVOfindport(a6) ;find Eagleplayer Port
tst.l d0 ;test Result
beq.w .error ;Port not found
move.l d0,a4 ;save it

```

Second Create a Messageport (Under 1.3 you have to do this yourself)

```

move.l 4.w,a6
jsr _LVOcreatmsgport(A6)
move.l d0,d7 ;save it

```

Third you must get your own Task for getting the Message back

```

move.l 4.w,a6
suba.l A1,A1 ;Our Own Task
JSR _LVOfindtask(A6) ;find the Task
move.l d0,d6 ;save it

lea mymess(a5),a1 ;Pointer to UM_Message Structure
move.w #UM_sizeof-20,mn_length(a1) ;set the size of this Message struct
move.b #nt_message,ln_type(a1) ;Message-Type
move.l d7,mn_replyport(a1) ;Portadresse,an die
;zurückgesendet wird

move.w #-1,UM_UserNr(a1) ;everytime -1 !!!
move.l #USM_Externalprg,UM_Type(a1) ;Type of Usermessage: External Prg.
move.l d7,UM_Userport(a1) ;for Userprogram compatibility:
;Replyportaddress
move.l d6,UM_TaskAdr(a1) ;Save our Taskaddress

move.l #-1,UM_Signal(a1) ;No Signal
move.w #USClass_Command,UM_class(a1) ;Kind of Message: always a Command
move.l #0,UM_Userwindow(A1) ;Write here your Windows Address

```

```

        ;(for correct Requesterhandling)
        ;not required

move.l  #UCM_Playmem,UM_Command(A1) ;Type of Command: here
        ;to play a certain Memory range

moveq  #EPT_String+30,d0 ;allocate extra command structure
movem.l dl-a6,-(Sp)      ;for submitting extra Informations
move.l  4.w,a6           ;like Name, Address, Size or
moveq  #1,d1            ;other Infos depending on command
jsr  _LVOallocmem(A6)   (UCM_...)
movem.l (sp)+,dl-a6     ;Memorysize=Structure+Stringsiz
tst.l  d0               ; =EPT_String+???
beq  .Error2

move.l  d0,a2

move.l  a2,UM_ArgString(A1) ;Save this Structure into Message
move.l  #EPT_String+30,EPT_Stringsize(A2) ;Save the Size, IMPORTANT
clr.l  EPT_Next(A2)      ;no next Text
move.l  Sourceadr(a5),EPT_Result1(A2) ;Arg1, here Startaddress of
        ;mem to play
move.l  Sourcelen(a5),EPT_Result2(A2) ;Arg2, here Size of Mem to play

moveq  #30,d1           ;copy a Name for this Memory Range
lea.l  my_filename(a5),a3 ;e.g. "ripped using Eagleripper"
move.l  a2,-(sp)       ;not required
lea  EPT_string(A2),a2
.copyfilename
move.b  (a3)+,(A2)+
dbeq  dl,.copyfilename

move.l  (sp)+,a2

clr.l  UM_Result(A1)   ;no result !

move.l  4.w,a6
move.l  a4,a0          ;EAGLEPLAYERPORT
jsr  _LVOputmsg(a6)   ;send the Message

.wait
move.l  d7,a0
jsr  _LVOwaitport(A6) ;wait for the Port

move.l  d7,a0
jsr  _LVOgetmsg(A6)   ;get it back
tst.l  d0
beq.s  .wait
move.l  d0,a1
cmp.l  #USM_Externalprg,um_type(A1) ;our Message ?
beq.s  .ok
jsr  _LVOreplymsg(a6)
bra.s  .wait
.ok
move.l  UM_result(A1),d5 ;get Result (Errorcode: EPR_... or 0)

move.l  UM_ArgString(A1),d0 ;get Reply String (Our String was freed,

```

---

```

        ;this is an other one we must free)
    beq.s .notanswered
    move.l  d0,a1
    move.l  EPT_Result1(a1),temp1 ;Save Results
    move.l  EPT_Result2(a1),temp2 ;Save Results
    moveq  #28,d1
    lea  EPT_String(a1),a3
    lea  Stringtemp(a5),a2 ;Copy result String
.copy
    move.b  (a3)+,(a2)+
    dbeq  d1,.copy

    move.l  a1,d6      ;Free String-Structure(s)
.clrnext
    move.l  d6,a3
    move.l  EPT_Stringsize(A3),d0
    move.l  a3,a1
    move.l  EPT_Next(A3),d6
    move.l  4.w,a6
    jsr  _LVOfreemem(A6)
    tst.l  d6
    bne.s .clrnext
.notanswered
    move.l  d7,a0
    jsr  _LVODeleteMsgPort(a6) ;delete our Messageport
    rts
.error

```

This example plays a certain memory range, via the command "UCM\_GetModuledata" it is possible to "rip" a module from Eagleplayer. It returns Address and Size of the Memoryblock(s) in the EPT\_String Structures in the same way we did it. If somebody has problems implementing these routines, he should contact me, Buggs, I would send you the important parts of the Noiseconverter. See also the eagleplayer.i file where commands and structures are listed.

## 1.35 die playerbatch-datei

### 4.4.1 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-recognition 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





```

| state, too short, too long | Soundproggi this was
|                               | created with
|                               |
Audiochannels      | You've already loaded a      | Kill this Program and then
already used !     | program needing the Channels| select "RETRY"
|                               | (else "Cancel" for Break)
|                               |
Can't alloc Cia-   |The CiaB-Interrupt is used by| Kill this program and then
Interrupt         | another program, e.g.      | click on "play" again
| Music, Modem    |
|                               |
Error whilst De-   | File corrupt or too short    | try to depack with original
crunching        | Cruncher too new ?          | Packer or get new
|                               |
-+- with Fimp &   | Packed File longer than     | -+-
Crunchmania      | unpacked one                |
|                               |
Error adding      |File corrupt, not recognized, | check memory, file ok ?
Player           | no Memory, lower Version    | depack player !
| number          | delete already loaded
|                 | player and try again !
|                 | get new
|                 |
Extern Loading    |Files belonging to the music- | test paths and settings
failed           | file couldn't be loaded     | look if all files are in
|                 | the right directory
|                 |
Open Error       | Disk out of order,file not   | ?? look
Read Error       | on disk, wrong name given   |
Save Error       | ...                          |
Write Error      |                               |
|                 |
Module too short | Module too short or packed   | save more
| file shorter than given      |
|                               |
Not Enough Memory | No Mem,can't load module     | throw out everything dis-
|                               | turbing or buy more RAM
|                               |
Error loading Instr|One or more instruments could| does the Sample exist ?
truments         | not be loaded, not be found | Have you got enough Mem
| disk out of order, not      | free ?
| recognized                   |
|                               |

```

## 1.38 fehler des programms

### 5.2 Fehler des Programms

We're sorry that this program countains errors, also called "Bugs" ,which we couldn't remove because of time problems or we just didn't find them in the program.

For example the Userprograms crash on an A4000 Machine when you open the

About-Requester and click then on "OK". The problem is that the Userprograms don't work with ARQ1.66, a requester improver which was responsible for these crashes. Throw out this piece or get a newer version (1.78). :(-==

Some older Versions did not work correctly with RTPatch. Exact Error reports of older Versions you find in the

History  
section.

If you find errors in this program, please send us exact bugreports (Enforcer tested, if possible). (Machine, Memory, running programs, processor)

The Eagleplayer crashes when working with VZBE, a tool from the VirusZ-Package. But some other Programs crash as well so that we don't think this to be our fault.

A User wrote that the Eagleplayer should have problems with "NoClick". Couldn't be tested until now.

## 1.39 problemecke

### 5.3 Problem-Sheet

-----

Problem: Some players (e.g. all 3 TFMX replays) play too fast under NTSC or Productivity. The problem is that they are based on "VBlank"-Routines reacting on the framerate in case of speed. So if this one is higher than the normal one the player will be faster. To solve this problem is nearly impossible because of their concept, so use the TFMX replays 5 (yes, 5) Interrupts, horrifying....

Problem: Suddenly the Eagleplayer doesn't show the menus anymore although there are some hundred KBytes Chip-Memory free it can't load modules with a size of more than xx-KBytes, the CLI-Command "AVAIL" produces a Recovery Alert.

Why ?: Some replayers - especially the Soundtracker and its mutants - delete 2-4 Bytes at the beginning of every sample. If the modules are too short free memory will be discarded. The Problem is that directly after the end of one Memory block the header of the next block follows and exactly this one will be destroyed. There is nothing more to do than restart the Machine by resetting. We tried solving this problem by allocating 2 KBytes more memory (and modifying the playroutines) but in the last time the error occurred again, we don't know why.

A big problem is to bind the player into the multitasking operating system. The Userprograms will be started with a priority of -5. If there is another program permanently working (e.g. raytracer or packer) there is no spare multitask time for the Userproggis. The biggest hit is that the Eagleplayer everytime it begins to play or stops playing sends a message to the Userprograms and has to wait until it comes back, but what when no Userprogram gets its calculating time ??

So this procedure can last sometimes one or more seconds. If in any case

---

a Userprogram hangs itself up, the whole player waits until the end of the time.

The next bad message is that some internal/external players don't come along with different Machine-Configurations. Some examples therefore are TFMX-7V, Hippel and VSS.

Soundsystems with playroutine within the module make the following effect occur: It sometimes lasts some seconds until the module starts. This procedure is needed to make the module analyzer-compatible. Especially with Whittaker or Hippel you will find this problem.

It may happen out of this that rebuilding isn't possible because a playroutine not known to us will be used utilizing different routines than those we need. If such a case should appear, the player-specific will be enabled but nothing happens. (see Promizer)

In the test-phase there were problems with an earlier version of the domino-graphic card. There the Analyzermodes with 2 Bitplanes didn't work partially. HiFi system didn't work as well.

Problem: The Eagleplayer crashes again and again, as often as a program can crash. The Problem is that we pack our Player with Powerpacker and when it gets depacked the POWERPACKER DOESN'T RECOVER THE FILE 100% so that these crashes happen (Use Imploder instead). SHITTY ERROR !! LAME LAME LAME BÄH !

The Userprograms need a lot of CPU-calculating time when they're active. But some CPU-Meters write an incorrect value, which blames the Eagleplayer. Such one is the Directoryopus !! It shows complete shit. When the Userprograms are inactive almost no CPU-time is needed (except when EP scrolls) In other cases the Player is in waiting status. Check it out using Perfmon or Xoper !

## 1.40 einleitung zu den externer playern

### 6.1 EXTERNAL PLAYER PROGRAMMING INTERFACE

---

Who already knows how to program the Delitracker can skip the next chapters because the basic method was overtaken and is identical. The chapters 6.1-6.5 are the original Delitracker documentation (only some changes) The extras of the Eagleplayer are described from chapter 6.6 on.

Delitracker/Eagleplayer supports so called external players. These are executables in a special format so that they can be loaded from Delitracker/Eagleplayer.

## 1.41 das externe player konzept

### 6.1.1 The external player interface

---

External Players are executables (object files) which means they are relocated. At the start of an external player you can find the characteristic playerstructure. This structure is generated with a macro that can be found in 'misc/deliplayer.i'. There are two kinds of players: normal players and custom modules.

## 1.42 schematischer aufbau von externen playern

### 6.1.2 The structure of external players

---

Normal players can distinguished from custom modules by the existence of a check routine and the lack of the DTP\_CustomPlayer tag.

```
{ player header } identifies the objectfile as player.
{ tag array     } description of the playerfuntions.
{ interfacecode } playername/registerconversion/checkcode...
{ replaycode    } replay code itself
{ optional data } optional data
```

## 1.43 schematischer aufbau von custom modulen

### 6.1.3 The structure of custom modules

---

These are not modules in the conventional meaning. They are more like external players, the difference is that custom modules contain the player and the module. With the custom module interface you can adapt almost every module. If you have more modules with the same replay routine we suggest to write an own player for this moduleformat.

```
{ player header } identifies the objectfile as player.
{ tag array     } description of the playerfuntions.
{ interfacecode } playername/registerconversion/checkcode...
{ replaycode    } replay code itself
{ optional data } optional data
{ SOUND DATA   } music data (the module)
```

## 1.44 anpassung von playern

### 6.2 How to build external players

---

It is not difficult to adapt a player if you have the replayroutine. You only need to write some interfacecode. Delitracker/Eagleplayer has some helpful builtin routines that will make this job a lot easier. To adapt a new

---

soundsystem you need the replayroutine of that soundsystem, and at least 5 modules for testing the adaption.

## 1.45 playerheader

### 6.2.1 Playerheader

---

The PLAYERHEADER macro generates the header that identifies the file as a valid external player for Delitracker/Eagleplayer. This macro must exist and the player must begin with the macro. The only parameter you must supply is a pointer to a Tag Array that contains all functions that the external player supports. In all calls to player functions (except the interrupt routine) a5 will contain the address to the global player datastructure (Base). For more about this structure read 'misc/deliplayer.i'. In your routines (except in DTP\_Interrupt) you may trash all registers.

```
PLAYERHEADER <tagarray>
```

tagarray Pointer to a tag array, terminated with TAG\_DONE.

## 1.46 modulerkennung

### 6.2.2 Modulerecognition

---

In order to recognize the different moduleformats every player contains a specific routine, that tells Delitracker/Eagleplayer if the player can play this module or not. This routine has the task to check certain positions in the module that are identical in every module (like 'M.K.' at offset \$438 in NoiseTracker modules) or significant assembler instructions (if the module contains the player). Testing against one or two branches or jumps is NOT enough, cause many modules with player have branchtables at the beginning of the module. If the player recognizes the wrong module, a system crash will be the result! So be careful with the Ccheckroutine. As you can see the playerstructure has two checkfunctions but the player must use exactly one check routine. This leads to two basic player types:

a) Type one Player  
Here the Check1 function is implemented.

Advantage: You can implement players that can load the module by itself.

Disadvantage: No Packsupport, more complex.

This type should only be used if you REALLY need to load the module by yourself!

(e.g. FTM, IFF-8SVX player that loads the sample while playing, ...)

---

- b) Type two Player - the easier way  
Here the Check2 function is implemented.

Advantage: The module may be packed and Delitracker/Eagleplayer handles the loading and allocation of the module for you.

Disadvantage: The module is completely loaded into CHIP-Memory.

Regardless of the playertype the checkfunction must return d0.l=0 if the player can handle this module or d0.l<>0 if not.

## 1.47 interrupts

### 6.2.3 Interrupts

---

Players can be divided in two categories:

- a) Player that uses the internal timer interrupt from Delitracker/Eagleplayer

Advantage: The player is independent from the selected videomode  
The player has automatically the faster/slower function.

No expense for interrupthandling (interrupt structure and the insert/remove code).

Compatible with the serial.device.

Disadvantage: The interrupt is not synchronous to the VBlank (This leads to problems only in certain cases).

- b) Player that generates their own interrupt

Advantage: You can use other interrupt sources (like AudioIRQ)

Disadvantage: You have to handle the interrupts by yourself, and if you use VBlank the player is not independent to the videomode.

If you use your own timerinterrupt you should allocate CIAB because under 2.x the CIAA is used by the system. Do not to run the soundcode directly in the timerinterrupt. Instead you should Cause() a SoftInt in the timer interrupt and execute the routine in the SoftInt. This is to assure maximum compatibility for modem users because the SoftInt has a lower priority than the RBF interrupt. Beware of writing directly to the 680x0 intvectors! You should use AddIntServer() or SetIntVector() from the OS.

## 1.48 bedeutung der tags

### 6.3 Tags

---

In addition to the system tags (TAG\_DONE, TAG\_IGNORE, TAG\_MORE, TAG\_SKIP) this tags may be used:

DTP\_CustomPlayer (BOOL) - this tag identifies a player as customplayer.

If this tag is used the following tags are ignored:

- DTP\_PlayerVersion
- DTP\_PlayerName
- DTP\_Creator
- DTP\_Check1
- DTP\_Check2
- DTP\_ExtLoad
- DTP\_Config
- DTP\_UserConfig

DTP\_RequestDTVVersion (WORD) - only if the Delitracker/Eagleplayer version is greater than or equal to the requested version (ti\_Data) will Delitracker/Eagleplayer accept the player. If your player uses functions that were introduced in later revisions of Delitracker/Eagleplayer you must set this tag according to the version that introduced this function.

DTP\_RequestV37 (BOOL) - if this tag is set, only the Kick 2.0 version of Delitracker/Eagleplayer will load the player.  
(dtg\_GadToolsBase is valid)

DTP\_PlayerVersion (WORD) - Tag that contains the revision number of the player. If there are two players with same name the player with the higher version is used.

DTP\_PlayerName (STRPTR) - ti\_Data contains a pointer to the playername. This string may be as long as you wish, but only the first 24 chars are actually used. This tag must exist !

DTP\_Creator (STRPTR) - pointer to the author/adaptor name. This string is visible in the prefs window if the player is selected. The string may contain \$A as line separator.

DTP\_Check1 (FPTR) - pointer to a module identification routine. This routine is called after the first 1024 bytes of the module are loaded. If the module is shorter, the rest will contain zero. If the routine recognizes the moduleformat it must return d0=0 else d0<>0.

DTP\_Check2 (FPTR) - pointer to a module identification routine. This routine is called after the complete module is loaded (and decrunched). If the routine recognizes the module it must return d0=0 else d0<>0.

DTP\_Extload (FPTR) - pointer to a optional loadroutine for modules. If an error occurs return d0<>0 else d0=0. Please remember to free all allocated resources (memory, locks,...), because no further player function is called.

DTP\_Interrupt (FPTR) - pointer to a interruptroutine. This routine is called every 1/50 sec. via a timerinterrupt. Note: your



interruptroutine is not executed in the timerinterrupt itself. This is the standard method for gaining the correct playspeed regardless of the videomode. If the DTP\_Faster/DTP\_Slower pointers are not supplied, Delitracker/Eagleplayer emulates this by changing the interrupt frequency. If this tag doesn't exist, you must supply DTP\_StartInt/DTP\_StopInt.

DTP\_Stop (FPTR) - pointer to optional stop routine. If this tag does not exist, Delitracker/Eagleplayer uses the following standard method:

```
stop interrupt (DTP_StopInt)
cleanup sound (DTP_EndSnd)
reinitialize the song (DTP_InitSnd)
```

This routine will stop playing the song, reset the 'patterncounter' to the begin and change the playspeed to default. This means that the interrupt is started again and the song should begin to play from the beginning.

DTP\_Config (FPTR) - pointer to an optional initialising routine. This routine is only called once after the player is loaded.  
Purpose: The player could load a specific configfile.

DTP\_UserConfig (FPTR) - pointer to a optional initialising routine. This routine is called if the user selects the 'Config' button in the prefswindow. Purpose: The player could open a player specific configwindow for setting special options (e.g instrumentpath for a sonix player) and saving them into a configfile.

DTP\_SubSongRange (FPTR) - This tag should be supplied if the player supports multimodules. ti\_Data points to a function that returns in d0 the minimum and in d1 the maximum subsong number.

DTP\_InitPlayer (FPTR) - pointer to an initialising routine, that is called if a module is loaded successfully. Must return d0=0 if all is ok else d0<>0. The audioallocation must be done here. (Delitracker/Eagleplayer has a function that does the allocation.) If the player supports subsongs it has to set dtg\_SndNum(a5) to the first subsongnumber. If a routine for DTP\_SubSongRange exists, Delitracker/Eagleplayer performs this for you and you may omit the initialization for dtg\_SndNum(a5).

DTP\_EndPlayer (FPTR) - pointer to a cleanuproutine, that is called if the module is removed from memory. Audiochannels have to be freed here. (Use the Delitracker/Eagleplayer internal supportroutine)

DTP\_InitSound (FPTR) - pointer to an optional initialising routine. This routine has the task to (re)initialize the module. If the interrupt is started the song should begin to play at the beginning.

DTP\_EndSound (FPTR) - pointer to an optional cleanuproutine. For example

it can be used to reset the volumeregister or the audiodma.

DTP\_StartInt (FPTR) - pointer to an initialising routine, that must exist if DTP\_Interrupt doesn't exist. It has the task to start the sound.

DTP\_StopInt (FPTR) - pointer to a cleanuproutine, that must exist if DTP\_Interrupt doesn't exist. It has the task to stop the sound.

DTP\_Volume (FPTR) - pointer to function that sets the volume. This function is called every time the volume is changed (via arexx or slider) and once at the initialising phase of the module (before DTP\_InitSnd is called). The mastervolume can be found in dtg\_SndVol(a5). The mastervolume is the highest volume allowed. The effective volume can be calculated using the following formula:

$$\text{VOL\_eff} = ( (\text{MASTERVOLUME} * \text{modulevolume}) \gg 6 ).$$

See also the example sources.

DTP\_Balance (FPTR) - pointer to a function that sets the balance. This function is called every time the balance is changed (via arexx or slider) and once at the initialising phase of the module (before tf\_InitSnd is called). The balance for the left channel can be found in dtg\_SndLbal(a5), for the right channel in dtg\_SndRbal(a5). Note: All players that support balance are capable of volume too! Then you must use the same routine for both operations. The mastervolume for the left channels can be calculated with this formula:

$$\text{LeftMaster} = ( (\text{dtg\_Volume}(a5) * \text{dtg\_SndLbal}(a5)) \gg 6 ).$$

For the right channels the formula is similar.

DTP\_Faster (FPTR) - pointer to a function that increases the playspeed.

DTP\_Slower (FPTR) - pointer to a function that decreases the playspeed.

DTP\_NextPatt (FPTR) - pointer to a function that increases the patternpointer.

DTP\_PrevPatt (FPTR) - pointer to a function that decreases the patternpointer.

DTP\_NextSong (FPTR) - pointer to a function that increases the subsongcounter (only if the subsong exists).

DTP\_PrevSong (FPTR) - pointer to a function that decreases the subsongcounter (only if the subsong exists).

DTP\_SubSongTest (FPTR) - (till version 1.35) The tag is only evaluated if DTP\_SubSongRange exists too. ti\_Data points to a routine that returns a boolean value. This indicates if the subsong number dtg\_SubNum(a5) is valid (d0=0) or not (d0<>0). This tag is only necessary for players where not every subsong in the subsong range is existant.

---

## 1.49 delitracker support funktionen

### 6.4 Delitracker/Eagleplayer support functions

---

Delitracker/Eagleplayer provides some support functions that can be called from the external player. Every function is called like this:

```
move.l    dtg_XXX(a5),a0    ; a5 must contain the base
jsr (a0)
```

All functions (except dtg\_SongEnd/dtg\_SetTimer) use d0/d1/a0/a1 as scratch register. A5 must contain the base (except dtg\_SongEnd/dtg\_SetTimer). Currently the following functions are available:

dtg\_GetListData

SYNOPSIS

```
memory size = dtg_GetListData(number)
a0    d0    d0.l
```

FUNCTION

Returns the address and the length of a file that was loaded with dtg\_LoadFile().

INPUTS

number - number of the file beginning with 0 for the file that was selected by the user.

RESULT

memory - startaddress of the files in memory, if error 0.  
size - length of the loaded file in bytes or 0 in case of an error

dtg\_LoadFile

SYNOPSIS

```
success = dtg_LoadFile(name)
```

FUNCTION

Loads and decrunches the specified file to chipmemory.  
Note: this function automatically adds '.pp' to the filename.

INPUTS

name - store the filename in the internal buffer  
(dtg\_PathArray contains a pointer to this buffer)

RESULT

success - success d0.l=0, else d0.l<>0.

dtg\_CopyDir

---

## SYNOPSIS

```
dtg_CopyDir()
```

## FUNCTION

Copies the directory of the selected file at the end of the string, that dtg\_PathArray points to.

```
dtg_CopyFile
```

## SYNOPSIS

```
dtg_CopyFile()
```

## FUNCTION

Copies the filename of the selected file at the end of the string, that dtg\_PathArray points to.

```
dtg_CopyString
```

## SYNOPSIS

```
dtg_CopyString(string)
    a0
```

## FUNCTION

a0 contains the address of a string, which is copied at the end of the string that dtg\_PathArray points to.

## INPUTS

string - a0 contains the pointer to the string

```
dtg_AudioAlloc
```

## SYNOPSIS

```
success = dtg_AudioAlloc()
```

## FUNCTION

Allocates the audiochannels

## RESULT

success - if we got them: d0.l=0, else d0.l<>0.

```
dtg_AudioFree
```

## SYNOPSIS

```
dtg_AudioFree()
```

## FUNCTION

Frees the audiochannels that were allocated with dtg\_AudioAlloc.

```
dtg_StartInt
```

---

## SYNOPSIS

```
dtg_StartInt()
```

## FUNCTION

Starts the soundinterrupt. If DTP\_Interrupt exists, Delitracker/Eagleplayer starts the internal timerinterrupt, else DTP\_StartInt is called.

```
dtg_StopInt
```

## SYNOPSIS

```
dtg_StopInt()
```

## FUNCTION

Stops the soundinterrupt. If DTP\_Interrupt exists, Delitracker/Eagleplayer stops the internal timerinterrupt, else DTP\_StopInt is called.

```
dtg_SongEnd
```

## SYNOPSIS

```
dtg_SongEnd()
```

## FUNCTION

Signals Delitracker/Eagleplayer, that the module was played once. This function doesn't change any registers and is save to call from interrupts.

```
dtg_CutSuffix
```

## SYNOPSIS

```
dtg_CutSuffix()
```

## FUNCTION

removes the suffix '.pp', '.im', '.xpk' at the end of the string, that dtg\_PathArray points to.

```
dtg_SetTimer
```

## SYNOPSIS

```
dtg_SetTimer()
```

## FUNCTION

programs the CIA-timer with the value that is stored in dtg\_Timer(a5). This function doesn't change any registers and is save to call from interrupts.

## 1.50 tips zur anpassung

---

## 6.5 Hints and Tips

---

The player should not change the LED condition because Delitracker/Eagleplayer will handle it.

This is a small list that you should match when you create your own player or custom module.

- [ ] checkroutine exact enough ?
- [ ] audiochannels allocated/freed correctly ?
- [ ] all allocated memory freed after playing?
- [ ] all locks unlocked after playing ?
- [ ] enforcer and mungwall proof ?
- [ ] viable error handling path taken for all possible errors ?
- [ ] player tested under 1.3 and 2.0 ?
- [ ] does the player work correct in all videomodes ?

### Problems

| Symptom                | possible source                                                                                                                                                                            | elimination                          |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| crash                  | a5 trashed, other registers<br>not saved, wrong stack usage<br>initialization wrong or too late<br>module too new for the replayroutine                                                    | better check                         |
| module sounds<br>wrong | audio data not in chipmem<br>wrong initialization<br>module too new for the replayroutine<br>player needs some special values<br>in some registers      extra init code<br>wrong videomode |                                      |
| no sound               | Audio-DMA off      :-)<br>>68000 player writes directly to processor<br>intvectors and VBR is not 0.                                                                                       | use OS routines to<br>set IntVectors |
| no free<br>CPU-time    | wrong interrupt handling<br>Z-Flag must be set<br>at the end of the<br>interruptroutine.                                                                                                   | VBlank-IRQ:                          |

## 1.51 die besonderheiten und neuerungen des eagleplayers

### 6.6 The special informations and news for the Eagleplayer

---

The Eagleplayer offers a lot of new possibilities, which are described within the following chapters. The 3 TAGS listed below are usable for custom modules: (the Delitracker ignored them)

DTP\_PlayerName

---

```
DTP_Creator
DTP_ExtLoad
```

The most important news are the new  
Eagleplayer-Tags

the

```
,
ModuleInfo-Function
,
Analyzersupport
and the new
Globals
.
```

## 1.52 die neuen tags

### 6.6.1 The new Tags

-----

EP\_Get\_ModuleInfo (FPTR) - This function of the replayer has to return within the register A0 a fully initialized Moduleinfo-taglist or zero for no taglist.

see also chapter 6.6.2 Moduleinfo

NOTE: it's possible that this function is called when no module was loaded, think of that !!

EP\_Free\_ModuleInfo (FPTR) - This optional function is for example to free previously allocated memory for the Moduleinfo-taglist. No return values expected.

EP\_Voices (FPTR) - using this function the informations about the 4 voices will be given by the Eagleplayer in D0. If the bit for the voice set, the voice is enabled, if the bit was cleared, the voice is disabled  
Bit 0 = Voice 0; Bit 1 = Voice 1; Bit 2 = Voice 2;  
Bit 3 = Voice 3

EP\_Structinit (FPTR) - returns in A0 a pointer to a UPS\_USER structure in which the replayer puts the informations for the Analyzerprograms (Userprograms) or ZERO. This structure will then be given to all Userprograms which use the infos of it.

EP\_StructEnd (FPTR) - optional, e.g. to free the memory used for the structure

EP\_LoadPlConfig (FPTR) - calls the routine loading the configuration of this replayer (usually 'ENV:Eagleplayer/xxx.config' )

EP\_SavePlConfig (FPTR) - calls the routine saving the configuration of this replayer (usually 'ENVARC:Eagleplayer/x.config' )  
(advise of the authors)

EP\_GetPositionNr (FPTR) - returns in D0.1 the actual Pattern-number played at this time. This value will be scrolled into the Main window of the Eagleplayer

EP\_SetSpeed (FPTR) - for players with own timer this function gives you the possibility using the speed-propgadget. Therefore in D0 there will you find the actual speed value between -25 and +25. 0 means the standard speed, if the speed should be higher, this value is positive, for lower

speed negative

EP\_Flags (LONG) - These flags within "TI\_Data" of this tag show which functions the player basically supports and must exist !!

This Tag was included because we wrote "self modifying" players which change the module to support Analyzer,Voices, Volume...

When the internal replays of this module differ to another then it may happen that the replayer can't change the module in this way we need it so that one or the other function won't be available from module to module.

Therefore the following flagbits for showing the basic abilities of the replayer in the Player window:

EPF\_Songend - the replayer supports Songend  
 EPF\_Restart - the module is restartable (partially impossible)  
 EPF\_Disable - The player is disabled  
 EPF\_NextSong - supports next subsong  
 EPF\_PrevSong - supports previous subsong  
 EPF\_NextPatt - supports jump to next pattern  
 EPF\_PrevPatt - supports jump to previous pattern  
 EPF\_Volume - volumecontrol possible  
 EPF\_Balance - balance possible  
 EPF\_Voices - can change the state of the voices  
 EPF\_Save - module can be saved  
 EPF\_Analyzer - Analyzer-support  
 EPF\_ModuleInfo- can tell infos to the current module  
 EPF\_SampleInfo- infos to the samples of the current mod  
 EPF\_Packable - module may be packed  
 EPF\_VolVoices - volume of each voice can be different to the next (EPG\_VolVoicel...)

EP\_KickVersion (WORD) - minimal Kickstartversion (37 for Kick 2.0)

EP\_PlayerVersion (LONG)- minimal Eagleplayerversion, if the replayer may only be loaded by the Eagleplayer you must specify "DTP\_RequestDTVVersion,\$7fffffff" additionally

EP\_EjectPlayer (FPTR) - if the replayer allocates any memory, filelocks etc. it can be freed at this point. Then the replayer will be thrown out of the memory.

EP\_Date (LONG) - date of creating this replayer e.g.

dc.l EP\_Date

dc.b 04,11,19,93 -> November the 4th in 1993

EP\_Check3 (FPTR) - Checkroutine, called after loading the first 1000 bytes of the module. If it was recognized it will be loaded into fastmem.

NOTE: Using this check you may not pack the module, so if you wanna also packed modules, use DTP\_check2 additionally.

## 1.53 moduleinfo



### 6.6.2 Moduleinfo

-----

For the moduleinfo-function the Eagleplayer offers some Tags which can give the user almost all informations about the actual module. The taglist will be overtaken by the Eagleplayer it calls the routine specified in the "EP\_Moduleinfo" Tag within A0 (see also chapter 6.6.1). Please notice that currently not all informations will be displayed in the scroller of the main window. In a future version there will be a window telling these informations completely.

MI\_SongName (STRPTR) - Name of the module (like in the sound/protracker ones)  
If you give a ZERO to TI\_Data an "unknown" appears as songname. This is a very comfortable method getting the right name of ripped modules.

MI\_AuthorName (STRPTR) - Name of the guy writing this song. If you return a ZERO in TI\_Data, the Eagleplayer displays an "unknown".

MI\_SubSongs (LONG) - Number of subsongs in the module

MI\_Pattern (LONG) - Number of patterns in the module

MI\_MaxPattern (LONG) - Max. number of patterns (e.g. Soundtracker: 64)

MI\_Length (LONG) - length of the song (e.g. in Patterns) <- NOT the SIZE

MI\_MaxLength (LONG) - max. length of the songs (e.g. Soundtracker 127)

MI\_Steps (LONG) - number of steps (e.g. BP Soundmon)

MI\_MaxSteps (LONG) - Max. number of steps

MI\_Samples (LONG) - number of used samples

MI\_MaxSamples (LONG) - Max. number of used samples (e.g. Protracker: 31)

MI\_SynthSamples (LONG) - Number of used synthetic samples

MI\_MaxSynthSamples (LONG) - max. number of synthetic samples

MI\_Songsize (LONG) - size of the Song in Bytes (without samples !)

MI\_SamplesSize (LONG) - size of the samples in Bytes

MI\_ChipSize (LONG) - used Chip Memory in Bytes

MI\_OtherSize (LONG) - used Fast Memory in Bytes

MI\_Calcsize (LONG) - calculated size of the whole module in bytes

MI\_SpecialInfo (STRPTR) - pointer to special informations as ASCII-String

MI\_LoadSize (LONG) - Overall size of all files loaded (for Soundsystem loading more than one file, e.g. Startrekker,TFMX)

MI\_Unpacked (LONG) - Unpacked size in bytes (e.g. how long would a Noisepacker-

module when converting back to protracker)

MI\_UnPackedSystem (LONG) (STRPTR) - tells to what soundsystem this module normally belongs if it was packed (e.g. with Noisepacker, Propacker, Promizer), either an internal number (see below) or a pointer to a string containing the name

The following names are internayl:

```
MIUS_OldSoundtracker
MIUS_Soundtracker
MIUS_Noisetracker
MIUS_Protracker
```

MI\_Prefix (STRPTR) - Pointer to the prefix for the name of the module, e.g. 'MOD.' or 'Mdat.'. So is it possible to save the module with the real name and a matching prefix.

MI\_About (STRPTR) - Pointer to a text about the player

MI\_MaxSubSong (LONG) - max. number of subsongs for this sound format

MI\_Voices (LONG) - Number of used voices in this format

MI\_MaxVoices (LONG) - Number of max. possible Voices in this format.

## 1.54 analyzersteuerung

### 6.6.3 Analyzer control

-----

The Analyzercontrol will be done via the UPS\_User structure described below. (how to offer the structure- see chapter 6.6.1 "EP\_Structinit)

The structure looks this way:

```
STRUCTURE UPS_USER,0

  APTR      UPS_Voice1Adr
  UWORD     UPS_Voice1Len
  UWORD     UPS_Voice1Per
  UWORD     UPS_Voice1Vol
  UWORD     UPS_Voice1Note
  UWORD     UPS_Voice1SampleNr
  UWORD     UPS_Voice1SampleType
  UWORD     UPS_Voice1Repeat

  LABEL     UPS_Modulo

  APTR      UPS_Voice2Adr
  UWORD     UPS_Voice2Len
  UWORD     UPS_Voice2Per
  UWORD     UPS_Voice2Vol
```

```
UWORD    UPS_Voice2Note
UWORD    UPS_Voice2SampleNr
UWORD    UPS_Voice2SampleType
UWORD    UPS_Voice2Repeat

APTR     UPS_Voice3Adr
UWORD    UPS_Voice3Len
UWORD    UPS_Voice3Per
UWORD    UPS_Voice3Vol
UWORD    UPS_Voice3Note
UWORD    UPS_Voice3SampleNr
UWORD    UPS_Voice3SampleType
UWORD    UPS_Voice3Repeat

APTR     UPS_Voice4Adr
UWORD    UPS_Voice4Len
UWORD    UPS_Voice4Per
UWORD    UPS_Voice4Vol
UWORD    UPS_Voice4Note
UWORD    UPS_Voice4SampleNr
UWORD    UPS_Voice4SampleType
UWORD    UPS_Voice4Repeat

UWORD    UPS_DMACon

UWORD    UPS_Flags
UWORD    UPS_Enabled
UWORD    UPS_Reserved

LABEL    UPS_SizeOF
```

These entries have the following meanings:

UPS\_Voice?Adr - Address of the sample played currently on this voice  
(must stay here as long as the sample is playing !)

UPS\_Voice?Len - size of the sample played on this voice

UPS\_Voice?Per - actual value of the sample period, plays a very important role, if a period value <>0 is set here it means that a new note was played. So the period is the most important thing for the analyzers. If you can't find out when you play a new Note, please set the period every time you access the audio hardware (\$dff0A6/B6/C6/D6), this is also advised for Samplesize and Address

UPS\_Voice?Vol - Volume to be written to the hardware, do not write here the volume you calculate from your own one and the volume currently set in the Eagleplayer, that means when e.g. the volume you will set basically is 64 but the Eagleplayer's volume is only 32 I don't wanna see 32 but 64, ok ? The other way will be used with another method (Mastervolume !)

UPS\_Voice?Note- not supported yet

UPS\_Voice?Samplnr - tells the actual samplenummer, neither supported yet by a userprogram nor by a playroutine

---

UPS\_Voice?Samplotype - Samplotype, not supported yet

UPS\_Voice?Repeat - tells if the sample will be played once or with repeat, if this value is 0 that means repeat on, if it is 1 the repeating of the sample is off

UPS\_DMACon - tells which voices are on, which are off, Bit 0 for voice 0, bit 1 for voice 1 and so on, if the bit set the channel is switched on (the name "UPS\_DMACON" is a little bit confusing, better was "UPS\_Voices", see also at chapter 6.6.1. "EP\_Voices)

UPS\_Flags - Flagbits telling which possibilities of the UPS\_User structure the replayer supports

UPSFL\_Adr - Sampleaddress

UPSFL\_Len - SampleSize

UPSFL\_Per - Sampleperiod (IMPORTANT!)

UPSFL\_Vol - Volume

UPSFL\_Note - Note, not supported yet

UPSFL\_SNr - Samplenummer

UPSFL\_STy - Samplotype, not supported yet

UPSFL\_DMACon - which voices are off/on

The rest of the entries is reserved for future versions of the Eagleplayer.

----- Attention -----  
 For the todays Userprograms it is expected that at least UPSF\_Adr, UPSF\_Len, UPSF\_Per, UPSF\_Dmacon and UPSF\_Vol are set and supported. The other parameters (UPS\_Voice?Adr, UPS\_Voice?Len, UPS\_Voice?Per and UPS\_Voice?Vol) have also to be set and UPS\_Enabled should always be "0" on leaving the interrupt !  
 -----

## 1.55 die neuen eagleplayer-globals

### 6.6.4 The new Eagleplayer Globals (from Eagleplayer 1.10 on)

-----

After trying not to do any changes within the globals it has become nessesary to add some subroutines,pointers and data value cells.

Using the new Eagleplayer-globals you don't have to use registers to deal with arguments but to enter the values into one of the 8 globals argument cells. Into EPG\_ArgN you always have to write the number of arguments. A subprogram always must get all the arguments required and so EPG\_ArgN must countain always the maximum value telled at "Inputs:" - see below. If the parameters will be given in another way it will be described separately.

The Subroutines of the Eagleplayer may called everytime by every replayer but NOT IN THE INTERRUPT ! If not told extra the Userprograms may only access the subroutines only after an USCLASS\_LockEP call.

---

Now follow the most important subroutines:

----- EPG\_SaveMem -----

A memory range will be saved using the savemode set in the actual preferences or by your own choice. This function will only work in the registered version. If the savemode is -1 the standard one set by the user will be taken.

Input: Arg1 = Startaddress  
Arg2 = Endaddress  
Arg3 = Stringpointer to full filename including the path  
Arg4 = SaveMode (-1=Eagleplayer settings  
0=not packed  
1=PP-Crunched  
2=LH-Crunched  
3=XPK-Crunched  
ArgN = 4

Output: Arg1 = Return value (0=no error)

----- EPG\_FileRequest -----

A filerequester will be opened using the default Filerequester mode. You may differ between a fileselection and a directory selection..

Input: Arg1 = Filerequester Titlename  
Arg2 = Directory Path  
Arg3 = Filename  
Arg4 = Window  
Arg5 = Filerequestertype (1=Fileselect 0=Dirselect)  
Arg6 = OutPut-Text for Eagleplayer-Statuswindow  
ArgN = 6

Output: Arg1 = Return value (0=Cancel or System error, else 1)

----- EPG\_TextRequest -----

A textrequester will be opened. You write the Address of an ASCII-text to the EPG\_Arg1. The window opened will be adapted to the size of the text. Within the text you may add arguments which are taken from the argument table. You may use as much gadgets as you want and you may use own Images in this window. Under OS2.0 and higher public screens are supported.

Note: This routine contains a bug not found yet. It should be used anyway because this error will be fixed in one of the next versions.

Input: Arg1 = TextAddress  
Arg2 = Pointer to Pubscreenname (only OS2.0+, else 0)  
Arg3 = Position on Screen (x.w & y.w)  
Arg4 = Pointer to Gadgetnames

---

Arg5 = Pointer to Requestername  
Arg6 = Pointer to ArgumentList  
Arg7 = Pointer to ImageDatas  
ArgN = 7

Arguments in ASCII-String: %s - String  
                  %d - Decimal number

Output: Arg1 = Return value   0=Error (e.g. Window too big)  
          else number of chosen gadget

----- EPG\_LoadExecutable -----

This function loads an executable program. This will be  
depacked, if possible.

Input: Arg1 = FilePath  
       ArgN = 1

Output: Arg1 = Jump Adr of the Program  
       d0    = Error (0=ok)

----- EPG\_NewLoadFile -----

Like DTG\_LoadFile, except that the memory type can be supplied

Input: Arg1 = Memory type (like \_LVOALLOCMEM - exec library)  
       ArgN = 1  
       DTG\_PathArrayPtr = Path of the file

Output: d0 = Return value (0=ok)

-- EPG\_ScrollText --

Scrolls the supplied text into the main window of the Eagle-  
player. If the text is ZERO terminated it stops when it was  
completely scrolled in, if it is 1-terminated it will be  
scrolled in a loop, if a 2 is at the end of the text it will  
be scrolled until it reaches the left side if the text is not  
already there.

Input: Arg1 = Textaddress  
       ArgN = 1

----- EPG\_LoadPlConfig -----

Loads the configuration of a player. This function was added  
for solving the problem to press cancel x-times when the  
"ENV:" directory could not be found and x players need a  
config from "ENV:". It will be tested if the config should  
exist in the ENV-Dir or not. If the directory doesn't exist  
the config won't be loaded.  
(not built in yet !)

Input: Arg1 = ConfigPath+Filename ("ENVARC:eagleplayer/xxx.config" !!!)  
ArgN = 1

Output Arg1 = Return value

----- EPG\_SavePlConfig -----

Saves a PlayerConfig.  
(not built in yet !)

Input: Arg1 = ConfigPath+file ("ENVARC:eagleplayer/xxx.config" !!!)  
Arg2 = Startaddress  
Arg3 = Endaddress  
Arg4 = SaveMode (see also "EPG\_SaveMem")

Output: Arg1 = Return value (0=ok)

----- EPG\_FindTag -----

Seeks a Tagitem within the given TagList. This Function depends  
not on the kickstart and may also called from userprograms  
without "USCLASS\_LockEP" at any time.

Input: a0 = Taglist  
d0 = Tag

Output: d0 = Ti\_Data's value  
d1 = Tag found (0=no,then d0 also 0, 1=yes)

----- EPG\_FindAuthor -----

Seeks the Author of a music file within the supplied range.  
This routine was written for the Soundtracker and its mutants  
but may also used for other Soundsystems. This routine seeks  
after "by" and "#" in the samplename. The next string will be  
taken as the authername. This name doesn't have to be copied  
but it is enough writing this address to the Moduleinfo Tag.

Input: Arg1 = Start of the first samplename  
Arg2 = Offset to the next samplename  
Arg3 = Size of the samplename  
Arg4 = number of samples  
ArgN = 4

Output Arg1 = Pointer to Authorstring or ZERO  
Arg2 = Size of Authorstring or ZERO

----- EPG\_Voice1Vol - EPG\_Voice4Vol -----

at these word-orientated addresses you will find the volumes  
for each of the hardware voices, pre-calculated from Volume,  
Balance and Voices. So you only need one routine for vol etc.  
instead of 3 !

to use these values the following example:

```
move.w 20(a0),d0 <- example for the volume of sample  
for voice 2
```

```

mulu   EPG_volvoice2(A5),d0 <- calculate the volume
        depending on settings
lsr.w  #6,d0          <- final value
move.w d0,$dff0a8 <- write to the hardware

```

Out of this the found and used librarybases are saved into the globals. Don't CHANGE ANYTHING IN THE GLOBALS !! (except EPG\_ARG1 - EPG\_ArgN)

## 1.56 einführung und beschreibung des arexx-portes

### 7.1 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.

An

Adaption to DOPUS  
is therefore no problem at all.

## 1.57 arexx-kommandoübersicht

### 7.2 The commands

---

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 !

The arguments in such "[]" brackets are optional, in "<>" brackets must be supplied !

If we talk filenames from now on we mean FILE AND PATH ! (if no other case is mentioned)

| Command | Meaning |
|---------|---------|
|---------|---------|

---

|                       |                                                                                    |
|-----------------------|------------------------------------------------------------------------------------|
| LoadModule [Filename] | loads the supplied module and plays it if "QUICKSTART" in the prefs (see at STATUS |
|-----------------------|------------------------------------------------------------------------------------|

---



command) is switched on. Else you must call "play" before it starts. If no filename was supplied, a filerequester will appear

LoadModule <Dirname> <Filename> <Filename> ...  
puts the given modules into a list which you can see in the Pulldown-menu. The Eagleplayer doesn't read the whole directory. The filename is given without path !

SaveModule [Filename] The loaded module will be saved under the supplied name. If no argument is given a filerequester appears. (r.V.)

About Module returns the scroltext for displaying the moduleinformations

EjectModule kills the module out of memory

Stop The Player stopps. Calling "play" would play it from the beginning on.

ReplaySong Stops playing and then restarts from the beginning of the module

Play The Eagleplayer starts playing

Pause The playing stops. If you select "play" the module continues at the point where it was paused

NextSong the next subsong will be played

PrevSong the previous subsong will be played

NextModule the next module of the list will be played

PrevModule the previous module of the list will be played

NextPattern the next pattern will be played

PrevPattern the previous pattern will be played

PubScreen <PubScreen> opens the Eagleplayer on the supplied PublicScreen.

Quit The Eagleplayer will be closed and removed from memory.

AboutEP Returns a String about the Eagleplayer in RX\_Result.

ScrollText <ScrollText> The supplied text scrolls into the main window.

Iconify The Eagleplayer will be iconified or if it already was in iconify mode, it will open

its window(s) again.

HelpMe <Nr> Returns the help string for the given function

DeleteFile [Filename] Deletes the supplied file.

UserPrg <UserPrgNr> <state> Switches the given userprogram on/off

LoadConfig <Filename> Loads a new Eagleplayer-Config.

SaveConfig <Filename> Saves the actual Eagleplayer-Config.

Font <Fontname> loads a new font for the main window

OpenMainWindow the main window will be opened

OpenPlWindow the playerinfo window will be closed

CloseMainWindow the main window will be closed

ClosePlWindow the playerinfo window will be closed

LittleWin switches the iconify mode to "little window"

AppItem switches the iconify mode to "AppItem"

AppIcon switches the iconify mode to "AppIcon"

Volume <Number> sets the supplied volume value.

Balance <Number> sets the supplied balance value.

TimeOut <Time> After the given delay (secs) value the player  
will stop

SetSpeed <Number> sets the given speed value

DefaultSpeed reset the speed to the normal one

SlowerSpeed the speed will get a bit lower

FasterSpeed the speed will get a bit higher

Voice <Nr> <state> Switches the state or the voice on/off

AddPlayer [Filename] the filename of the supplied replayer will  
be used for loading it, if no filename was  
specified, a filerequester will appear.

AddPlDir [Dirname] The supplied directory will searched for  
replayers and found ones will be loaded. If  
no directory name was given a filerequester  
will open.

DeleteAll deletes all external Replayers.

DeletePlayer <Nummer> the player with the given number will be

---

deleted

Enable <Nummer>      The given player will be enabled

Disable <Nummer>      The given player will be forbidden

LoadPlConfig <Nummer>      the Config of this Player will be loaded

SavePlConfig <Nummer>      the Config of this Player will be saved

PlConfig <Nummer>      the configuration routine of this player will  
be called

AddUserPrg [Filename]      The given userprogram will be loaded. If no  
filename is given a filereqeuster will open

HelpMode <state>      the HelpMode will switched to on/off

FadeOut <On/Off>      FadeOut will be switched on/off

FadeIn <On/Off>      FadeIn will be switched on/off

Filter <On/Off>      the Audio filter will be switched on/off

Quickstart <On/Off>      Quickstart will be switched on/off

LoadDir <On/Off>      LoadDir will be switched on/off

AutoDir <On/Off>      AutoDir will be switched on/off

LoadSubDirs <On/Off>      LoadSubDirs will be switched on/off

UseSongName <On/Off>      Use Songname will be switched on/off

Useprefix <On/Off>      Use Prefix is switched on/off

SaveT <On/Off>      SaveT is switched on/off

QuitSurface <On/Off>      Quit Surface is switched on/off

QuitEagle <On/Off>      Quit Eagle is switched on/off

AutoPassword <On/Off>      Auto Passwort is switched on/off

EjectPlayers <On/Off>      Eject Players is switched on/off

UsePlBatch <On/Off>      PlayerBatch is switched on/off

LoadPlBatch [Filename]      A PlayerBatch-File will be loaded  
If no filename is supplied, a filerequester  
will appear.

EjectUserPrg <On/Off>      Eject Userprograms is switched on/off

UseUPrgBatch <On/Off>      UserprogrammBatch is switched on/off

LoadUPrgBatch [Filename]      A UserProgrambatch-File will be loaded

---

If no filename is supplied, a filerequester will appear.

ScrollInfos <On/Off> ScrollInfos will be switched on/off

FlashPointer <On/Off> Flashpointer will be switched on/off

MasterVolume <On/Off> Mastervolume will be switched on/off

SongEnd <On/Off> Songend will be switched on/off

LoadNext <On/Off> LoadNext will be switched on/off

AutoSubSong <On/Off> AutoSubSong will be switched on/off

RandomModule <On/Off> Randommodule will be switched on/off

RandomSong <On/Off> Randomsong will be switched on/off

EPDir <On/Off> EPDir is switched on/off

Notify <On/Off> Notify is switched on/off

ScanAlways <On/Off> Scan Always is switched on/off

ShowDirNames <On/Off> ShowDirNames is switched on/off

HideAll <On/Off> HideAll is switched on/off

CheckDifference <Number> Not implemented yet

EntriesPerMenu <Number> Set max. entries for modules directory. (0..63)

Menumem <Number> Change Size of Directory Memory.

NoNewModule The program mode is set to ...

PrevModule The program mode is set to ...

NextModule The program mode is set to ...

RandomModule The program mode is set to ...

LoadBefore <On/Off> Load Before is switched on/off

LoadAlways <On/Off> Load Always is switched on/off

SmallModule <On/Off> Warning on Small Modules is switched on/off

BigModule <On/Off> Warning on Big Modules is switched on/off

ModuleFault <On/Off> Warning on corrupt modules is switched on/off

LittleWin Switches the Iconify-Mode to Little Window

AppItem Switches the Iconify-Mode to AppItem

AppIcon Switches the Iconify-Mode to AppIcon

Unpacked New Crunchmode for saving set

PPpacked New Crunchmode for saving set

LHpacked New Crunchmode for saving set

---

```

XPKpacked      New Crunchmode for saving set
CRMPacked      New Crunchmode for saving set

Overwrite <On/Off>  Overwrite is switched on/off
AutomaticSave <On/Off>  AutomaticSave is switched on/off
SSDir <On/Off>    SSDir is switched on/off
Protect <On/Off>   Protect is switched on/off
PBits <On/Off>    PBits is switched on/off
SampleMode <On/Off>  SampleMode is switched on/off
SaveAsPt <On/Off>  Save As ProTracker is switched on/off
XPKPackMethod <Mode>  Changes the XPK-CrunchMode (4 Chars !!)

NoFilerequester  Set other filerequester mode
Req.Library      Set other filerequester mode
ReqTools.Library Set other filerequester mode
ASL.Library      Set other filerequester mode

```

```
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> |
| hef | Helpfile found            | <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> |
| upb | UPrg-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   |
| chd | Size of Checkdifference   | Number   |
| epm | Entries per Menu          | 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   |

## 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> |
| msm | Module saveable            | <yes no> |
| mii | ModuleInfo possible        | <yes no> |
| sai | SampleInfo possible        | <yes no> |
| msa | 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 userprogram 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

```

## 1.58 anpassung an directory opus

### 7.3 Arexx-Adaption to DOPUS 4

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 "rex:" 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 sound-module". Click on "New Entry" and enter "rx rexx:EP\_load\_one\_Module.dopus {f}", make it run asynchronously (as described above) and click on "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.

## 1.59 thanksagungen

### 8. Acknowledgements

-----

First we want to thank DELIRIUM without whom our program probably never was released. That's because of the fact that we adapted the really good external player system for our purposes. Special thanks to Frank Riffel helped us out with a lot of modules for test purposes.

Second we thank all members of our group for support with graphics (M.U.D.U., Ramon), programming of our Eagleintros (Crazy Copper), informing about FAKE versions of the Eagleplayer on different boards (CLIFF), modules (Animal) and for new Ideas.

Third we wanna thank Turbo/Infect helping us beside the newest versions of his ripper we got with Tools/modules/sources and tips. Another thank for the Exotic Sourcecode where we took some size calculations from for our save function.

Another thank goes to all our registered users for supporting us and for bug-reports and new ideas. Please dunderstand that not everything can be build in

---

at the same time.

A special thank goes to Oliver Borrmann (our 1st registered User) for the many bugreports and modules and the InStereo 2.0 demo for which we didn't have a playroutine yet.

Much thanks to Manfred Jacobi and Michael Wesp for the lots of suggestions, bugreports and very nice pictures.

MC Noise we thank for the music of our Eagleintros (he's still in PRIME DESIGN, but how long he'll stay in this lame group anyway ?) DEFECT RULZ !

Another special thank to the authors of XPK whose package is the best one for packing and unpacking files ever seen (except XFH who's packing twice when the file was already packed before - LAME ERROR)

Another thank to Capitain Bifat (TEK) for his advises and the new surfaces, I'll try to make use of them as fast as I can.

Thanks to Thomas Schwarz for his CRM and the documentation of the Library.

Thomas Winischhofer we want to thank for his MusicMaker replayers he adapted himself and the critical suggestions to our player

Braintrace Design we wanna thank for the original Sonic-Arranger which helped us much developing the replayroutine.

Furthermore we wanna thank Michael Sinz for the Enforcer, one of the best debugging tools I've ever seen, using this one we've found a lot of illegal hits. (in connection with "mungwall")

M. Praschel danken wir dafür, daß wir den Player auf seinem A4000 testen konnten.

Thanks to our Modemtrader in Schwerin who's spreading our player to everywhere in the world.

A - a little bit less happy - regard to Stefan Stuntz not needing to answer to a very long letter with bugreports and suggestions although we're registered users of MFR. Anyway the proggy is quite nice. (Also when it doesn't work with the new regtools.library and Asl.library and shows the Filenamegadget when selecting a dir and and and.....)

A nice greeting to Ron Birk, who's coding on the wow Playsid, yet thanks for the playsid-Eagleplayer coming up soon. (use the Delitracker Version till then)

Thanks also to all BETA Testers not mentioned, Sorry !

and thanks to Mathias "MAT" Uhlig who translated the first 400 lines of this file into the english language and corrected the file!

Ok, that was it, much fun with our product and best wishes from the coding team

---

Eagleeye & Buggs of DEFECT

## 1.60 registration

### 2.5 Registration of the Eagleplayer

---

The Shareware-fee for the Eagleplayer is DM20 (US\$ 30 outwards of germany) which we expect as chash or Eurocheque. (other currencies or cheques we cannot change so don't accept them !!!)  
 You'll recieve the latest registered version of the Eagleplayer containing the keyfile and all Userprograms. 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.  
 ONLY PRIVATE ADDRESSES !! We won't write to PLK's !!

Please send the filled  
                           registration form  
                           to one of these  
 addresses:

Eagleeye

-----

Jan Blumenthal  
 Erich Mühsam Straße 33  
 18069 Rostock  
 Germany

Buggs

-----

Henryk Richter  
 or Stephan-Jantzen-Ring 47  
 18106 Rostock  
 Germany

Phone/FAX: +49 / 0381 / 83979

Phone: +49 / 0381 / 1208867

E-Mail: [dtc@informatik.uni-rostock.de](mailto:dtc@informatik.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. NO DISK, NO ANSWER !!! We just can't send away (nearly) thousands of Disks. Send us a Disk and you will get the latest Demo- or the according registered version back.

When you describe bugs please look first into the chapter  
                           Error description  
                           !

## 1.61 registrierkarte

---

```

!           Eagleplayer           !
!           Registration           !
!           Form                   !
  
```

---





A simple Error was removed which caused the Eagleplayer at A4000/40 machines to crash in Copyback Mode. (We removed this bug some time ago but it seems that there was a little trouble with the sourcecode, sorry.) We added the external Replayer "Aprosys". The next change was done in the Installationfile to install the Libraries correctly. (DON'T FORGET TO START THE POWERPACKER PATCHER BEFORE INSTALLING THE EAGLEPLAYER !!) Another Userprogram called "Space Scope" was added but it uses a lot of CPU-time on not accelerated machines. In the Noiseconverter Utility the loaders for Promizer 2.0 and Mexx-Packer were added. The Mexx-Packer is now an intern Replayer. The randomplay routine was rewritten. Some little bugs within the Show Dirnames Function, Voice control and some Error Statements were removed.

V1.50 The Eagleplayer was rebuilt in lots of intern routines. In Main- and Playerwindow shortcuts were added. The gadgets of the Playerwindow finally look better. The intern Sound/Protrackercheckroutines were changed one time more. The Songendroutine works better now. Some bugs in the Soundtrackerplayroutines were removed. (e.g. a Enforcer-Hit of the original Protrackerroutine) 13 internal replayers were added by building in a converter routine which converts the patterns back to Protracker which replays these modules with the original playroutine - we think the best way, original sound, less playroutine bugs and as less as possible memory wasted by the replays. The load Dir routine was rewritten and works better now. If the private dirmem is full it doesn't crash anymore. (size can now be set directly) It overtakes now as much entries as fit into the current size. A new feature is the possibility to select subdirectories and parent dirs so that the filerequester is not needed therefore. Using Appwindow it is allowed to use Icons from different directories or volumes. Adding an external replayer you may use the multiselect feature of the filerequesters now. Another very new feature is that the external replayers may be loaded when they're really needed, not earlier and may be ejected when the module is thrown out, saves a lot of memory. The Eagleplayer may be controlled by extern programs (Noiseconverter, Exotic Ripper) partially. Registered Users may use the "load before" feature which makes loading the next module suddently when a module was loaded. So it is possible to start this module directly after the last one without a break. Very useful for megamixes like "RAF Megamix II", "LSD-Total Kaos" or "Max Mix" (by Flash ??). Not the Hotkey function works better. The default Hotkey now is "Ctrl Alt E". Now the Style guide compatibility is better used in case of commodity handling. (Hide Command in Pulldown Menu, doesn't quit on Closewindow if not set in the preferences) New is the powerful Save-Mode. You may choose between Normal and Automaticsave. The modules may be packed now. You may choose Samplemode (NEW NEW: Powerpacker Samplemode © DEFECT with wow results) and Protected. You may set an option to automatically protect the module against deleting and writing. The XPK Packmode may be chosen directly. The full Crunchmania Support was added (using the CRM library) including loading and saving all formats. Whilst packing a status display shows how far the packing progress is. The actual Public Screen to open the Eagleplayer on may be entered directly in the main window.

Auto Password and Quit Eagle were added.

Now the Eagleplayer works correctly together with "RTPatch".

Eagleplayer will scroll when the Diskfont.library is not found.

Now we added a "warning menu". You may choose if e.g. a warning requester is to appear if a loaded module hasn't got the right size or is corrupted.

Almost all extern replayer were rebuild to the new Eagleplayer Standard and won't work with the Delitracker anymore. Music Maker module may be packed now. From this version on the english doc is available - as you see. An error in the load config routine was fixed occuring on machines with 32 bit Ram.

A new submenu was added: "Modules-Directory". There you may change the prefs for the Modules-Pulldown Menus. The OS2.0 notify function was added for better directory refreshing.

You may choose between 3 different filerequesters and a command line input. The Eagleplayer doesn't need any external library to start.

Userprograms may be quitted now while the Eagleplayer is running.

Now the number of Userprograms is limited up to 32. A list controlling and generating unit is now available as userprogram offering several additional features for CD-Player like programming.

Some Errors in the control port for the USerprograms were removed so that the HIFI-System won't crash when it is situated in the 32 Bit Ram.

V1.40 Now the Eagleplayer supports an own fastdir called ".EPDIR" this is a simple method for loading the directory within a very short time. Furthermore the function "SaveT" was build in making it possible to load packed modules with almost all soundsystems. The Eagleplayer has got an own hotkey function. Except the hotkey all other functions are only for registered users. The following replayers were added: LME, MTH, the Player V5.0, IFF-SMUS, IFF-8SVX, Music-Maker 4+8, Rob Hubbard and Soundcontrol. The self modifying replayers use for rebuilding their playroutines the intern routines "EPG\_ModuleChange" and "EPG\_Modulerestore". The routines "EPG\_Hexdez" and "EPG\_TypeText" were added as well. Another option added is "Quit Surface". Here you may quit the Eagleplayer by closing one of the Userprograms (surface). An error loading the config (APPICON) was removed. A bug of the volvoice routine has been removed.  
Release Date: September-20-1993

V1.29b A fake version not coming from us. This is a Powerpacked and then depacked Eagleplayer. Because modifying the code while unpacking the player crashes at any time it wants. Out of this all version numbers in the file were changed.

V1.11 The first version for registered users using the keyfile. Registered users may use now the save function, the help function and the other userprograms. 3 new replayers were added, Player V5.0A, Syntracker and Major Tom's Player. Within the replayers SIDMon II, Actioamics, Noisepacker, Prorunner V1.0/V2.0, Synth&Instereo errors were removed. The Prorunner x.x was renamed to his real name, Noiserunner. Almost all players changed from V1.11 on won't run with an older Eagleplayer or Delitracker. The reason is the new volvoice routine offering the possibility changine the volume of each voice separately. The Program-Mode got the functions Randomsong, AutoSubSong and LoadAlways. All new functions will only be available for registered

users.

The configuration of the userprograms was changed. A big bug of the intuition.library will be avoided. The bug happened when no User-program was loaded. Then the pulldownmenu "special->Userprograms" had got only the Item "Add userprg". Normally a second IText-Structure was given drawing a "~~~~~"-line under the item above. This one won't be drawn but trashing the memory. The number of the modules in the menu to the bottom depends now on the screen's height. The priorities of the Userprograms will be set correct now.

V1.10 Now is it possible loading and saving of an ASCII-configuration file.

An Arexx-port was build with a very complex instruction set. Example scripts for installing using directory opus are supplied.

The Eagleplayer is a commodity now. A pubscreen support is also build in. Additionally the Eagleplayer is able to load fonts but only 8\*8 and non proportional ones. The Iconify mode was finished as well. Now "App Item" and "App Icon" are possible. An online Help is available for registered users in the case that the file "Eagleplayer.help" is found. The Tags "EP\_Date" and "EP\_Check3" were implemented. Now unpacked FTM and Oktalyzer 8 voices modules can be loaded into fastmemory. The EPG\_Globals were expanded (EPG\_File-request, EPG\_TextRequest, EPG\_FindTag, some Librarybases and important data areas.

Attention: The Eagleplayer 1.0 doesn't test the EP\_Playerversion !!! The Eagleplayer-Status-Structure was removed because all datas are now defined within the EPG\_Globals. A Helpfile structure and a gadget structure for the textrequester are now defined in the Include-file. The Userprogramport was expanded and the message class "Command" implemented- It's now possible creating new surfaces and run the original Eagleplayer in background. An Example for this new possibilities is the Userprogram "NoiseEagle", a simple Noiseplayer like gag.

In the Replayers Actionamics, Sidmon II, Player 4.0a/4.0b/4.1a, Propacker, Oktalyzer bugs were removed. The routine to find the module author was much bettered and may be used by extern replayers. A stupid bug in the SetSpeed Routine was removed (false parameter handed over. The output of the supported functions in the player window now works fine. Some time ago the Userprograms chashed while leaving the player, this shouldn't happen anymore. Some bugs of the Loadmodule routine and Change Voice Routine were removed. The "TAG\_SKIP" is supported now. The hippel replay plays modules where the init bit isn't set. A bug in the Jamcracker player was removed. The promizer 4.0 and Chaos Engine players were added. The check-routines for intern Soundtrackers were updated.

The error-messages in the include file were renamed for compatibility purposes. A resource symbol file to disassemble Eagleplayers is supplied.

Release Date: July-10-93 on the Sun'n'Fun Party Hellental

V1.01 Some Bugs in the Load routine and the OpenWindow function were removed. The globals got the function "DTG\_WaitaudioDMA" so that the Eagleplayer got Delitracker 1.37 compatible. The programming guides for coding replayers were expanded. The Eagleplayer was enforcer-tested for the first time and - we found some bugs we did not know until this time.

The intern Startrekker checkroutines have been changed and the



Startrekker player shouldn't crash anymore. To the Eagleplayer package now belong additionally the Promizer 0.1 and Promizer 2.0. Grouleff uses finally the EP-standard. Almost all replayers were rewritten and Enforcer hits (ActionAmics, SidmonII, P4xx-Tracker) were removed. Furthermore almost all replayers find Songend !  
Release Date: May-15-93

V1.0 First freely distributable demo version with a lot of bugs and disabled functions. Furthermore the docs and coding instructions (especially Userprogram-support) were not finished. The include file wasn't supported completely.  
Release Date: April-24-93 on the DATALIVE in Dessau

## 1.64 externe player

Something about the until now  
for the Eagleplayer adapted playroutines

© 1993 DEFECT

All players we adapted have been tested carefully. It may anyway happen that some modules, especially those with playroutine within them don't work or some functions, the Eagleplayer normally provides, could be disabled while other modules of the same soundsystem work fine. On these Players we included a special notice. It is also possible that some modules will be recognized, but do not play or even crash down. Such critical players, where the play-routine changes from module to module are e.g. Hippel, Whittaker and also Sonic Arranger.

When you describe Bugs of the external replayers, please look first into the chapter

Bug reports

! Look also into the

chapter mentioning

bugs of the external replayers

we couldn't fix until now

because the creators have hidden them too good.

When you send us the bug report we'll try to fix it and send you the corrected playroutine back.

If you have got any player/soundsystem not supported by the Eagleplayer, please submit it us (ReplaySourcecode + Modules if possible), we'll try to adapt it.

In this Version the Eagleplayer has got 18 intern replayers. So if it neither finds the directory "Eagleplayers" nor "Deliplayers" (nor the Playerdirectory you specify in the configuration file) it can only play Protracker/Startrekker/OldSoundtracker/Audio-Sculpture. Prorunner, Propacker, Wantonpacker, Unic Tracker, GMC, Digital Illusions, P4XX and a special Custom-Module Format.

If not extra mentioned, Volume, Balance and Songend are usually supported and the playroutine uses the Eagleplayerinternal CIA-Timer.

If you miss a certain playroutine you can also use the external playroutines of the Delitracker V1.3...V1.37...??? if available.

```
*****
***** Informationen zu den Externen Playern *****
*****
```

ActionAmics: Ripped Playroutine, used e.g. in the Game "Dynablaste".  
Tempo-Bug (Thanks Turbo/IFT) and Enforcer-Hits removed.  
Action Amics supports Subsongs, Patternswitching, Volume,  
Balance, Voices and Userprograms.  
Songend will be recognized.  
Action Amics is 99% related to SIDMon II !

---

ActivisionPro Soundsystem mostly used in older games. The playroutine is terrible (sometimes even 2 after another), uses wait-dbf-loops and - how could it be else - is placed within the module.  
Our Player supports Analyzer, Volume, Balance, mostly also Songend (depends on Patternswitching)

Problems with turboboards expected !

---

Audio Sculpture Commercial Successor of the Startrekker with extended AM-Abilities. The files differ in the recog string of the NT-File. The replayer is already "Eagleplyer-built in" and is also able to load songs !!  
Supports Songend, Patternswitching, Volume, Balance, Voices, Userprogrammcontrol, Songend and so on.  
It is able to save modules in the registered version.

---

ChipTracker look at "KRIS"

---

Deltamusic1.0: We took the playroutine of the Delitracker, but extended it with much features like Songend, Volume, Balance, Voices, Analyzer.  
I wonder that Delta 2.0 uses an intern replay, not like here, Playroutine and Module separated.

---

Deltamusic2.0: PlayRoutine within the Module. Soundsystem runs also at Kick2.0 and is even PD !

But: for synthetical Songs the Modules are too long.

Deltamusic 2.0 supports Analyzer, Volume, Voices ....

Songend will be found.

Note: Don't save Delta-Modules too short because the player could crash in this case.

---

Digital Mugician:

Soundsystem with cool effects, runs under 2.0 but is a total hardwarehack (supports only DF0: !!!).

Playroutine taken from Delirium but, I had anyway to do almost everything by myself. The Playroutine has been enshortened 500 Bytes and moved into fastmem. Die number of subsongs can be calculated and Songend will be found. Sizecalc of the module is included, patternswitching is also possible. With a registered Eagleplayer you can also save these Modules.

---

DSS Digital Sound Studio, a Soundtrackerclone as additional feature to the GVP soundsampler. When somebody not absolutely needs a sampler, he should go for the Protracker (TM) the DSS-Tracker offers almost no SpecialFX. Anyway, we did all the Soundtracker-typical adaptations, the playroutine we took from delirium but these guys were even too lazy to put a volume control into this piece and that on a soundtracker, embarrassing.

---

EMS A french product, I don't know anything about this piece, the playroutine I got (like also my only module) from Delirium. But anyway there was a lot to do in the Deliplayer, because there were neither Volumecontrol nor Songend and so on and the playroutine is documented (a little bit strange to read french, English rulz !)

---

FC1.3/1.4 Future Composer V1.0 to V1.4 Players. Playroutine by Super Sero of Superions. Often used in Intros because of the short Modules. Volume, Balance, Songend, Voicecontol. The internal Eagleplayer-CIA-interrupt is used. Does anybody know how to convert Hippel-Modules into FC ??

---

Fred: And again a player with playroutine in the module ...

---

(the people doesn't learn that is a bit bullshit)  
Our Fred-Player supports Volume, Balance ...  
A SubSongRange was included.

---

FTM Face the Music by Maxon,  
the 8-Voices Musicprogram with the best sound of all available  
for Standard-Amigas, but the worst coder support, no play-  
routine given and an absolutely awful protected CLI/WB-Player.  
It was a Hard Work to get this player to run, not this  
player supports all needed features, the internal loader  
was removed, (modules now may be packed with the supported  
crunchers) the Hack to \$70 has been removed and the audio-  
channels will be allocated correctly.  
In the registered version it is possible to save a module.

---

GMOD MultiPlayer-CustomPlayer  
adapted by Delirium. No further Informations available.  
No Analyzercontrol. Eventually Volume  
(depends on Module).

---

GMC I don't know what to say to this Program, nobody uses it and  
the playroutine was unoptimezed in a terrible way. I took  
the deliplayer (because of my lazyness) and extended it  
(e.g. enshortened the reloc-table). Supports everything what  
the other Soundtrackerroutines also do (yes, it is a  
Soundtrackercclone, at least the module-format)  
In the registered version it is possible to save a module.

---

Grouleff A Soundsystem (lousy one) with Playroutine within the really  
short modules. All usual adaptations, Songend also built in.

---

Hippel Arrg, Hippel.. Playroutine within the modules, and every module  
is different from the other.  
The Hippel-Player looks at now for 27 Sub-Routines to be  
changed that Hippel gets Analyzer-able, yes, it's true, Hippel  
supports Analyzer, Volume, Balance and also Songend.  
I don't believe all Subroutines to be found. There is anywhere  
100% a Hippel-Module doesn't working. Hippel is the "father"  
of the Future-Composer !!!  
Hippel is an already very old Soundsystem.  
Note: Problems on A 4000 !!

---

---

Hippel-Coso: Hippel-Coso-Modules are Hippel-Modules without playroutine and the Module starts with "COSO". The Player tries to load an extra Sample-File. We are using an extended Delirium-Player. Playroutine is now in Fastmem. Songend included.

---

Icetracker A Soundtracker 2.6 Clone, which almost no special Commands supports. (even less than the good old Noisetracker) Therefore runs this piece sometimes under OS 2.0. The Eagleplayer supports all usual Soundtracker-facilities. What luck, I found a Replay-Source on an Internet-Pack, but oh what interesting, the playroutine was partially disassembled. Very Strange Piece... Next funny feature. These guys neither 2nd Interrupt nor DMA/Loop-Wait. Wow, how this all coders do ask now. Easy: You use 3 Music-Routines called "IC\_MusicA", "IC\_MusicB" and "IC\_MusicC" and give the problem to the user, cool eh? In the registered version it is possible to save a module.

---

InStereo/Synth: Two related Soundsystems, InStereo was the Successor of Synth. You may play InStereo-Modules also with the Synth-Player. To avoid complications we took anyway two different players. Some Bugs (no, not BUGGS) were removed. Moduleinfo is also included (Songname, Authorname, Sizecalc, Subsongs). The replays find Songend. Both routines were moved to fastmem.

---

IFF-SMUS Playroutine taken from Delirium. SMUS-Files may be packed now using the "Save T:" function. Now you may load a module whilst another one is playing. SMUS does support the Analyzers now.

---

Jamcracker Not often seen Soundformat with clearly arranged replay, synthsamples preferred used. Runs problemless under 2.0, at least the Version of XGAG known to me.

---

Kris Real Name ChipTracker (thanks Venom), we also used the name "4MAT-Tracker" for it. It is a Noisetrackerformat, a little bit packed, supports all abilities of our trackers. I think matthew simmonds to be the creator, it not, let me

---

know.

In the registered version it is possible to save a module.

---

LME ripped Playroutine from a Customsong by Marley/Infect.  
We build up a replayer from it in a hurry.

---

Major Tom's Player

Playroutine overtaken from marley/Infect and adapted to the Eagleplayer. We've got only 2 Modules. Who's got the original Program and/or Modules ?? The replayer supports now Songend, Patternswitch, Moduleinfo & Analyzer.

---

Mark-II A Soundsystem by Cachet. MII has a strange sound and is when only one Userprogramm is active (On A4000 ofcourse all :) funny to see. Old-MarkII will be also played with this player. Songend, Volume, Balance ....

---

Martin Walker removed, because this format works under the name "Activision Pro" in the Eagleplayer list !!

---

MCMD Ripped Playroutine (1 TestModule), looks like Hippel-Playroutines. Also the Module looks so. Anyway, we included this piece.

---

MED Program by Teijo Kinnunen, probably created out of the idea of a soundtracker under DOS, is able to load and save soundtrackerfiles, into MED converted Soundtrackermodules are shorter.  
This player is for MED and OctaMED songs with 4 Voices. The player I took from the Delitracker, to save a little bit time.  
MED has a terribly long playroutine, who uses an own CIA-Interrupt. Also the Audio-Channels will be allocated by this player itself. Volume, Balance, Voices and Analyzer are ofcourse supported.

---

Mexx-Packer From a Mexx-Intro ripped playroutine. This one is totally

---

in a mess and self-modifying like a ....--censored--....  
This coder has to be forbidden, bullshit.  
Second Interrupt removed, self modifying crab also.  
Mexx-Packer supports Songend, Patternswitching, Volume,  
Balance .....

---

In the registered version it is possible to save a module.

From Version 1.51 on this replayer is Eagleplayer-intern.  
It is possible to save the modules as Protracker

---

ModuleProtector:Modulformat, used by LSD in their productions, we only support  
the one mode used by LSD and in the Voice Attack II MusicDemo.

---

MON MON means Maniacs-of-Noise Player. MON was used e.g. in Unreal  
(parly at fixed Adresses, normally not the case). M.O.N. has  
the Playroutine within the module.  
It will be modified at startup,so that MON supports Analyzer,  
Volume, Balance und Voices-Control. Songende will be  
recognized.  
Partly problems occured at A 1200.

---

MusicAssembler: MusicAssembler has also an internal playroutine in the module.  
I own a MA module with only 3 KBytes and that with play-  
routine !! Amazing, don't you think ?  
Music-Assembler is Analyzer-able ... .  
Songend will partly be recognized.

---

Music-Maker 4/8 The Playroutine we've got directly from the author. Thanks  
Thomas ! Sounds very good (better than every other 8-voices  
player, even much better than FTM !) The problem is that  
the songfile must be named "\*.sdata" and all files must exist  
in one directory.  
Our replayroutine is MusicMaker 2.4 compatible yet !

---

Noisepacker2 Program to enshort Noisetrackerfiles, more efficient than  
Prorunner2.0, but only Noisetrackersupport.  
Comes from Twins of Phenomena, is used often in Quartz,  
Andromeda and also Anarchy - Productions . The Player can  
calculate the real Modulesize and also the number of  
patterns and so on (useful for guys ripping by hand)  
The internal Interrupt is used, the routine for recognizing  
the module is written in a way that the module may be also  
too short or too long, the Eagleplayer is able to tell the

---

right values (ABOUT MODULE- Funktion)

Es wird der interne Interrupt genutzt, die Erkennungsroutine ist so geschrieben worden, daß das Modul ruhig auch zu lang oder zu kurz sein kann, die richtige Länge erscheint bei About Module im Scrollwindow und man sieht genau, wie lang das Modul sein muß.

In the registered version it is possible to save a module.

---

Noisepacker3 Remake of NP2, with NP2, own Window, much better packing rate, even better than Propacker 2.1. Rest see Noisepacker2

---

NoiseRunner This crypter/player comes from Chaos of Sanity. The "M.K." sign is still within the module what makes every standard protracker replayer think it to be Protracker. This piece is intern and the modules will be converted back to Protracker and played with the Protrackerreplay.

---

NoisetrackerCompressed

The United Forces Noisetracker in the Version 1.3 D, has the possibility to save Modules in a packed way. For people, who still work with noisetracker surely a good alternative to noisepacker. Eagleplayeradaptions done in the usual way, pretty packing rate.

Modules with playroutine within them not supported yet.

In the registered version it is possible to save a module.

---

MTH We got the playroutine from Turbo and adapted it. Supports Subsongs and most of the functions a real Eagleplayer must have got.

---

OctaMED Player for 5-8-Voices Octamedsongs. Was taken over from Delitracker and extended in the usual way. The Bug which occurred in the Delitrackerversion 1.3 (Fish 748) was removed. Rest, look at MED.

If anybody owns a playroutine for 8-Voices in HQ-Mode and this is freely distributable, please send it, because on turboboards is then a quality possible, comparable to FTM !!

---

---



Oktalyzer First 8-Voices-Musicprogram I know and in this way the playroutine looks like (I even heard that the creator himself doesn't see through nowadays)  
I included a CIA-Timer, so that it runs correctly under the NTSC, Productivity and VGA-Modes.  
This Routine supports now also Songend, Moduleinfo and Patternswitching. The little Problems with Monoscope, Stereoscope and Quadrascope of the 1.0-Eagleplayerversion have been fixed.

---

#### Old Soundtracker

Playroutine for old Soundtracker (16 Samples). We used the D.O.C. - Soundtracker 2.0 Playroutine, which is at least partially compatible to most of our old ST-Mods, some old trackers coded, adapted, stolen, cloned by a lot of confuzed guyz used different commands, almost impossible to check and who uses still the old onez ??  
The player is already intern in the eagleplayer and supports also Songloading (including packed songs). You can furthermore change the settings of the default-instruments-path. The Samples - if you load songs - may be packed with the supported crunchers (e.g. Powerpacker, File Imploder, CrM, LH, XPK, Stc, Pack)  
In the registered version it is possible to save a module.

---

#### Player-P40A/P40B/P41A/P50A

A very strage one, a Protrackerplayer, but more a packer, not to say, a lot of Beta-Versions.  
But now a good feature: It's the best packer.  
The Bad features: Module lies in Versions until 4.1 at fixed addresses (therefore we included a dereloc-routine, he he :) )  
Playroutines were like always like the other ones, endless hardwarehacking, but fast, very fast, not to say ultrafast. Second Interrupt removed, not a little slower, but system-friendly and CIA-Compatible. It was really a mess to play all modules we got with only one replay (we had to rip them) and can still not face, how it is possible to write replays incompatible to themselves, e.g. we have some modules not able to play even with the P40B mainprogram !! A lot of Enforcer-Hits removed and the players should run now a lot safer.  
Partly Moduleinfo and Songend supported.  
In the registered version it is possible to save a module.

ATTENTION: From Eagleplayer V1.50 on P40A/P40B and P41A are intern and the modules are played with the standard Protracker replay for playing the modules in the original way. It is possible to save the modules as Protracker

---

---

Promizer0.1: The Protracker Optimizer by Frank Hülsmann (Tech). First version of the Promizer. PlayRoutine not within the module and no SubSongs.

Promizer1.x: Well, of the Promizer 1.x there are so much versions now on market, that it is a veryyyy big problem to get them all into one replay, one day we adapted something new for a single module and next day there we rip another one even worse.

This Promizer-Version supports subsongs, a really nice feature.

Promizer2.0 Next Version of Promizer. No longer Subsong-Support.

Why ??????

All Promizer-replayz support PatternSwitching, Songend, Volume, Balance, Voices & Userprogramms.

Promizer4.0 Wow, the latest Promizer has its playroutine separated from the module and with a short recog-string, but the playroutine comes all in all from Prorunner 2.0 and the module-format, well I saw it before at Propacker 2.1, it's almost the same !

Note: Promizer-Module are mostly to be found TECH-Demos !!!

(and now ofcourse also at Masque)

99% rippable !!!

---

Propacker V1.0a

Was written by Azatoth of Phenomena, packs rather poor, playroutine was self-modifying, endless hardwarehacking, some commands were missing and the top of the hill is - this guy wants to see money for that silly piece.

My tip to Azatoth: STAY CODING DEMOZ, YOU ARE NOT TALENTED FOR DOS-PROGGIS.

For the Eagleplayer the second Interrupt was removed, Module-info built in, which is able to calculate the correct size and packing rate and I implemented the Tempo-function, so that also modules like "Comdom Corruption" (by Spaceballs) and "Blow ya Nose Now" (E.O.K.D.P.-Anarchy) work with the correct speed.

-----> Because of the fact that the patterns are situated behind the samples (what an idea) the checkroutine doesn't tolerate modules being too short because of the very possible danger of a system-crash. So when you rip this modules out by hand, better save some bytes more, the player tells you the correct size (the registered version ofcause saves them !)

I added the possibility of a config which means that you are able to decide whether there occurs a requester telling you that the module is too short and also how much

---

(approximately, coz this is a packer) longer the module must be.

The playroutine uses an own CIA-Timer.

Attention: The Packer is able to remove double used or unused patterns.

So it is possible that in the "About Module"-Mode a different number of Patterns occurs than the original file consisted of (so the original size is also different)

Attention II:

This piece exists also under the names "Strangeplayer" and "Hannitracker", both with a catastrophically checkroutine not taking care of the problems mentioned above, not even Volume supported.....

(and both "Versions" are within the Delitracker 1.37 Package on Internet, nice eh ? I suppose the guys of Delirium don't see the sun anymore or what ?? :)--== )

From Version 1.50 on this replayer is Eagleplayer-intern.

It is possible to save the modules as Protracker

---

Propacker 2.1 Alternative to Noisepacker3, packs only a little bit worse but supports all Protrackercommands.

The checkroutine I took from Turbo of Infect, but I removed a lot of Errors. :)--==

It is easily to recognize, not like the "Vxx" called Propackers.

In the registered version it is possible to save a module.

From Version 1.50 on this replayer is Eagleplayer-intern.

It is possible to save the modules as Protracker

---

Propacker 3.0 read above, little changes in the module (different

Offsetformat), not with PP2.1 playable, therefore own

Player. I found this version only in Parasite-Music-

Disks. (I ripped also the playroutine there)

In the registered version it is possible to save a module.

From Version 1.50 on this replayer is Eagleplayer-intern.

It is possible to save the modules as Protracker

---

Propacker 1.0 (SB)

see at Moduleprotector , only the difference is that the patterns are splitted into 4 parts and the routine comes from the creator of Propacker 2.1/3.0

---

---

ProrunnerV1.0 A Playroutine written by Cosmos of Sanity, which converts Protrackermodules to gain speed. Well, when I ripped my first Prorunnermodules I didn't own the program itself, so I had to rip the replay, but there were some problems and I decided to build an "Online Decoder" into my Protrackerreplay. It works fine and all abilities of the Protrackerplayer you could also use with this one.

From Version 1.50 on this replayer is Eagleplayer-intern.  
It is possible to save the modules as Protracker

---

ProrunnerV2.0 This piece comes also from Cosmos/Sanity and has the following new features:

1. easy to recognize, easy checkroutine
2. fast replay
3. comfortable Eagleplayerreplay :-)
4. packs almost as good as Noisepacker 2
5. has the Eagleplayer the Protracker Tempo-Function

But there are - as always - some little problems with Prorunner 2.0, for instance that in times of req and req-tools there isn't any comfort for converting the module at all. Another big problem is that it has a bug in the packing routine I had this problem with "Condom Corruption" by Spaceballs.

From Version 1.50 on this replayer is Eagleplayer-intern.  
It is possible to save the modules as Protracker

---

Protracker What more to say, THE standard in the field of sound on the one and only AMIGA, often covered, stolen, overtaken (e.g. the fabulous Archimedes), even to the LamETARI and the worst shitty PC.

We use the Protracker 1.1 Playroutine. The new CRB Protracker 3.0 still uses the old commands, so we left the old one in our player.

The Protracker is already intern and doesn't need to be loaded extra. It supports Songloading !! and also packed songs.

A path for the Samples may be set. (look into the Players-Window) The Samples may be packed with the supported crunchers. (Powerpacker, File Imploder, CrM, LH, XPK, Stc, Pack)

We do not differ between Noisetracker, Soundtracker and Protrackermodules because all Modules of us (about 1800) work fine and the routine of the Delitracker, the creator Frank Riffel called it "intelligent", has too much problems - seems rather to be a gag to increase the number of supported replays.

It is possible to save the module even when it was loaded as a song.

---

---

Pumatracker1.0 Remake of the legendary Futurecomposer, but not 2.0 compatible, doesn't load the futurecomposermodules, has a nerving Intro direct before the main-program, but I must say, it sounds rather good (for those who are able to handle the Editor) Done in a hurry out of the Puma-Delireplay by Turbo/Infect. Songend not supported yet.

---

Rob Hubbard Soundsystem with Subsongs. Caution: No SubSongRange was built in, on clicking to next Subsong there is the Danger to crash !! Analyzersupport.

---

Ron Klaren Yes, it's true, the Eagleplayer supports as the first program of it's class (or is it the first of all ??) the wonderful Ron Klaren Modules including Volume, Voices, Balance, Analyzer, Songend. Out of this our replay removes some Hardwarehacks.

To play these Modules is something needed and coding experiences are nessesary because these modules run only on fixed addresses and a special format needs to be used. Our suggestion is to disassemble the module completely so that ALL fixed jumps and pointers are changed into Labels (don't forget the ones in the Data-part) Then you load it into the Assembler, include the "Eagleplayer.i"-file and set the macro "KLarenheader" to the start of the module. The module has to be only one section and must be loaded into the chipmem ! Last you save the module as object-file. (Seka & ASM-One : WO for Write Object). Ron Klaren Modules may not be packed !!!

A disassembled module you find in the archive "Examples.run". Have fun and turn up the volume.

Example:

```
section 0,Code_C
Klarenheader end-start ;this number is nessesary
    ;and submits the size of
    ;the Objectfile ! Important !
<put here the disassembled module>
End
```

!!!! Attention !!! You may not change anything in the module, the Eagleplayer does it itself, all original hardwarehacks have to stay intact !!

---

## Remark:

The Ron Klaren Modules start with some fixed-Address Jumps followed by some (sometimes only one) move #xx,d0 bra xx - Instructions also not to be deleted, they are needed for subsong calculations.

---

SIDMon1: Soundsystem written in 1988, by R.v. Vliet, runs only under Kick 1.3, total hardwarehack, playroutine within the module. Sidmon supports Analyzer..., Songend and Nextpattern. (Hard Work MAN !)  
If the module can't be changed it runs in normal way like with every other player. The impossible functions (Volume...) will be disabled.

---

SIDMon2.0 Sidmon II, now MIDI-abilities, but runs only under 1.3, the usual hardwarehacking, a lot of time necessary to come along with this piece, but all in all it is possible to compose very good synthetical songs. The modules are at least without playroutine so that the player was really easy to adapt. We support Patternswitching, Songend, Volume, Balance, Analyzer ...  
What more ??  
An Error discovered using the Enforcer was removed.

---

Sonic-Arranger This player plays both, the Sonic-Arranger songs (unpacked) and Sonic-Arranger modules (with replay). To play Modules with intern replay we use the Song-replay, it works better but compatibility-problems could happen.  
The player supports Patternswitching, Subsongs, Volume, Balance, Analyzer and Voices.  
We took the Delirium-Player and bettered it (as always).  
The replay for SA-Songs now is situated in fast-ram.  
We suppose the Sonic-Arranger as the successor of InStereo/Synth.  
The Sonic-Arranger is a commercial musicprogram by Brain-trace Design. At this time we also want to thank for the demo-version of the Sonic-Arranger.  
It was very hard to find the songend, but it works now.

---

SoundControl We got the original playroutine from Turbo/IFT. All typical adaptations for the Eagleplayer were done.

---

SoundFX1.3 I think this program was sold as a commercial program,

---

intresting when I take a look at the "Playvoice"-routine which was obviously stolen from Soundtracker. So we have done everything what the other soundtrackerplayers also support, Patternswitching and Moduleinfo inclusive. In the registered version it is possible to save the module.

---

SoundFX2.0 New version of the soundsystem above, supports now 31 samples ( !!!!! ), other news I could not found, perhaps less bugs. In the registered version it is possible to save the module.

---

Soundtracker2.6 Successor of the legendary D.O.C. Soundtracker, 31 Samples, splitted Patterns (every Soundchannel with its own Pattern), but as less Special FX as all old Soundtrackers. In the registered version it is possible to save the module.

---

SoundMaster PlayRoutine within the module. Should support everything what is usual for these ones, we would need a good testmodule, we did only our tests , bababa baba ba ba bababa.

---

SoundMon A soundsystem by brian posta, very old one, but runs under 2.0 and works partially (if you assign "ST-00:" correctly and there exists a "modules"-directory) together with hard-disks. The modules sound good anyway and are usually very short so that they are often used in intros. Our playroutine is now in fastmem. Songend and Moduleinfo including sizecalc are also built in. In the registered version it is possible to save the module.

---

Syntracker: A nice looking playroutine. But the creator doesn't seem to hear from PC-relative coding. I could cut down to Playroutine itself from 4100 to 3400 Bytes. I've only got one Testmodule. The Program itself looks like a 4 voices Octalyzer and countains some bugs. The playroutine destroys the memory.

---

Startrekker Here we take the 4-voices AM-playroutine of Audio Sculpture. The Startrekker was written by Exolon of Fairlight, also known as Björn Wesen. As a special extra the ".NT" file is not needed if the module doesn't countain AM-Sounds. In this case that the ".NT" file could not be found you won't hear sound on

---

trying to use the synthsamples. The Startrekker supports all abilities of all other trackers including Songend and Module-info. The Startrekker is already intern within the Eagleplayer and supports also Songloading (ofcourse the Song and the Samples may be packed using Powerpacker, File Imploder, CrM, LH, XPK, Stc, Pack)  
In the registered version it is possible to save the module.

---

TFMX A soundsystem by Chris Hülsbeck

Here implemented in 3 variants:

1. TFMX V1.5
  2. TFMX Pro
  3. TFMX 7 Voices (idea by J.Hippel, Code J.Hippel ? & C.H.)
- TFMX is the best of all professional musiceditors and sounds in the hand of a profi wonderful. Very intresting is it to watch the samples in the Scopes (Monoscope, Stereoscope and Quadrascope) how there the surround-effects are simulated, wow. Well, we took the Delitracker routines over because of the fact that the guys of Delirium have more experiences with TFMX and direct contacts to Chris Hülsbeck. All TFMX players now have Volume, Balance, Voicecontrol ..  
Because of the fact that TFMX pro and TFMX 7V use 5 !!! interrupts they may trouble unter Productivity, NTSC, A2024 playing too fast. The TFMX 7V player I enshortened to about 1/3 of the original size.  
Restart and Songend will be recognized.

Attention: The TFMX 7V player doesn't play correctly on A1200 and A4000, it troubles because of it's DMA and Interrupt-handling, when I get an A4000 (I hope around christmas) I will try to fix it.

A TFMX with playroutine within Modul, like at Powerplayer ( I can't believe, how such a piece could be named "POWER...") claimed doesn't exist !! (confirmed by Frank Riffel of DELIRIUM)

---

TME The Musical Enlightenment, a soundprogram I only have got one module for so that a real test wasn't possible. The playroutine I took from Delirium but as always, almost "naked", only Subsongs were supported and TME only acesses at one, repeating at one point the Audio-Hardware ...  
Now the TME is one of the few players where no gadget in the mainwindow is diasbled. It is furthermore able to calculate the correct size and to save it (registered version).

---

Tronic Shit, I can't remember where I have seen this playroutine

---



before. I'm 100% sure that something very similar I worked on some time ago. Anyway, it is with a high probability NOT A TRACKER ! The playroutine was ripped by Marley/IFT and brought out as a deliplayer. I build additionally Volume, Balance, Voices, Analyzersupport and Songend in. A bug of the playroutine was removed.

---

Unic-Tracker This piece is a soundcrypter by Laxity of Kefrens. (in a rather easy rippable way)

The original playroutine was - like all those coded by hardwarehackers - a "something" with all types of shit normally avoided. (never heard of compatibility attempts, eh ?) Well after a lot of hard work (and even more swearwords) we present a Unic-Tracker without self-modifying code and a single CIA-Interrupt so that the Unic works under NTSC. Well, the playroutine looks like the attempt to change a noisetracker one into a protracker compatible. I added the functions "Sampleoffset" and "Tempo" so that most Protrackermodules of mine work correctly. (own converter used) Advantage of Unic-Tracker: Patterns ¼ shorter. In the registered version it is possible to save the module.

---

Unic-Tracker 2 In the "Desert Dream" Demo used "Unic"-Format without recog string and modulename but therefore even worse playroutine. Because of being lazy I rebuild my old UNIC-replay. P.S. the modules you find in the demo at address \$1532, End-address at \$152e .

---

VSS Voodoo Supreme Synthesizer, was probably direct by the author adapted to the Delitracker, got a Voices-Routine, CIAB-Timer, Analyzersupport and Speedcontrol for the Eagleplayer. Well, have a look at the Demosong "Voo4", the Samples look very funny, cool eh ! Problem : at the A4000 no sound at all.

---

Whittaker The first Whittaker player with Balance, Volume and Voices to be changed. Who knows the Whittaker-modules should know that almost every WH-Module has a playroutine differing from the other. This playroutine is rebuild at every "Init Player" call, that means we look for a special routine we change then into ours. Not working modules you should send immediately to us. A bug in the playroutine occuring at modules being too short will be stopped by our player !!! Nowadays the Whittaker works with all modules we've got but no guarantee given. Songend will partly be found.

---

---

### Wizard of Sound

I have never found a Wizard of Sound module (except the ones included at the main program) probably because nobody liked it ....

Well it took me a very long time to adapt this lousy piece. Only the - very short - playroutine had 2 kB, yes 2 kByte reloc. And only half of the registers are used, terrible. Furthermore the checkroutine to code was very hard, I find the module in general out only of its size, 9000 Bytes. (I check also the commands, but hm, terrible, I said it before) ofcause Songend, Analyzer, Lautstärke ....

the samples may be packed. (Powerpacker,File Imploder,Crunch-OMania,LH,XPK,Stc,Pack)

The Player supports a Config-Routine, is able to save its Instruments directory as a standard configuration. ("ENVARC:Eagleplayer/EP-Wos.Cfg")

---

### Thanks to:

- Turbo of Infect for some "exotic" Modules, playroutine and Converterprograms, and also for the quite cool Exotic Ripper which finds a lot of modules (attempts to ??)
- Frank Riffel & CO. for Sources of the Deliplayers ( so we hadn't to look for them), the many modules they sent us and the best available player-principle which is nowadays a standard yet we continued to use (and extended)
- the Authors of XPK, which wrote a powerful package around coding as freeware, stay this way guys !
- Venom for some hints and the P41A/P50 he sent us
- Captain Bifat of Electronic Knights for some help and deizgn recommendations (we'll try to use them soon)
- Thomas Winischhofer for his hints and adapted replayers

## 1.65 fehler in den playroutinen

### 4.2.5 Bugs within the players

---

Whilst testing with all external replayers we found several bugs. Now follows a list of Replayers probably doing some shit:

Action Amics (destroys memory)  
 ActivisionPro (not Turbo-compatible)  
 GMOD (some modules)  
 Hippel (works after recoding quite well)  
 Hippel7V (doesn't run on A500 as well as 50MHz 68030 Machines)  
 Old Whittaker  
 Player V5.0  
 SynTracker  
 alle TFMX (mostly the new songs of Turrican III and others)

---

Whittaker

Generally we must say that all Soundsystems with playroutine in the Module are dangerous as well as stone-old Soundsystems. Even the Soundtrackerclones countain mostly lots of bugs (that`s why we included intern "BACK TO PROTRACKER" routines)

---