

Eagleplayer

COLLABORATORS

	<i>TITLE :</i> Eagleplayer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 31, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Eagleplayer	1
1.1 Eagleplayer 1.53 Dokumentation	1
1.2 copyright und vertriebsbedingungen	4
1.3 distribution	5
1.4 was ist und kann der eagleplayer ?	7
1.5 entstehung des eagleplayers	9
1.6 was fehlt in dieser version noch ?	10
1.7 wann gibt es eine neue version des eagleplayers ?	10
1.8 update-service	11
1.9 fehlerbeschreibung	11
1.10 keyfile-benutzung	12
1.11 systemanforderungen	12
1.12 benötigte zusätzliche dateien	13
1.13 installation auf diskette,festplatte,...	13
1.14 start des eagleplayers von cli & shell	14
1.15 start von der workbench aus	15
1.16 das konfigurationsfile	15
1.17 das hauptfenster	15
1.18 speed	17
1.19 plwi	17
1.20 voi	17
1.21 prevmod	17
1.22 prevsub	17
1.23 prevpatt	18
1.24 restart	18
1.25 play	18
1.26 fast	18
1.27 nextpatt	18
1.28 nextsub	18
1.29 nextmod	18

1.30 stop	19
1.31 eject	19
1.32 disk	19
1.33 icon	19
1.34 filter	19
1.35 unnamed.1	20
1.36 about	20
1.37 close	20
1.38 vol	20
1.39 bal	20
1.40 steuerung über die pulldown-menüs	20
1.41 tastaturkürzel	28
1.42 die appwindow-funktion	29
1.43 pubscreens	30
1.44 font-unterstützung	30
1.45 interne depacker	30
1.46 der lha-extraktor	31
1.47 commodity-funktionen	32
1.48 öffnung und funktion des player-fensters	32
1.49 steuerung über die schalter und erklärung der funktionen	32
1.50 tastaturkürzel im playerwindow	33
1.51 interne player	33
1.52 tips zur konfig	36
1.53 einleitung zu den userprogrammen	38
1.54 steuerung der user-programme	38
1.55 zu den einzelnen bisher verfügbaren userprogrammen	39
1.56 keine programmierrichtlinien für userprogramme	42
1.57 einbindung des eagleplayers in andere programme	42
1.58 die playerbatch-datei	47
1.59 die userprogrammbatch-datei	47
1.60 fehlermeldungen	48
1.61 fehler des programms	49
1.62 problemecke	50
1.63 einleitung zu den externer playern	51
1.64 das externe player konzept	51
1.65 schematischer aufbau von externen playern	52
1.66 schematischer aufbau von custom modulen	52
1.67 anpassung von playern	52
1.68 playerheader	53

1.69 modulerkennung	53
1.70 interrupts	54
1.71 bedeutung der tags	54
1.72 eagleplayer support funktionen	58
1.73 tips zur anpassung	61
1.74 die besonderheiten und neuerungen des eagleplayers	62
1.75 die neuen tags	62
1.76 moduleinfo	64
1.77 analyzeransteuerung	65
1.78 die neuen eagleplayer-globals	68
1.79 einführung und beschreibung des arexx-portes	72
1.80 arexx-kommandoübersicht	73
1.81 anpassung an directory opus	81
1.82 dank sagungen	82
1.83 registration	83
1.84 registrierkarte	84
1.85 updatekarte	85
1.86 biographie	86
1.87 externe player	92
1.88 fehler in den playroutinen	111

Chapter 1

Eagleplayer

1.1 Eagleplayer 1.53 Dokumentation

Anleitung
zum
Eagleplayer

Version 1.53

Anleitung \$VER: V1.14 (16.11.94)

© 1993/94 Defect Software Productions
externes Playerkonzept © Delirium, erweitert von Defect

Inhaltsverzeichnis:

1. Copyright

1.1 Copyright und Vertriebsbedingungen

1.2 Distribution

2. Einleitung"

2.1 Was ist und kann der Eagleplayer ?

2.2 Entstehung des Eagleplayers

2.3 Was fehlt in dieser Version noch ?

2.4 Wann kommt der nächste Eagleplayer ?

2.5 Registration des Eagleplayers

2.6 Registrierkarte

2.7 Update-Service

2.8 Fehlerbeschreibung

2.9 Keyfile-Benutzung

3. Installation und Start

- 3.1 Systemanforderungen
 - 3.2 benötigte zusätzliche Dateien
 - 3.3 Installation auf Diskette, Festplatte, ...
 - 3.4 Start des Eagleplayers von CLI/SHELL
 - 3.5 Start von der Workbench aus
 - 3.6 Das Konfigurationsfile
 - 3.7 Tips zu der Konfiguration
 - 4. Benutzung des Programms
 - 4.1 Das Hauptfenster
 - 4.1.1 Steuerung des Programms über die Schalter (Gadgets)
 - 4.1.2 Steuerung über die Pulldown-Menüs
 - 4.1.3 Tastaturkürzel
 - 4.1.4 Die Appwindow-Funktion
 - 4.1.5 Pubscreens
 - 4.1.6 Font-Unterstützung
 - 4.1.7 Interne DePacker
 - 4.1.8 Der LHA-Extraktor
 - 4.1.9 Commodity-Funktionen
 - 4.2 Das Player-Fenster
 - 4.2.1 Öffnung und Funktion des Player-Fensters
 - 4.2.2 Steuerung über die Schalter und Erklärung der Funktionen
 - 4.2.3 Tastaturkürzel im Playerwindow
 - 4.2.4 Interne Player
 - 4.2.5 Externe Player
 - 4.2.6 Fehler in den Playern
 - 4.3 Die User-Programme
 - 4.3.1 Einleitung
 - 4.3.2 Steuerung der User-Programme
 - 4.3.3 Zu den einzelnen bisher verfügbaren Userprogrammen
 - 4.3.4 Keine Programmierrichtlinien für Userprogramme !
-

- 4.3.5 Einbindung des Eagleplayers in andere Programme
 - 4.4 Die Batch-Dateien
 - 4.4.1 Die PlayerBatch-Datei
 - 4.4.2 Die UserprogrammBatch-Datei
 - 5. Fehler
 - 5.1 Fehlermeldungen
 - 5.2 Fehler des Programms
 - 5.3 Problemecke
 - 6. Externe Player
 - 6.1 Einleitung zu den externen Playern
 - 6.1.1 Das externe Player Konzept
 - 6.1.2 Schematischer Aufbau von externen Playern
 - 6.1.3 Schematischer Aufbau von Custom Modulen
 - 6.2 Anpassung von Playern
 - 6.2.1 Playerheader
 - 6.2.2 Modulerkennung
 - 6.2.3 Interrupts
 - 6.3 Bedeutung der Tags
 - 6.4 Eagleplayer support Funktionen
 - 6.5 Tips zur Anpassung
 - 6.6 Die Besonderheiten und Neuerungen des Eagleplayers
 - 6.6.1 Die neuen Tags
 - 6.6.2 Moduleinfo
 - 6.6.3 Analyzeransteuerung
 - 6.6.4 Die neuen Eagleplayer-Globals
 - 7. Das Arexx-Interface
 - 7.1 Einführung und Beschreibung des Arexx-Portes
 - 7.2 Kommandoübersicht
 - 7.3 Arexx-Anpassung an DirOpus 4
 - 8. Biographie
 - 9. Danksagungen
-

9. Grüße und Danksagungen

1.2 copyright und vertriebsbedingungen

1.1 Copyright und Vertriebsbedingungen

~~~~~

Der Eagleplayer, die externen Programme, die Dokumentation, sämtliche Interfaces und auch die Entwicklerbeispiele sind ©1994 Henryk Richter und Jan Blumenthal. Wir behalten uns das Recht vor, juristische Schritte einzuleiten, besonders in dem Fall, daß Teile des Eagleplayerpaketes in anderen Paketen, Softwareprodukten usw. vertrieben werden.

Der Eagleplayer ist in der vorliegenden Version als

Shareware  
zu betrachten.

Dies heißt, daß wir bei regelmäßiger Benutzung eine Gebühr von 20 DM (bzw. 20US\$ außerhalb von Deutschland) erwarten, die in BAR oder als Eurocheque zu entrichten ist (andere Währungen oder Cheques können wir leider nicht akzeptieren). Als Gegenleistung erhalten Sie die neueste Version des Eagleplayers zurück, die einige, in der SharewareVersion gesperrte Funktionen (z.B. Save, EPDir, alle Userprgs) besitzt. Außerdem gibt's die verbesserte Version inclusive neuer Userprogramme alle gratis, so daß die Sharewaregebühr angesichts der Postpreise durchaus gerechtfertigt ist.

Schicken sie die ausgefüllte

Registrierkarte  
(Orderform) an:

Bugs            Eagleeye

-----

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

Telefon: 0381 / 1208867

Telefon/Fax: 0381 / 83979

E-Mail: [henryk.richter@stud.uni-rostock.de](mailto:henryk.richter@stud.uni-rostock.de)

Falls Sie irgendwelche Bemerkungen, Vorschläge, Bugreports, neue Soundformate, Drohbriefe, zu verschenkende Amigas, ... zu vermelden haben oder Probleme in der Benutzung haben, können Sie sich auch an die obenstehenden Adressen wenden. In diesem Falle: NO DISK, NO ANSWER !! Wir können nicht tausende (na ja fast) Disks verschicken. Wer uns zwei Disks schickt, bekommt auf Wunsch die neueste Demo- oder entsprechende registrierte Version zurück.

Der Eagleplayer darf unter folgenden Bedingungen auf Public-Domain Serien, Shareware-Serien und Netzwerken vertrieben werden:

- das Eagleplayerpaket muß komplett vorliegen, die zu diesem Paket gehörenden Dateien sind in  
`Distribution`  
aufgelistet

- Der Eagleplayer wird auf 2 Disketten ausgeliefert, wobei die Dateien auf der zweiten Disk komplett gepackt sind (mit Ausnahme der Module)
- weder das Programm, noch die Dokumentation dürfen verändert werden, eine Ausnahme stellt das Packen dar, solange sich der Originalzustand wieder 100% herstellen läßt (z.B.LHA,ZOO,ARC,ZIP,Powerpacker)
  - der Vertriebspreis für Eagleplayer selbst oder der Preis für Disketten, auf denen der Eagleplayer enthalten ist, darf nicht mehr als den Betrag von 5 DM oder den entsprechenden Betrag in einer anderen Währung ausmachen, gleiches gilt für die Übertragungsgebühren in Netzwerken
  - die Firma Mallander Software darf den Player nicht vertreiben !!!
  - der Eagleplayer darf nicht in Softwarepaketen vertrieben werden, mit Ausnahme von PD- oder Sharewareserien (z.B. Fish,Time,Bernd PD usw.), ohne daß eine SCHRIFTLICHE Genehmigung der Autoren vorliegt. Diese Genehmigung ist vor dem VERTREIBEN einzuholen !!!!!!!!!!! Wir haben absolut nichts gegen PD/Sharewareserien, möchten aber sicher gehen, daß die jeweils aktuelle Version auf diesen Disks erscheint.
  - Selbiges gilt für CD-Roms mit Ausnahme der Aminet CD's.
  - dies gilt ebenso für kommerzielle Nutzung oder Verbreitung, die ohne SCHRIFTLICHE Genehmigung der Autoren ausdrücklich untersagt ist.

Wir, die Autoren des Programms, übernehmen keinerlei Haftung für die durch dieses Programm eventuell entstandenen Schäden an Hard- und Software. Die Nutzung des Programms erfolgt auf eigene Gefahr.

WIR, DIE AUTOREN, BEHALTEN UNS DAS RECHT VOR, DEN VERTRIEB DES EAGLEPLAYERS EINZELNEN ANBIETERN ZU UNTERSAGEN, FALLS DIESE SICH NICHT AN UNSERE COPYRIGHT-BEDINGUNGEN HALTEN. Der Eagleplayer wurde nicht geschrieben, damit irgendwelche Leute sich daran eine goldene Nase verdienen können. Er soll die Amigawelt um ein sinnvolles Musikabspielprogramm zum -fast- Nulltarif erweitern. Und wer nicht vorhat, sich registrieren zu lassen, wird auch weder geächtet noch verbannt.

Der Eagleplayer ist KEIN SASG Produkt und kostet daher keine DM30 ! Updates sind in jedem Fall kostenlos !

```

xfermaster.library          (C) Georg Hörmann
req.library                 (C) Colin Fox & Bruce Dawson
reqtools.library,powerpacker.library (C) Nico Francois
lh.library                  (C) Stephan Furmann
LHA                          (C) Stefan Boberg

```

```

Installer Programm          (C) Commodore Büromaschinen GmbH
  Installer and Installer project icon
  (c) Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved.
  Reproduced and distributed under license from Commodore.

```

## 1.3 distribution

### 1.2 Distribution

---

Der Eagleplayer ist ein Programmpaket, das auf 2 Disketten oder als LHA

---

Archiv auf dem Aminet verbreitet wird.

Nach Möglichkeit sollten alle Dateien auf den beiden Disketten verbleiben, da es ansonsten beim Booten oder Installieren Probleme geben kann.

Bitte beachten Sie: Wenn der Name der Disketten geändert wird (Beispielsweise für eine PD-Serie), so muß diese Änderung auch in das "Install"-Script eingetragen werden, ebenso in den Scripts: "Deutsche\_Anleitung" und "English\_Documentation".

Die Weitergabe ist nur erlaubt, wenn folgende Dateien bzw. Directories unverändert mitgegeben werden:

```
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
    Delete
    Failat
  Prefs (dir)
    Env-Archive (dir)
      EaglePlayer (dir)
        Eagleplayer.Config          Eagleplayers.Batch
        Userprograms.Batch
  Eagleplayers (dir)
    FC 1.3                          FC 1.4
    Jamcracker                       Oktalyzer
    SonicArranger                    TFMX Pro
  EP-UserPrograms (dir)
    DirListViewer                    Patternsroll
    Quadrascope                      Space-scope
    Time
  libs (dir)
    XFD (dir)
      XFD_FIMP_LHLib_XPK.Slave
    Compressors (dir)
      xpkFAST.library               xpkIMPL.library
      xpkLHLB.library               xpkNUKE.library
      xpkRDCN.library               xpkSMPL.library
      xpkSQSH.library
    amigaguide.library              commodities.library
    diskfont.library                explode.library
    icon.library                     lh.library
    powerpacker.library              req.library
    reqtools.library                 xpkmaster.library
  Utilities (dir)
    AmigaGuide                       AmigaGuide.info
    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           Developer.lha
Disk.info                   Docs.lha
Eagleplayers.lha           Userprograms.lha

```

## 1.4 was ist und kann der eagleplayer ?

### 2.1 Was ist und kann der Eagleplayer ?

-----

Der Eagleplayer ist ein Programmpaket zum komfortablen Abspielen von Musik verschiedenster Soundformate. Er ist NICHT zum Erstellen von Musikstücken nutzbar, und sollte auch nicht mit einem Musikeditor, Sequencer, Tracker usw. verwechselt werden. Er besitzt bereits die Abspielprogramme für Sound/Noise/Protracker, Startrekker(AM) & AudioSculpture(AM) sowie weitere ProTracker-Packer und einen speziellen Customplay - Port intern und ist in der Lage, weiterhin benötigte, externe Player nachzuladen. Der Vorteil dieser Methode liegt auf der Hand: Es brauchen nur die jeweils benötigten Player mitgeladen werden, was Ladezeit und Speicher spart, außerdem steigert sich die Länge des Hauptprogramms nicht ins Unendliche. Alle internen SoundTrackerRePlayer unterstützen Song-Loading (außer die ProTracker-Packer). Die Songs und Samples können gepackt sein. Der Instrumenten-Pfad kann über das Konfigurationfile eingestellt werden. Die Samples dürfen gepackt sein. Sie werden alle als RAW-Daten geladen.

Einige Fähigkeiten des Eagleplayers:

- Modularer Aufbau
- unterstützt in allen seinen internen Windows AppWindow (Kick2.0+)
- Äußerst komfortable Bedienung: z.B. kann die Auswahl des zu spielenden Musikstückes direkt über das
  - Das PullDown-Menu
  - erfolgen
- Iconify-Modus (AppItem, AppIcon, LittleWin)
- Verzeichniswechseln im Pulldown Menü
- die Anzahl der nachzuladenden Player ist nur durch den vorhandenen Speicher begrenzt (Kap. 3.4)
- alphabetische Ordnung der Player im Playerwindow
- Auswahl von Playern und Userprogrammen mittels Filerequester unterstützt Multiselect, sowie AddPlayer-Dir Funktion und AddUserDir Funktion

- Player erlaubt/ausgeschaltet
  - Möglichkeit von bis zu 32 extern zu ladenden Userprogrammen (z.B. Analyzer-, Patternscroller wie beim Protracker, ← Listenverwaltung)
  - umfangreiche Einstellungsmöglichkeiten (Fade, Randomplay, LoadNext ...)
  - Geschwindigkeitsregelung über Schieberegler (Kap 4.1.1)
  - alle Stimmen getrennt ein- und ausschaltbar
  - Statusanzeige durch Scroller
  - sorgfältig und umfangreich angepaßte Playroutinen, fast alle funktionieren auf A4000 und so gut wie überall lassen sich Lautstärke und Balance einstellen (Siehe Player.Dok)
  - hat 7 interne Depacker kann Powerpacker- (auch ohne XPK), FileImploder-, Crunchmania- ← und LH- (mit Libraray) Module entpacken (XPK-Files sowieso) sowie Stonecracker4.0. Außerdem wird die neue XFDMaster.library von Georg Hörmann unterstützt, die durch ihren modularen Aufbau auch für zukünftige Packerformate geeignet ist.
  - umfangreiche Moduleinfos, wenn der Replayer entsprechend angepaßt wurde
  - eine Version für alle Kickstarts (ab Kick1.2)
  - Gadgets passen sich beim Window öffnen automatisch der Farbpalette an (3D-Lock)
  - zeitmäßiges Styling
  - interne SoundTrackerplayer können Songs laden
  - PlayFaster-Gadget
  - mächtiger ARexx-Port
  - Randomsong/AutoSubSong-Funktion
  - LoadPrev und LoadBefore Funktion
  - Eagleplayer-Batch (Laden des Players, erst wenn er benötigt wird, somit Speicher- und Ladezeitparende Arbeitsweise)
  - UserPrg-Batch (Laden des Userprogrammes, wenn er benötigt wird)
  - LHA-Extraktor
    - verschiedene Filerequester (ASL, Req, ReqTools, None)
  - eigener Textrequester
  - wahrscheinlich der einzige Player, der Protracker timingrichtig und ohne Aussetzer auf dem A4000 spielt !!!
    - läuft problemlos im Multitasking
    - Abspielen der Protrackermodule (und aller internen Packer) aus dem FastMem möglich
    - externe Analyzermodule laufen mit niedriger Priorität, dadurch werden rechenintensive Programme kaum gebremst
    - Spielzeitberechnung für verschiedene Musikformate (Protracker, Oktalyzer, SonicArranger)
    - ist ein Commodity (Hotkey-Aktivierung, Steuerung über Exchange)
    - 100% in Assembler geschrieben
-

- Ein-und Ausblenden der Musik möglich
- Abspeichern der kompletten Konfiguration in eine editierbare ASCII-Datei mit nur einem Mausklick möglich (inklusive aktuelles Musikstück, Zustand und Position der Userprogramme, aktuelles Verzeichnis, Packmodus beim Speichern usw.)
- Abspielen eines zufällig ausgewählten Modules aus dem aktuellen Verzeichnis sofort beim Programmstart.

Internet Release: (als LHA Datei zusammengepackt)

```

Eagleplayer (dir)
  InstallData (dir)
    Prefs (dir)
      Env-Archive (dir)
        EaglePlayer (dir)
          Eagleplayer.Config           Eagleplayers.Batch
          Userprograms.Batch
    libs (dir)
      Compressors (dir)
        xpkCRMS.library                xpkLHLB.library
        xpkNUKE.library                xpkSMPL.library
        xpkSQSH.library
      CrM.library                       lh.library
      PowerPacker.library               req.library
      reqtools.library                 xpkmaster.library
  ArexxExamples.lha                   Developer.lha
  Docs.lha                             Eagleplayer
  Eagleplayer.info                     Eagleplayers.lha
  LhA                                   Userprograms.lha
  install                              Install_Deutsch.info
  Install_English.info                 Liesmich_SOFORT.Dok
  Liesmich_SOFORT.Dok.info             Readme_NOW.Doc
  Readme_NOW.Doc.info

```

Ein besonderer Vorteil des Eagleplayer gegenüber vielen Konkurrenten ist, daß er mit den allermeisten

Playroutinen  
(diejenigen, die nicht hinhauen,

siehe

Problemecke  
) unabhängig von der Bildwiederholrate arbeitet,

also genauso schnell unter PAL, als auch NTSC oder Productivity spielt.  
Durch seinen modularen Aufbau ist eine optimale Anpassung an das jeweilige System und die Vorlieben seines Nutzers möglich.

## 1.5 entstehung des eagleplayers

### 2.2 Entstehung des Eagleplayers

---

Alles fing mit dem Phenomena-Demo "Enigma" an. Dort wurde NoisePacker verwendet. Die Musik hatten wir gerippt, aber es gab keinen Player. Nun wurde schnell ein Tool zum Abspielen dieses wirklich guten Modules geschrieben. Irgendwann wurden

---

dann auch andere Playroutinen implementiert (NoiseTracker...). Daraus entstand dann dann der Megaplayer 1.0, der am 16.04.1992 released wurde. Er konnte zu dem Zeitpunkt 16 Playroutinen abspielen. Daraus wurden dann über 25 und das Hauptprogramm wurde immer unübersichtlicher.

Im Juni 1992 entschied sich Jan, nachdem dauernd über seinen Noiseplayer-ähnlichen Megaplayer gelästert wurde, ein neues Soundplayerprogramm zu schreiben, das Maßstäbe setzen sollte. Er programmierte fleißig, erstellte eine ansprechende Oberfläche, versuchte möglichst noch mehr Playroutinen heranzuschaffen und diese in das Programm einzubauen. Es wurde ein Konzept für externe Player entworfen.

Da das damalige Konzept von Delirium besser war, wurde dieser Port eingebaut. Später wurde unser Port für externe Player so stark erweitert und verbessert, daß die Nutzung unserer Player nur noch mit dem Eagleplayer möglich war.

Da wir auf grafische Spielereien nicht verzichten wollten, entwickelten wir ein System für externe, asynchron laufende Analyzerprogramme, die wir in der frommen Absicht, diesen weiteren Port offen zu dokumentieren, anfangen zu programmieren. Mit der Zeit nam diese Arbeit allerdings Dimensionen an, die ein Außenstehender wohl kaum begreifen würde. Daher der seltsame Name. Aus diesem Analyzerport wurde ein universelles Interface für verschiedenste Anwendungen, beispielsweise Konverter (Noiseconverter), Listenverwaltung, Moduleinformationen, PubscreenManager, Patternscroller und vieles mehr.

## 1.6 was fehlt in dieser version noch ?

### 2.3 Was fehlt in dieser Version noch ?

---

Eine ganze Menge, bis heute stehen noch aus:

- \* Sampleinfo (mit Abspeicherung einzelner Samples, Sampleabspielen usw.)
- \* UserPrgr Manager als Überblick über die geladenen UserProgramme

Anmerkung: die mit '\*' bezeichneten Funktionen wird es vorraussichtlich nur in der registrierten Version geben! Bitte beachten Sie, daß Sie vom registrierten Eagleplayer den aktuellen Stand erhalten, d.h. es können noch einige Sachen fehlen, die geplant waren und aus Zeitgründen noch nicht fertig sind. Diese Teilprogramme können dann aber per Keyfile in späteren Versionen auch genutzt werden.

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

### 2.4 Wann gibt es eine neue Version des Eagleplayers ?

---

Das steht in den Sternen. Da wir, Eagleeye und Buggs, neuerdings nur noch am Wochenende zu Hause sind, kann es etwas länger dauern, bis die nächste Version fertiggestellt ist.

User, die sich registrieren lassen wollen, möchten wir hiermit bitten, etwas Geduld zu haben, bis die Antwort unsererseits kommt (kann schon ein paar Wochen dauern).

---

## 1.8 update-service

### 2.7 Update-Service

-----

Jeder registrierte User kann sich bei uns die neuste Version bestellen, ohne Gebühren zu bezahlen. Allerdings muß er 2 Disketten und einen an sich ausreichend frankierten Rückumschlag mitschicken !!!

Auf den Disketten muß sich das Keyfile des registrierten Users befinden. Bitte schicken Sie auch Ihr Konfigurationsfile sowie ihre eigenen Eagleplayer/Userprogramm-Batchdateien mit.

Weiterhin bitten darum, die

Updatekarte

auszufüllen und vor allem die User-

Nr (sie finden sie, wenn sie im "Project Menü" des Hauptfensters "About" anwählen und dann "Key" anklicken) anzugeben, ohne diese ist es uns unmöglich, zu reagieren.

Bitte beantworten Sie folgende Fragen:

- Sind Sie mit der Zustellung des Eagleplayer zufrieden ?
- Gibt es Probleme bei speziellen Rechnerkonfigurationen ?
- Laufen alle Funktionen für registrierte User einwandfrei ?
- Haben Sie mehr bzw. weniger von Ihrem registrierten Eagleplayer erwartet ?
- Ist die Dokumentation ausreichend ?
- Welche Funktionen sollten unbedingt noch eingebaut werden ?
- Arbeitet der Eagleplayer mit allem Programmen zusammen ?
- Funktionieren die Playroutinen korrekt ?
- Benutzen Ihre Freunde auch den Eagleplayer ? Wenn nicht, wieso nicht ?
  - Verbraucht Ihnen der Eagleplayer zuviel Rechenzeit ?  
(Wenn ja, schauen Sie sich doch die Version 2.0 eines gewissen Konkurrenzplayers an :) )

Wenn Sie Fehler beschreiben, schauen Sie unbedingt in das Kapitel

Fehlerbeschreibung

!

Wir werden dann auf diese Disketten die aktuellen Eagleplayer kopieren und ihn zurückschicken. Sollte noch derselbe Eagleplayer aktuell sein, wird auf die nächste Version gewartet.

## 1.9 fehlerbeschreibung

### 2.8 Fehlerbeschreibung

-----

Wenn Sie uns einen Fehler im Eagleplayer beschreiben wollen, sollten Sie einige Sachen beachten und versuchen, den Fehler so rekonstruierbar wie möglich zu machen.

Eine Aussage "Der Eagleplayer stürzt manchmal beim Laden ab", hilft uns nicht im geringsten. Es könnte sein, daß das Modul defekt ist oder der Depacker

---



falsch entpackt, daß jemand den Speicher zersägt hat. Da gibt es viele Varianten. Hier nun einige Sachverhalte, die Sie nach Möglichkeit unbedingt beantworten sollten.

- \* Sind Sie registrierter User ?
- \* Welche Eagleplayerversion nutzen Sie ?
- \* Hardwarekonfiguration angeben !
- \* Welche Patches haben Sie installiert ?
- \* Freier Speicher zur Zeit des Absturzes ?
- \* Welche Player/Userprogramme waren geladen ? Ist es offensichtlich egal, ob Userprogramme oder Player geladen wurden ?
- \* Tritt der Fehler auch bei anderen Rechnerkonfigurationen auf ?
- \* Konfigurationsfile angeben !
- \* Bei welchem Module gibt es Probleme ? Bitte immer mitschicken !!!
- \* Tritt der Fehler nur bei Rechnern mit Cache auf ?
- \* Wie waren die internen Einstellungen des Eagleplayers ? Ist es offensichtlich egal, wie diese sind ?
- \* Wurde das Module richtig erkannt ?
- \* Sind die Gadgets schon freigegeben worden (bei Moduleloading) ?
- \* War das Hauptwindow (noch) offen ?
- \* Bei welcher Kickstartversion trat der Fehler auf ? Ist es offensichtlich egal ?
- \* Welche Userprogramme waren aktiv ?

## 1.10 keyfile-benutzung

### 2.9 Keyfile-Benutzung

-----

Registrierte User erhalten von uns ein Key-File. Dieses muß nach Devs: kopiert werden und ermöglicht ihnen den Zugang zu den bisher gesperrten Funktionen.

Das Keyfile ist personengebunden. Es enthält die volle Adresse des Users sowie die UserNr und weitere Daten. Sollten wir erfahren, daß ein Keyfile an Dritte weitergegeben wurde, werden wir uns gerichtlich gegen diese Personen vorgehen. In der nächsten Version des Eagleplayers wird diese UserNr dann logischerweise gesperrt.

Alle Keyfiles können von nachfolgenden Eagleplayer-Versionen genutzt werden. Es ist durchaus möglich, daß spätere DEFECT-Produktionen auch mit diesem Keyfile laufen. Mehr wird nicht verraten !!!

## 1.11 systemanforderungen

### 3.1 Systemanforderungen

-----

Der Eagleplayer benötigt zum Start mindestens Kick 1.2 und 512 KByte Ram. Es muß aber dazu gesagt werden, daß einige Funktionen des Eagleplayers, so z.B. Appwindow (siehe dort) nicht unter 1.2/1.3 nutzbar sind. Gleiches gilt für einen Großteil der von uns verfaßten Userprogramme. Auf Rechnern, die noch mit 2.0 Quasi-Versionen arbeiten (alle 36-er Kickstart Versionen), garan-

---

tieren wir für nichts, da diese Betriebssystemversionen noch nicht ausgereift waren. Wenn wir im folgenden von "2.0" sprechen, meinen wir die Betriebssystemversionen ab 37.175 (OS2.04), die serienmäßig ab A500+, A2000E als ROM vorliegen. Eingeschlossen sind die darauffolgenden Versionen 3.0 usw., die im A1200, A4000 verwendet werden. Tests auf diesen Maschinen waren erfolgreich. (neuerdings wird auf A1200/50MHz programmiert)  
Weiterhin erscheint des öfteren der Hinweis (r.V), dies bedeutet lediglich, daß diese Funktion nur in der registrierten Version benutzbar ist oder nutzbar sein wird.

Natürlich hat der Amiga an einen Monitor mit Lautsprechern oder an eine Anlage/Henkelmann angeschlossen zu sein, da ansonsten nichts zu hören ist. (wie auch?)

Es ist darauf zu achten, daß nicht gleichzeitig ein anderes Soundprogramm arbeitet, das die Tonausgabekanäle belegt. In diesem Fall gibt der Player nach dem Laden eines Modules eine Meldung aus, siehe bei "Fehlermeldungen"

Achtung: Auch wenn hier davon gesprochen wird, daß der Eagleplayer mit 512KByte Speicher läuft, so ist die Lage derzeit so, daß mit allen unseren bis jetzt fertigen Userprogrammen und allen zur Verfügung stehenden externen Playern selbst 1 Megabyte Ram nicht ausreicht. Wer also über nicht viel Speicher verfügt, sollte abwägen, welche Player er braucht oder auf die

Batch

Funktionen

zurückgreifen, durch die sich der Speicherverbrauch auf ein Minimum reduzieren läßt

## 1.12 benötigte zusätzliche dateien

### 3.2 Benötigte zusätzliche Dateien

Der Eagleplayer ist nun auf keine externe Library mehr angewiesen.

Ansonsten versucht der Eagleplayer die XPKMaster.Library, die PowerPacker-Library, CRM.Library, XFDMaster.library, sowie die LH.Library zu öffnen, um dementsprechend gepackte Files zu laden. Diese Libraries sind NICHT UNBEDINGT zum Start notwendig, aber durchaus zu empfehlen, wenn Sie solcherart gepackte Module haben, sowieso.

Weiterhin wird versucht die ASL.Library zu öffnen. Sie muß nicht vorhanden sein.

Der Eagleplayer versucht ein  
Konfigurationsfile  
zu laden

Zum Betrieb der weiteren Filerequester sind außerdem die "req.library" und die "reqtools.library" notwendig.

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

### 3.3 Installation auf Diskette, Festplatte, ...

---

Wer zu faul ist, das Ganze manuell zu erledigen, kann dies mit dem beiliegenden Installationsskript tun. Alle Aktionen werden dann durch Hilfstexte erklärt und die Dateien werden automatisch komplett entpackt. Es wird dazu empfohlen, nicht von Diskette zu booten, sondern von Festplatte, da so einige wichtige Verzeichnisse automatisch gefunden werden können.

Um den Eagleplayer per Hand zu installieren, sind folgende Schritte nötig:

- als erstes sind die
  - o.g.
  - Librarie(s) nach "Libs:" zu kopieren
  - (Verzeichnis "libs" auf der Bootdiskette/Bootpartition)
- der zweite Schritt ist das Kopieren der Datei "Eagleplayer" in das gewünschte Verzeichnis (ein Beispiel wäre: "DH0:Music/Players/" oder "DF0:" ...)
- als nächstes sind die sofort beim Programmstart zu ladenden Players zu bestimmen und in eines der folgenden Verzeichnisvarianten zu kopieren. (der Eagleplayer sucht in der untenstehenden Reihenfolge und lädt dann vorerst nur die externen Player, die sich im erstgefundenen Verzeichnis befinden)
  - "Eagleplayers/" (im aktuellen Verzeichnis)
  - "Sys:Eagleplayers/"
- als letztes sind noch die Userprogramme zu installieren, indem entweder das Verzeichnis "EP-Userprograms" oder "Sys:EP-Userprograms" angelegt wird, in das die gewünschten Userprogramme zu kopieren sind.

Beispiel:

```
copy Eagleplayer to dh0:Music/Players/
makedir dh0:Music/Players/Eagleplayers
makedir dh0:Music/Players/EP-Userprograms
copy eagleplayers/ to dh0:Music/Players/Eagleplayers/ all
copy EP-Userprograms/ to dh0:Music/Players/EP-Userprograms/ all
```

Achtung ! Ab Sofort sind die externen Players und Userprogramme mit LHA archiviert auf der zweiten Disk zu finden. Falls Sie nicht mit LHA klarkommen, nutzen Sie das automatische Installscript.

## 1.14 start des eagleplayers von cli & shell

### 3.4 Start des Eagleplayers von CLI/SHELL

---

Zum Start von Cli/Shell aus, genügt ein einfacher Aufruf des Players. Vorher wechseln Sie bitte in das Verzeichnis (DOS-Befehl 'CD'), in dem sich der Player samt Playern und Userprogrammen befindet. Dies ist für den Fall wichtig, in dem sich das Playerverzeichnis (Deli- oder Eagleplayers), sowie das Userprogramsverzeichnis im selben Verzeichnis befinden, wie der Eagleplayer selbst.

---

Sofern die Eagleplayers und Userprograms im Verzeichnis "SYS:" liegen, ist der 'CD'-Befehl überflüssig, es reicht, den Player einfach aufzurufen. Der Befehl 'Run' ist nicht notwendig, um den Eagleplayer von der Cli/Shell loszulösen, nach dem Start koppelt sich der Eagleplayer von der Cli/Shell ab, so daß diese sofort weiter nutzbar ist und auch geschlossen werden kann.

## 1.15 start von der workbench aus

### 3.5 Start von der Workbench aus

---

Zum Start von der Workbench genügt ein einfacher Doppelclick, durch den der Eagleplayer geladen wird.

Beachten Sie bitte, daß der Eagleplayer keine Tooltypes auswertet.

## 1.16 das konfigurationsfile

### 3.6 Das Konfigurationsfile

---

Der Eagleplayer lädt ab Version V1.10 eine Konfigurationsdatei. Sie besteht aus einer Reihe von Kommandos, die als Strings eingegeben werden. In der Konfig kann u.a. angegeben werden, welches das aktuelle Playerverzeichnis ist oder ob Fadein ein/ausgeschaltet werden soll, usw.

Die Datei befindet sich normalerweise im "ENV:Eagleplayer/" Verzeichnis. Es wird aber beim Start des Players versucht, die Konfig vom aktuellen und danach vom S: Directory zu lesen. Das Konfigurationsfile ist nicht unbedingt erforderlich zum Start des EaglePlayers.

Beachten Sie bitte, wenn Sie eine Konfig saveen wird der aktuelle Stand des Eagleplayers abgespeichert, also Modulename, derzeitige Userprogrammstellungen, sämtliche Pfade alle Preferenceeinstellungen.

Die Kommandos zum Konfigurationsfile finden Sie in der Config auf der Main-Diskette oder wenn Sie eine Konfig saveen.

Es müssen in einer Konfig-Datei nicht alle Kommandos vorkommen.

Die Kommandos des Konfigurationsfiles werden teilweise auch bei den

Arexx-Kommandos  
verwendet.

## 1.17 das hauptfenster

### 4.1 Das Hauptfenster

---

Sofort nach dem Start sollten die meisten der Schalter des Hauptfensters gesperrt sein (durch Rasterung erkennbar). Um nun ein Musikstück zu laden, ist es einfach nur nötig, das Diskettensymbol anzuklicken. Danach öffnet sich ein Filerequester, in dem man sich die gewünschte Datei auswählt und mit "OK!"

---

bestätigt. Danach können je nach dem aktuellen RePlayer folgende Schalter freigegeben sein:

```

|
|          *
|          | Eagleplayer V1.53 Generic/Registered          | | |
|-----|
| |
|          1
|          |   | Volume |
|
|          |
|          zz
|          |
| | _|
|          ?
|          |
|          P
|          |-----|---|
| _|
|          2
|          |   | Balance|
|
|          |
|          ] [
|          |
|
|          | _|-----|
| -|
|          3
|          |
|          |<
|          | @ { "<" link PrevSub } |
|          <<|
|          |
|          [|>
|          |
|          |>
|          |
|          >>
|          |
|          >>|
|          |
|          >
|          |
|          >|
|          |
|          [|
|          |
|          ^
|          |
|          00
|
| | |-----|
| |

```

4

| Welcome to Eagleplayer |

## 1.18 speed

Speed - Regler für die Abspielgeschwindigkeit, steht dieser in der Mitte, ist die Normalgeschwindigkeit eingestellt, eine Bewegung nach oben erhöht die Geschwindigkeit, eine Bewegung nach unten verlangsamt die Geschwindigkeit

## 1.19 plwi

Open/Close - Hiermit läßt sich das Playerinfo-Fenster öffnen/schließen  
Playerwindow

## 1.20 voi

Voices - die 4 Stimmen der Standard-Amigas lassen sich hiermit einzeln ein- und ausschalten, ist der Schalter "reingedrückt" (und unterscheidet sich in den Rändern somit von den anderen Schaltern) , ist die jeweilige Stimme ausgeschaltet, ansonsten an

## 1.21 prevmod

previous

Module - veranlaßt den Player, das vorherige Modul der aktuellen Liste zu laden, ist das aktuell gespielte Modul bereits das erste, so wird dann das letzte geladen. Wenn Load Dir ausgeschaltet ist, wird die Funktion gesperrt.

## 1.22 prevsub

previous

(Sub)song - für Soundsysteme, bei denen ein Module mehrere Songs enthält, diese Funktion spielt den vorherigen Untersong ab, ist der gespielte bereits der erste, passiert nichts. Wenn das Module nur einen SubSong unterstützt, wird das Gadget gesperrt.

## 1.23 prevpatt

previous

Pattern - spielt das vorherige Pattern, ist bereits das erste erreicht, wird das letzte Pattern gespielt.

Für weniger Versierte : Ein Pattern ist mit einem Notenblatt zu vergleichen, einige Soundsysteme bauen darauf auf, indem eine solche Liste von Befehlen nacheinander ausgeführt wird, diese Befehlslisten (oder Notenblätter, eben Patterns genannt) lassen sich meist beliebig kombinieren.

## 1.24 restart

Replay Song - das geladene Musikstück wird noch einmal von vorn gespielt.

## 1.25 play

1Play / Pause - der Eagleplayer stoppt oder startet den Song, (wechselt den Modus ständig) je nachdem ob gerade gespielt wurde oder nicht.

## 1.26 fast

Fast Play - "schneller Vorlauf", erleichtert die Suche nach einer bestimmten Passage im Titel.

## 1.27 nextpatt

next Pattern - spielt das nächste Pattern (siehe auch oben) ab, wenn das Ende des Musikstückes erreicht wurde, wird das erste Pattern abgespult (ich meine abgespielt).

## 1.28 nextsub

Next (Sub)song - spielt den nächsten Untersong eines Musikstückes mit mehreren UnterSongs. Wenn das Module nur einen Song hat, wird die Funktion gesperrt.

## 1.29 nextmod

Next Module - der Player lädt das nächste Musikstück der Liste. Wenn Load Dir ausgeschaltet ist, wird die Funktion gesperrt.

---

### 1.30 stop

Stop - hält den Song an, setzt den Zeiger für die Position wieder auf den Anfang des Stücks.

### 1.31 eject

Eject Module - stoppt das aktuelle Module (falls dieses gespielt wird) und entfernt es aus dem Speicher.

### 1.32 disk

Load Module - öffnet den Filerequester zur Anwahl eines anzuhörenden Musikstückes, es dürfen auch mehrere ausgewählt werden (Shift-Taste gedrückt halten); ist dies der Fall, wird die Liste der ausgewählten Dateien übernommen und kann über die Funktionen 'previous Module' und 'next Module' genutzt werden; weiterhin bietet sich beim Eagleplayer die Möglichkeit, die Musikstücke direkt über das Pulldownmenü aufzurufen (Modules-Directory); wird nur ein Modul selektiert, so wird je nach Einstellung (siehe bei Kapitel 4.1.2 'Pulldown Menüs') das gesamte Verzeichnis eingelesen, dabei werden die Dateien, die mit 'smpl.' beginnen, sowie mit '.nt', '.nt.pp', '.txt' und '.s' aufhören automatisch "versteckt", also nicht berücksichtigt

### 1.33 icon

Iconify - verkleinert das Hauptfenster auf ein Minimum, beziehungsweise schließt es je nach Einstellung ganz und legt in diesem Falle entweder ein sogenanntes "Appicon" auf der Workbench ab oder fügt einen Menüeintrag im Tolls-Menü der Workbench dazu, (beide Funktionen sind erst ab OS2.04 zugänglich) um das Hauptfenster des Eagleplayers wieder zu öffnen, genügt ein Doppelclick auf das Appicon beziehungsweise eine Anwahl des Menüitems

### 1.34 filter

Filter - Schaltet den Tiefpaßfilter des Amigas ein und aus (ohne Wirkung beim Amiga 1000 und Amiga 2000A)



### 1.35 unnamed.1

Playerfenster - Öffnet/schließt das Fenster, in dem Sie sich über die externen/internen Abspieler einen Überblick verschaffen können.

### 1.36 about

Info About

Module - Scrollt einen Informationstext durch das Hauptfenster, der Angaben über das aktuelle Musikstück und den verwendeten Player macht. Der Umfang des Textes variiert je nach Player, im ungünstigsten Fall wird lediglich der Name der geladenen Datei und deren Länge ausgegeben. Wenn kein Module geladen wird, wird das Aboutwindow des Eagleplayers geöffnet.

### 1.37 close

Close - das Close-Gadget schließt alle Fenster, die zum Eagleplayer gehören. Unter Kickstart 1.3 wird der Eagleplayer sofort beendet. Ist unter Kickstart 2.0 oder höher die "Quit Eagle" Option aus dem "Special->Preferences Menü" aktiviert, geschieht selbiges. Ansonsten kann man das Hauptfenster wieder öffnen, indem man das Hotkey aktiviert (normal: CTRL-ALT-E), es über "Exchange" öffnet oder aber den Eagleplayer neu lädt. (dieser sendet dem im Speicher befindlichen automatisch das Signal, sich zu öffnen)  
>> Um den Eagleplayer zu beenden, muß entweder die oben beschriebene Option aktiviert sein oder aber Sie wählen "Quit" aus dem "Project"-Menü.

### 1.38 vol

Volume - Regelung der Lautstärke.

### 1.39 bal

Balance - Lautstärkeverhältnis zwischen linkem und rechtem Kanal beeinflussen.

### 1.40 steuerung über die pulldown-menüs

---

#### 4.1.2 Steuerung über die Pulldown-Menüs

-----

Der Eagleplayer besitzt folgende Pulldown-Menüs:

Project Commands Special Modules-Directory

Die Funktionen des 'Commands'-Menüs sind identisch mit denen, die im vorherigen Kapitel 'Gadgets' erläutert wurden, es wird hier nicht weiter darauf eingegangen.

Die Funktion des "Modules-Directory" ist im vorherigen Kapitel unter "Load Module" mit erläutert worden. Achtung: in der unregistrierten Version kann nicht in Unterverzeichnisse geschaltet werden, Sie haben dazu den Filerequester zu benutzen. (DAS NERVT MEHR ALS EIN REQUESTER ODER

?)

)

Das Project Menü

-----

Load Module - lädt ein Module (siehe im Kapitel zu den Gadgets)

Save Module - nur in der registrierten Version zugänglich, öffnet den Filerequester, um Datei zu selektieren, in die das aktuell geladene Modul abgespeichert werden soll

About

Module - Scrollt einen Informationstext durch das Hauptfenster, der Angaben über das aktuelle Musikstück und den verwendeten Player macht. Der Umfang des Textes variiert je nach Player, im ungünstigsten Fall wird lediglich der Name der geladenen Datei ausgegeben.

Wenn kein Module geladen wird, wird das InfoRequester über den EaglePlayer ausgegeben.

Add Player - Öffnet den Filerequester, um einen zum Eagleplayer hinzuzufügenden Player zu auszuwählen.

Add P1Dir - Öffnet den Filerequester, um ein Directory mit Eagleplayern hinzuzufügen

Show Players - Öffnet das Playerinfofenster (siehe Kapitel 4.2.1)

Load Config - Konfiguration laden, nur in der registrierten Version verfügbar.

Save Config - Konfiguration sichern, nur in der registrierten Version verfügbar.

Delete File - Datei löschen

Help - Online Hilfe, nur in der registrierten Version verfügbar.

Iconify - siehe unter "Iconify" im Kapitel 4.1.1. "Gadgets"

- About - Scrollt Informationen über das Programm durch das Hauptfenster.
- Hide - Alle Windows werden geschlossen.
- Quit - beendet den Eagleplayer.

#### Das Special-Menu ~~~~~

#### Preferences: ~~~~~

- ScrollInfos - Die Informationen wie Filter on/off werden gescrollt.
- FlashPointer - Beim Entpacken eines Modules flackert der Mauspfel.
- Fadein - Ein Module wird mit steigender Lautstärke eingeblendet.
- FadeOut - Ein Module wird mit abnehmender Lautstärke ausgeblendet
- Filter - Gibt an, ob der Filter beim Initialisieren des Modules gesetzt werden soll oder nicht.
- MasterVolume - Ist Mastervolume gesetzt, werden die Lautstärkewerte für die UserProgramme verändert, das heißt in Abhängigkeit von der Lautstärkeeinstellung des Eagleplayers ändert sich auch der Ausschlag von Analyzer, Levelmeter, Levelgraph usw.
- AutoPassword - Wenn bei einer Funktion ein Password benötigt wird, wird das zuletzt eingegebene automatisch benutzt.
- Use SongName - Beim Saven des Modules wird der Songname benutzt, falls vorhanden. (r.V)
- Use Prefix - Beim Saven des Modules wird ein Prefix vor den Namen gehängt, falls durch den Replayer unterstützt. (r.V)
- Quit Surface - Soll beim Beenden einer Userprogrammoberfläche der Eagleplayer beendet werden ?
- Quit Eagle - Soll der Eagleplayer beim Betätigen des Close-Gadgets beendet werden ?
- SaveT - Soll bei gepackten Mods, die normalerweise nicht packbar sind, das File nach "T:" gespeichert und danach ungepackt geladen werden, so daß solche Modules gepackt werden dürfen ?
- LHA-Extraktor - legt fest, ob LHA-gepackte Dateien entpackt werden sollen (mittels LHA aus dem "C:"-Verzeichnis)  
Es wird dringend empfohlen, einen anderen Packer z.B. XPK-SQSH, XPK-SMPL oder LH.library mit Sample-

mode zu verwenden, da dort sowohl Packrate, als auch Entpackgeschwindigkeit besser sind

LoadFast - Legt fest, ob die nächste Datei nach Möglichkeit ins Fastmem statt ins Chipmem gelegt werden soll (möglich bei MusicMaker, Oktalyzer 8-stimmig und allen internen Trackern bis auf Startrekker/Audiosculpture, sowie TFMX)

~~~~~

Eject Players - Alle unbenutzten Eagleplayer werden gelöscht, sobald ein neues Modul geladen wird. Beachten Sie bitte, daß wenn diese Option ausgeschaltet ist, jeder Player über die Batchdatei nur ein einziges Mal eingeladen wird.

Eject UserPrg - Wenn ein Userprogramm geschlossen wird, dann wird es gelöscht.

Use PlBatch - Das Batch-Loading wird ein/ausgeschaltet

Use UPrgBatch - Das Batch-Loading wird ein/ausgeschaltet

Load PlBatch - Eine Eagleplayers -BatchDatei wird geladen

Load UPrgBatch - Eine UserPrg -BatchDatei wird geladen

Modules-Prefs

~~~~~

Rescan Dir - Das Directory wird neu geladen.

~~~~~

Loaddir - Das Directory wird ins PullDown Menü geladen. Wenn Load Dir ausgeschaltet ist, werden NextModule und PrevModule gesperrt. Ein LoadNext findet ebenfalls nicht statt.

Use EPDir - Soll ein EPDir geladen werden oder das ganze Dir gescannt werden. Das EPDir ist auch auf Rechner mit Dirchaching schneller. (nur für registrierte User)

Autodir - Wenn das Directory geändert wird oder die Diskette gewechselt wird, wird das Dir neu geladen.

Notify - Ab Kick2.0 kann die Notify-Funktion eingeschaltet werden, welche der normalen Autodir Funktion vorzuziehen ist.

Achtung ! Es kann passieren, daß die Notify-Funktion mit der EPDir-Funktion kollidiert und plötzlich ständig das Verzeichnis neu geladen wird. Zur Behebung einfach "Autodir" und "Notify" oder "LoadDir" ausschalten.

Scan Always - Soll das Directory jedes Mal geladen werden, wenn ein Module über File-

requester, Arexx oder ein externes
Userprogramm geladen wird ?

~~~~~

Show DirNames - Sollen Dirnamen ins Pulldown-Menü über-  
nommen werden ?

HideAll - Sollen beim Direktoryladen alle als  
nicht-Module bekannten Filetypes (\*.asm)  
versteckt werden ?

~~~~~

CheckDifference - Noch nicht eingebaut

DirMemsized - Die Größe des internen Direktorypuffers
kann eingegeben werden.
Ein File benötigt 100 Byte.

Entries Per Menu- Die Anzahl der maximalen Einträge für
das Modules-Pulldown Menü kann
angegeben werden, von 0-63.

Null entspricht dabei der maximal zu-
lässigen Anzahl für den jeweiligen
Screen.

User-Programs

~~~~~

Add UserProgram - Ein Userprogramm kann geladen werden

Add UserDir - Es kann ein ganzes Dir von Userpro-  
grammen geladen werden.

#### Save-Mode:

~~~~~

Uncrunched - Das Module wird ungepackt gespeichert.

PPcrunched - Das Module wird vorm Abspeichern mit
dem Powerpacker-Mode gepackt. Dabei
kann jetzt sogar zwischen Normal,
Protected und SampleMode sowie einer
Kombination dieser entschieden werden.

!!! Powerpacker © Nico Francois, Powerpacker Samplemode © DEFECT !!!

LHCrunched - Das Module wird vorm Abspeichern mit
dem LH-Mode gepackt. Dabei kann zwischen
Normal und SampleMode entschieden
werden.

XPKCrunched - Das Module wird vorm Abspeichern mit
dem eingestellten XPK-Mode gepackt.
Dabei zwischen Normal und Protected
entschieden werden.

CRMCrunched - Das Module wird vorm Abspeichern mit
dem CrunchMania-Mode gepackt. Dabei
kann jetzt sogar zwischen Normal,
Protected und SampleMode sowie einer
Kombination dieser entschieden werden.

~~~~~

Automatic Save - Bei Aktivieren der Module-Save-Funktion  
wird das Module automatisch, ohne  
Filerequester, im angegebenen Direktory

gesichert. ("Set SaveDir")

Overwrite - Es wird vor dem Überschreiben einer Datei beim Module speichern gewarnt.

Safe Save - Bei Sichern einer Datei wird erst eine Sicherheitskopie erstellt und dann das Original, falls vorhanden, gelöscht.

SS Dir - Ein Module wird automatisch in das nach dem Soundsystem benannten Dir gesichert. (Beispielsweise wird beim Speichern eines "Prorunner 1.0" Modules versucht, dieses im Verzeichnis "Prorunner 1.0" abzulegen. Sinnvoll in Verbindung mit "Automatic Save".

SampleMode - Beim Packen wird ein bestimmter Mode verwendet, der es ermöglicht, daß diese Dateien sich besser packen lassen.

Save As PT - Das aktuelle Module wird als Protracker gespeichert, falls dies möglich ist. (zur Zeit nur mit den internen Replayern, außer Customplay, Startrekker und Audiosculpture)

Protect - Das Module wird beim Packen geschützt. Man benötigt zum Entpacken das eingestellte Passwort.

ProtectionBits - Beim Speichern einer Datei werden die Bits einer Datei so gesetzt, daß sie nicht überschrieben werden kann.

~~~~~

Set SaveDir - Mit Hilfe eines Filerequesters kann das Save-Direktory eingestellt werden.

Change XPK-Mode - Der XPK-CrunchMode wird eingestellt.

Change Password - Man kann das Passwort neu festlegen.

Programm-Mode:

~~~~~

No new Module - Bei einem Songende wird kein neues Module geladen und das Abspielen stoppt.

LoadPrev - Wenn Songend oder PlayTime eingeschaltet sind, wird das vorherige Module geladen.

Loadnext - Wenn Songend oder PlayTime eingeschaltet sind, wird das nächste Module geladen.

RandomPlay - Wenn Songend oder PlayTime eingeschaltet sind, wird ein zufällig gewähltes Module geladen.  
Bei Betätigung von Next- oder PrevModule wird ebenfalls ein zufällig ausgewähltes Module abgespielt.  
Dabei werden erst alle Module des geladenen Dirs gespielt, ehe ein Module doppelt geladen wird. Allerdings ist es möglich ein Module über das Pull-Down Menü anzuwählen, ohne daß der interne Zähler sich das merkt, d.h.

es kann per Randomplay auch noch mal gespielt werden.

~~~~~

Songend - Bei Songenderkennung wird das Abspielen unterbrochen und das nächste Module geladen, wenn oben LoadNext, Loadprev oder Randomplay eingestellt sind.

Load Before - Es wird nach Initialisieren des Sound-systems bereits das nächste Module geladen. Damit wird ein recht reibungsloser Übergang zwischen Modulen erreicht, um Megamixe wie "RAF Megamix2", "LSD-Total Chaos" oder "Max Mix" (Flash Prod.) korrekt abspielen zu können.
(r.V)

AutoSubSong - Wenn Songend oder PlayTime eingeschaltet sind, wird der nächste SubSong abgespielt, falls möglich, anstelle sofort das nächste Module zu laden. (r.V.)

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.

Time Mode -

Calc Duration - Hier kann eingestellt werden, ob die Spielzeitberechnung durchgeführt wird oder aus Rechenzeitgründen nicht.

Randomstart - soll beim Start des Eagleplayers ein zufällig ausgewähltes Modul geladen werden ?

Quickstart - Das Module wird sofort nach laden abgespielt.

~~~~~

min PlayTime - Es kann eine min. Spieldauer eingegeben werden, in dezimal.

max PlayTime - Es kann eine max. Spieldauer eingegeben werden, in dezimal.

Warning:

~~~~~

Small Module - Sollte das Module zu kurz sein, wird ein Warnrequester ausgegeben.

Big Module - Sollte das Module zu lang sein, wird ein Warnrequester ausgegeben.

Module Fault - Sollte das Module kaputt sein, wird ein Warnrequester ausgegeben.

PS: Alle Requester werden nach Init-Player bzw. GetModuleInfo ausgegeben.

File-Requester:

~~~~~

No Filerequester- Es wird kein Filerequester benutzt. Alle typischen Eingaben erfolgen über ein

Stringgadget im HauptWindow.  
Req.Library - es wird die Req.Library benutzt  
Reqtools.Library- es wird die Reqtools.Library benutzt  
ASL.Library - es wird die ASL.Library benutzt

#### Iconify:

~~~~~

- Wenn Sie Kick2.0 oder höhere Kickstartversionen besitzen können Sie in den Iconify Modus zwischen LittleWin (Default) , AppIcon oder AppItem auswählen. Benutzen Sie den Appicon Modus, so können im iconifizierten Zustand Files nach dem drag&drop System der Workbench auf das AppIcon gezogen werden, so daß versucht wird, diese Files als Modules abzuspielen.

- Set Voices - Die Stimmen können einzeln an- und ausgeschaltet werden, dabei ändern sich natürlich die Zustände der Gadgets.
- Set Volume - Die Volume kann auf den max. Wert gesetzt werden oder eine Stufe lauter bzw. leiser. Dabei ändern sich natürlich die Zustände der Gadgets.
- Set Balance - Die Balance kann nach Links oder nach Rechts bzw. auf den Mittelstand gesetzt werden. Dabei ändern sich natürlich die Zustände der Gadgets.
- Set Speed - Der Speed kann schneller oder langsamer gesetzt werden. Wenn der derzeitigt benutzt Replayer einen Speed-REGLER zuläßt, kann der Speed zusätzlich auf den Normalwert gebracht werden.

Timing:

~~~~~

Mittels der Funktionen dieses Menüs können Sie die DMA-Wartezeit optimal auf Ihren Rechner konfigurieren. Die DMA-Wartezeit sorgt dafür, daß beim Abspielen keine Aussetzer auftreten. Aufgrund verschiedener Schwierigkeiten der vergangenen Zeit (besonders auf A4000) wurden diese Funktionen mit eingebaut. Sollten Sie hiermit immer noch nichts anfangen können, so lassen Sie am besten die Finger davon, mit den Grundeinstellungen ab V1.53 hat es bisher keine Probleme gegeben. (A500-A4000/40)

- CIA-Timing - Die Abspielgeschwindigkeit wird über einen der internen Timer geregelt, was die stabilste Methode überhaupt ist.
- VBLANK-Timing - Die Abspielgeschwindigkeit wird über die Bildwiederhol-frequenz geregelt. Da diese von Modus zu Modus veränderlich ist (PAL->NTSC->EURO72->SUPER72) und außerdem einige Replayer auf einen Timer angewiesen sind, sollte diese Funktion nur für entsprechend inkompatible Module genutzt werden (MOD.Face another day, MOD.Techdust)

~~~~~

Automatic Wait- DMA-Warteroutine, die auf einer sich am Timer

orientierenden Routine basiert und somit in allen Bildschirmmodi mit der selben Geschwindigkeit arbeitet. Auf dem A4000 läuft es nach unseren Erfahrungen seitdem diese Routine drin ist bestens.

Raster Wait - "übliche" Warteroutine, die eine entsprechende Anzahl von Bildschirmzeilen abwartet. (4..8=A500..A3000) Diese Routine bekommt Probleme unter DBLPAL/NTSC und Productivity, wo eine Bildschirmzeile nur noch halb so lange dauert. Daher läßt sich die Anzahl der Zeilen per Hand einstellen. (siehe unten) Benutzen Sie Werte zwischen 8 und max. etwa 30 für entsprechend beschleunigte Rechner. Auf dem A500 reichen 4 wie oben gesagt aus.

DBF Loop - Der Vollständigkeit halber Diese Routine führt einfach eine Warteschleife aus, deren Länge einstellbar ist.
A500: um 300
Turborechner: Wer will denn sowas benutzen ??

~~~~~

Set Rasterlines - Anzahl der abzuwartenden Bildschirmzeilen über Stringgadget an der Position, wo sonst der Scroller ist, eingeben.

Set Loops - Anzahl der Durchläufe eingeben

Change PScreen - Es erscheint ein StringGadget im MainWindow, in dem Name des neuen PubScreens eingegeben werden kann, auf dem der Eagleplayer dann erscheinen soll.

Toggle Filter - Dreht den Zustand des Filters um. Wenn z.B ein Powerpacker oder CrunchMania den Filter hin- und herschaltet, kann man ihn mit diesem Item umdrehen.

## 1.41 tastaturkürzel

### 4.1.3 Tastaturkürzel

---

Die wichtigsten Funktionen des HauptWindows bzw. des Pull-Down-Menus unterstützen Tastaturabfragen.

Im folgenden werden die Tastaturkombinationen aufgelistet durch die einige Funktionen aufgerufen werden können.

Da wären: ("A" = rechte "AMIGA"-Taste)

\* Project-Menu \*

Load Module   A + l  
Save Module    A + w  
About Module   A + a  
Add Player     A-Taste  
Show Players   P-Taste  
Help           Help-Taste

---

```

Iconify      A + i oder I-Taste
Hide        A + h oder ESC
Quit        A + q

* Command-Menu *
Prev Module  A + b
Prev Song    A + c
Prev Pattern A + y
Replay Song  A + r
Play Song    A + p
Player Faster A + o
Next Pattern A + x
Next Song    A + v
Next Module  A + n
Stop         A + s
Eject Module A + e

* Special-Menu *
Modules-Prefs
  Rescan Dir  R-Taste
User Programs
  Add UserProgram U-Taste
Programm-Mode
  Set Playtime A + z
Set Voices    A + 1-4 (je nach Voice)
Set Volume
  Default Volume A + 5
  Higher Volume A + 6
  Lower Volume A + 7
Set Balance
  Default Balance A + 8
  Balance to Left A + 9
  Balance to Right A + 0
Set Speed
  Default Speed A + #
  Higher Speed A + +
  Slower Speed A + -
Toggle Filter A + f

```

## 1.42 die appwindow-funktion

### 4.1.4 Die AppWindow-Funktion

Der Eagleplayer nutzt in allen Windows die AppWindow-Funktion. Wenn Sie z.B. ein Icon für ein SoundModule haben, ziehen Sie es über eines der Windows des EaglePlayers und lassen es los, das Module wird dann geladen und abgespielt. Sollten Sie durch durch Multiselect mehrere Icons über den Windows loslassen, werden diese in alphabetischer Reihenfolge ins Pull-Down Menü übernommen. Die Icons können aus verschiedenen Verzeichnissen stammen!

Bitte beachten Sie: in der unregistrierten Version kann nicht in Unter-verzeichnisse geschaltet werden. SIE GLAUBEN GAR NICHT WIE DAS NERVEN KANN, BESONDERS, WENN MAN SCHON MAL ANDERS GEARBEITET HAT !

; -)

## 1.43 pubscreens

### 4.1.5 Pubscreens

---

Der Eagleplayer kann ab Version V1.10 nun endlich auch auf Pubscreens geöffnet werden. Die Pubscreenunterstützung betrifft alle Windows der zum Eagleplayer gehörenden Windows bzw. Programme, falls nicht anders angegeben, d.h. alle Userprogramme öffnen sich auf demselben Screen.

Hier nun eine kleine Liste von Publicscreennamen:

```
CygnusEdScreen1 (bei Rendering choices im Environment-Menu
                 make screen public auf on setzen)
BootX
PowerPacker
DOPUS.1
TERM
```

Seit der Version 1.52 existiert das Userprogramm "PubScreenManager", über das man auf einen einfachen Mausklick hin den PubScreen, auf dem der Eagleplayer mit seinen Fenstern erscheinen soll, wechseln kann, einfacher geht's nicht.

## 1.44 font-unterstützung

### 4.1.6 Font-Unterstützung

---

Beim Eagleplayer kann ab V1.10+ angegeben werden, welcher Font benutzt wird. Bitte beachten Sie, daß nur 8\*8 ungepackte, nicht proportionale Fonts benutzt werden dürfen.

Diese Funktion wird wahrscheinlich nicht mehr erweitert werden, da wir zur Zeit an einer viel weitreichenderen Idee arbeiten.

## 1.45 interne depacker

### 4.1.7 Interne DePacker

---

Der EaglePlayer in der vorliegenden Version kann Datenfiles entpacken, wenn diese mit folgenden Crunchern gepackt sind:

- CrunchMania ;nur mit Library, aber alle Modi
  - Powerpacker (falls vorhanden, wird die Library benutzt)
  - LH (PMC vom Powerplayer- doch was ist am Power-  
player POWER ???)
  - Fimp
  - Pack (Soundtracker-Songpacker)
  - XPK (bei den meisten Packern [SQSH,NUKE] wird  
während des Ladens entpackt !!!)
-

- StoneCracker V4.0  
weitere Packer bietet die ebenfalls unterstützte XFDMaster.library, bitte schauen Sie in deren Dokumentation nach, welche Formate diese entpacken kann

Bei Fimp wird nur ein Memoryblock allokiert, in den das Module geladen wird und auch entpackt wird. Diese Methode funktioniert wunderbar, aber eben nur bei Fimp. Bei den anderen Packern muß erst ein Memoryblock zum Laden und einer zum Entpacken besorgt werden. Diese Methode verbraucht natürlich viel Speicher, und das ist nicht gut. Nach dem Entpacken wird der Speicher des geladene & gepackten Files wieder freigegeben.

Der EaglePlayer unterstützt die XPKMaster.Library, wenn die entsprechenden Libs: auf der Diskette sind. Sollte eine Datei ein Passwort benötigen, kann dieses über ein StringGadget im Hauptwindow eingegeben werden.

Für das Entpacken von PCM-Files, also mit der LH.Library gepackten Files, wird diese Library benötigt, ist diese nicht vorhanden, kann das Module nicht entpackt werden.

Bei Pack handelt es sich um den SongPacker der ganzen Tracker (NoiseTracker, Protracker, Startrekker). Diese Files werden automatisch beim Laden entpackt, d.h. jeder externe Player könnte auch gepackte Songs seines Formats laden. Powerpacker-, Fimp-, Pack- und StoneCracker4.0-Files können ohne Hilfsmittel entpackt werden. Die entsprechenden Decrunchroutinen sind im EaglePlayer enthalten. Wird die Powerpacker.Library gefunden, wird diese zum Entpacken von PP-Dateien genutzt.

Der EaglePlayer kann weiterhin auch executable Files entpacken, wenn diese mit ImploderV4.0 (NormalMode ohne Library) gepackt sind. Diese Funktion wurde implementiert, weil das Programm zunehmend länger und länger wurde, vor allem die Player und UserProgramme. Um diese einem durchschnitts-User auch zugänglich zu machen, kann man sie eben alle packen. CustomModule können ebenfalls mit Imploder gepackt vorliegen. Ron-Klaren-Modules müssen ungepackt sein ! In späteren Versionen werden auf jeden Fall mehr File-Entpacker unterstützt, tut mir leid, das es gerade Imploder erwischt hat, aber besser als gar nicht packen, oder ?

Ab Version V1.52 wird die XFDMaster.Library unterstützt, d.h Sie können auch PowerPacker-Executables laden, wenn Sie den entsprechenden Slave besitzen. Sie können also auch Player und Userprogramme mit dem Powerpacker oder CrunchMania oder StoneCracker packen, soweit dies durch die jeweilige Version der XFDMaster.library unterstützt wird.

Die XFDMaster.library können Sie bei

Georg Hörmann  
Martinswinkelstraße 16c  
82467 Garmisch-Partenkirchen

oder auf gutsortierten PD-Serien bekommen.

## 1.46 der lha-extraktor

### 4.1.8 Der LHA-Extraktor"

---

Ab dem Eagleplayer V1.50 besitzt der Eagleplayer die Möglichkeit LHA-Archive

---

zu entpacken. Dabei wird zur Zeit nur der erste Eintrag entpackt und geladen. Als Hilfsprogramm wird natürlich LHA benutzt, welches sich im C-Verzeichnis des Boot-Laufwerkes befinden muß.

Das Module wird entpackt nach T: und dann von dort geladen.

In Zukunft wird bei Mehrfilearchiven der Inhalt des Archives in das Pull-Down Menü übernommen.

## 1.47 commodity-funktionen

### 4.1.9 Commodity-Funktionen

-----

Der Eagleplayer besitzt seit der Version V1.10 einen Commodity-Port. Dieser ist nur ab Kick2.0 aufwärts verfügbar.

Der Eagleplayer kann über selbstdefinierte Hotkeys aufgerufen werden. Voreingestellt ist die Tastenkombination "CONTROL ALT E". Bei Aktivierung des Hotkey wird entweder das Hauptwindow mit Userprogrammen, Konfigurationsfenster etc. geöffnet und in den Vordergrund gebracht oder geschlossen.

In Zusammenarbeit mit Commodities-Exchange können weitere Commandos, wie u.a. Quit realisiert werden.

Die Optionen "CX\_Popup", "CX\_Popkey" und "CX\_Priority" müssen derzeit noch im Konfigurationsfile angegeben werden.

Voreingestellt ist auch, daß der Eagleplayer beim Schließen des Hauptfensters im Speicher verbleibt und erst durch die Menü-Option

Quit

aus dem Speicher entfernt wird. (Diese Funktion läßt sich durch ↔

die Option

{"Quit Eagle" Link "Steuerung über die Pulldown-Menüs"} ersetzen.)

## 1.48 öffnung und funktion des player-fensters

### 4.2.1 Öffnung und Funktion des Player-Fensters

-----

Das Playerwindow können Sie über die Menü-Funktion "Show Players" öffnen und schließen. Bei einem Iconify wird das Window ebenfalls geschlossen. Sollten Sie LittleWin eingestellt haben, können Sie es im Iconify trotzdem öffnen. Über dieses Window erhalten Sie Informationen über die einzelnen Player, wie z.B Creator, VersionsNr des Players, unterstützte Möglichkeiten, Anzahl der Player usw.

## 1.49 steuerung über die schalter und erklärung der funktionen

### 4.2.2 Steuerung über die Schalter und Erklärung der Funktionen

-----

---

In dem eingerahmten Feld links können Sie die Player anwählen. Haben Sie einen Player selektiert, können Sie folgendes tun:

- lesen, was er alles unterstützt, z.B. ModuleInfo, Save, Packable
- ihn löschen (außer interne Player)
- ihn deaktivieren/aktivieren, d.h Sie können ihn ausschalten ohne ihn zu löschen.

Über das Playerwindow können die folgenden Aktionen getätigt werden:

Add-Player - Laden eines neuen Players. Falls der Player bereits vorhanden ist, wird er nur geladen, wenn die Versionsnummer gleich oder größer ist.

Del-Player - Löscht den selektierten Player

Config - Ruft eine interne Routine des angewählten Players auf, die z.B ein Intrumenten-Directory für diesen Player festlegt (falls vorhanden)

Add-PlayerDir - Sie haben die Möglichkeit ein ganzes Directory mit Playern einzuladen.

Load - Ruft eine interne Routine des angewählten Players auf, die eine Konfigurations-Datei laden soll (falls vorhanden)

Save - Ruft eine interne Routine des angewählten Players auf, die eine Konfigurations-Datei save soll (falls vorhanden)

Enabled Yes/No - Hiermit kann der selektierte Player erlaubt oder verboten werden. Der Player wird nicht aus dem Speicher entfernt !

## 1.50 tastaturkürzel im playerwindow

### 4.2.3 Tastaturkürzel im Playerwindow

-----

Die wichtigsten Funktionen des Playwindows unterstützen Tastaturabfragen. Im folgenden werden die Tastaturkombinationen aufgelistet durch die einige Funktionen aufgerufen werden können.

Da wären:

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

Weiterhin kann per Cursortasten das "ListViewGadget" bewegt werden.

## 1.51 interne player

### 4.2.4 Interne Player

-----

Der EaglePlayer besitzt in der vorliegenden Form 31 interne Player. Interne Player können nicht gelöscht werden, aber enabled/disabled. Bei der Erkennungs Routine werden als erstes die externen Player der Reihe nach durchgecheckt und dann die internen der Reihe nach. Bei den meisten internen Player handelt es sich um Protracker-Packer, die in Protracker zurückkonvertiert werden. Dabei kann man davon ausgehen, das die beste Playroutine für diese Format benutzt wird, nämlich die, auf der die meisten Module erstellt wurden. Es können alle PT-Packer als Protracker gespeichert werden. (reg. Version) Wenn Sie diese Modulformate als Protracker haben wollen, können Sie auch den "Noiseconverter" zu benutzen.

Folgende Player besitzt der Eagleplayer intern:

- AC1D-DC1A-Packer - Format aus dem Anarchy-Demo "In the Kitchen"
  - Audio Sculpture - Siehe Startrekker
  - CustomPlay - Hier handelt es sich um ein Interface, um z.B. Songs, die auf absoluten Adressen liegen oder die keinen Soundsystem angehören abzuspielen. Dafür muß der Song entsprechend seinen Möglichkeiten angepaßt und ein Playerheader, wie bei einen externen Player, vorgesetzt werden.
  - Digital Illusions- SoundPacker, vor allem von der gleichnamigen Firma verwendet.
  - Eureka-Packer - Soundpacker, der in verschiedenen Demos zu finden ist
  - Game Music Creator- Ein Soundformat, das seine Verwandtschaft mit dem alten Soundtracker kaum leugnen kann
  - Kefrens Sound Machine - Der erste Soundtrackerpacker überhaupt, findet sich nur in uralten Kefrens Demos. Geschrieben von Razmo/Kefrens.
  - Kript - Ein Protrackercodierer, bei dem die Pattern lediglich um einen variablen Langwortwert addiert werden. Dieser Player ist grundsätzlich disabled. Bei Benutzung muß der richtige Code über die Config eingegeben werden. Er erkennt übrigens alle ProTrackerMods und sollte daher mit Vorsicht genossen werden.
  - Noisepacker 2.x - Zwei Packer, die von der Gruppe "Phenomena" entwickelt worden sind. Da sie normalerweise nur Noisetracker unterstützen, sind sie für neueste Module kaum geeignet
  - Noiserunner - Ein ProTrackerPacker von Chaos of Sanity, der auf Zeit getrimmt ist und nicht auf die Packrate (und die unterstützen Kommandos).
  - Noisetracker Compressed Ein gepacktes Musikformat, das von einem deutschen Noisetrackerduplikat benutzt wird.
  - Old SoundTracker - Das Module wird in ProTracker konvertiert. Der Player kann trotzdem Songs laden. Die Playroutine wurde entfernt, weil einfach zu fehlerhaft und inkompatibel war.
  - Powermusic - Ein Protrackerformat, bei dem die Samples für eine bessere Packrate umcodiert wurden. (nach dem selben Schema, wie bei der lh.library)
  - Promizer 2.0 - Die Version2.0 des Promizers wurde integriert, nachdem es auf Turborechnern diverse Probleme mit der Playroutine der Module gab.
  - ProPacker 1.0 - Ein PT-Packer von Azatoth of Phenomena.
-

Achtung !!

Alle zu kurzen Mods werden nicht erkannt, weil die Patterns am Ende des Mods liegen und vollständig sein müssen.

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

ProRunner 1.0 - Ein Packer von Cosmos of Sanity. Hier handelt es sich lediglich um einen Kodierer.

ProRunner 2.0 - Ein PT-Packer von Cosmos. Packt recht gut, obwohl der Packer einige Bugs hat.

ProTracker - Der am meisten genutzte Replayer überhaupt !!!

Es wird keine Unterscheidung zwischen Noise-, Sound- und Protracker gemacht. Wenn Sie Module besitzen, die in diesem Replayer nicht funktionieren, senden Sie uns diese bitte zu.

Es können Songs geladen werden. Der Instrumenten Pfad kann per Konfig eingestellt werden. Dieser Pfad gilt für alle internen Samplepfade und wird mit der Gesamtkonfiguration des Eagleplayers abgespeichert !

Hier kommt eine original Protracker V1.0b Playroutine zum Arbeiten, in der mehrere Bugs behoben wurden und in die alle Kommandos des Protrackers 3.10 integriert wurden !

Auf eine Unterscheidung zu Noisetracker und Soundtracker wurde verzichtet, da so etwas eh nicht 100%ig möglich ist und die Unterschiede im Kommandorargument nicht erkannt werden können. Es ist grundfalsch anzunehmen, daß ein Modul, in dem als einziges ein Kommando vorkommt, das im Noisetracker anders ist, als im Protracker, auch dieses Modul mit dem Noisetracker erstellt worden ist !!! Delitracker Sucks !

Außerdem spart's den Speicher zweier

Playroutinen .....

Startrekker - Als Replayer wird eine Audio Sculpture (kommerzieller Nachfolger vom Startrekker) Routine benutzt. Die Unterscheidung zwischen Startrekker und Audio Sculpture erfolgt im NT-File. Falls das NT-File nicht geladen wird, wird das Module als Startrekker erkannt. Mit der eingebauten Routine können Sie Startrekker Normalmodule, Startrekker AM-Module, AudioSculpture Normal-Module und AudioSculpture AS-Module abspielen, d.h. das NT/AS File ist nicht unbedingt erforderlich, außer es handelt sich um einen AM-Sound. Ein Song-Loading ist ebenfalls möglich.

The Player 4.x - Endlich ein ordentlicher Player für das Mistding. Es werden Die Formate mit den Kennungen "P40A", "P40B" sowie "P41A" geladen. Ein sehr effizienter ProTracker-Packer. Packer von Jarno Paananen.

The Player 5.0a/6.0a - Nachfolger des P4.x, noch bessere Packrate, noch schwierigerer Modulaufbau und noch schwerer anzupassende Playroutine. Daher wurde für dieses Format auch kein eigener Player angepaßt, sondern das Modul



wird zurückkonvertiert. ACHTUNG: BEIM ABSPEICHERN WIRD IM P60-FORMAT GESAVED !  
 Packer von Jarno Paananen.

TrackerPacker2 - Ein von der Gruppe "Mexx" häufig benutztes Musikformat. Der Player erkennt 2 verschiedene Typen, einen mit Vorblock "MEXX\_TP2"+Songnamen, 28 Bytes lang, bzw. ohne.  
 Lief früher unter dem Namen "MEXXPacker".

TrackerPacker3 - Nachfolger des Trackerpacker2 (logisch eigentlich), selbe Packrate, nur anderer Patternaufbau. Nach "P60A" zweite Wahl in Geschwindigkeit und Packrate, bei Benutzung von Patternbreak in vollem Umfang der derzeit beste Player. [außer Protracker :) ]  
 Auch die Playroutine ist brauchbar. Anscheinend hat sich das Meckern doch gelohnt :)

Unic-Tracker - Ein Packer von Laxity of Kefrens.

Unic-Tracker2 - Dito

WantonPacker - Ein Packer von Wanton of Bloodsuckers. Hier handelt es sich lediglich um einen Kodierer.

XANN-Packer - In irgendeinem uralten Demo fanden sich mal ein paar Module. Da diese Routine nicht sehr lang ist, wurde sie integriert.

## 1.52 tips zur konfig

Tips zur Konfiguration:

- Der Eagleplayer ist in der Lage, das aktuelle Verzeichnis mit abzuspeichern, sogar das aktuelle Modul merkt er sich. Sollten Sie dies nicht wollen, so laden Sie einfach ein Modul aus dem aktuellen Verzeichnis und werfen Sie es raus (
 

```

      ^
      ), bevor sie die Einstellungen speichern.
      
```

 Eine weitere Möglichkeit ist die
 

```

      Random Start
      
```

 Funktion, die es ermöglicht, beim Start des Eagleplayers ein zufällig gewähltes Modul aus dem aktuellen Verzeichnis zu laden.
- Falls beim Einlesen eines Verzeichnisses die Meldung "Error! Dir Memory full !" erscheint, so reicht die Größe des Directorypuffers nicht aus. Der Menüpunkt "Dirmemsize" aktiviert ein Stringgadget im Hauptfenster, durch das sie die Größe in Bytes angeben können. 10000 Bytes reichen etwa für 80 Einträge, so daß 80KB für die allermeisten Anwendungen ausreichen sollten. (meist wird es dann schon eng mit der Anzeige auf dem Screen, auf einem Euro72 Productivity Screen ist Platz für etwas über 700 Module.)
- Wer es nicht mag, daß ständig ein anderes Modul geladen wird, sollte die "Songend" Funktion deaktivieren.
- Der Eagleplayer unterstützt 3 verschiedene Filerequester Libraries, wem also die reqtools.library mißfällt, der kann auch auf die req.library oder die ASL.lib als Systemstandard zurückgreifen.

- Es wird empfohlen, nur die wirklich benutzten Userprogramme mitzuladen, da die anderen nur unnötig Speicher verbrauchen. Man kann die unbenutzten einfach in einem in "EP-Userprograms" angelegten Unterverzeichnis ablegen, so daß diese beim Start nicht mehr mitgeladen werden, aber weiterhin zur Verfügung stehen.  
Für registrierte Benutzer bietet sich hier die  
    Userprogram Batch  
    Funktion  
an, durch die ein Userprogramm erst dann geladen wird, wenn es aktiviert wird. Diese Funktion läßt sich sogar dahin gehend erweitern, daß ein geschlossenes Userprogramm aus dem Speicher entfernt wird und erst bei Reaktivierung wieder geladen wird.
  - Durch die bei der Installation mitgelieferte Konfigurationsdatei wird die  
    Playerbatch  
    Funktion aktiviert, welche die externen Abspieler erst dann lädt, wenn diese benötigt werden. Dieser Prozess erfolgt durch die Namens-  
kennung einer Datei. (Vorteil: Die Datei kann gepackt werden und der Eagleplayer wird schneller geladen, da die Ladezeit für über 70 externe Dateien entfällt)  
Das mögliche Problem besteht darin, daß selten komplett gleiche Kennungen für Module verwendet werden. Da die Stapeldatei als ASCII-File geladen wird, können Sie diese nach Belieben mit einem Editor Ihrer Wahl auf die eigenen Bedürfnisse angepaßt werden.
  - Öffnen Sie nicht zuviele Scopes auf einmal ! Da hier für höchstmögliche Kompatibilität insbesondere beim Betrieb mit Grafikkarten der Blitter zum kopieren verwendet wird, ist die freie Rechenzeit schnell erschöpft. Dank C= ist der Blitter bei AGA Maschinen übrigens immer noch gleich langsam wie beim A1000 ! Daher kommt es auch auf dem A4000 schnell dazu, daß nichts mehr geht. Machen Sie die Scopes auch nicht zu groß, im Kapitel zu den Userprogrammen steht ein Rechenbeispiel, was so an Daten pro Sekunde geschauelt werden muß.
    - Wundern Sie sich nicht, wenn bei extremer Systemauslastung die ←  
    Scopes  
stehen bleiben, sie laufen auf niedriger Taskpriorität, um andere "wichtige" Programme nicht auszubremsen.
  - Wer registrierter User ist und den EP im Hintergrund laufen läßt [wie ich, Buggs :) ], kann sich ja mal die "Load Before" Funktion anschauen, die es ermöglicht, daß sofort nach dem Laden eines Moduls das je nach Einstellung nächste, vorherige oder ein zufälliges Modul geladen wird. Dies hat den Vorteil, daß praktisch übergangslos zum nächsten Modul gewechselt werden kann und man den Anfang des letzten Moduls während des Ladens nicht noch einmal hören muß.
  - Es soll Leute geben, die nicht gern mit dem Menü arbeiten. Für diese gibt es den "Dirlistviewer", durch den man das aktuelle Verzeichnis zusätzlich zum Menü in einem größenveränderlichen Fenster sehen und durch Doppelklick Module laden kann.
  - Um den Eagleplayer komfortabel auf einen andren Schirm wechseln zu lassen, kann man sich ja mal mit dem "Public Screen Manager" befassen.
-

## 1.53 einleitung zu den userprogrammen

### 4.3.1 Einleitung zu den Userprogrammen

---

Die Userprogramme stellen eine Neuerung auf dem Markt überhaupt dar. Sie sind extern zu ladende, eigenständige Programme, die asynchron (d.h. als eigene Tasks) arbeiten. Ursprünglich als reine Analyzerprogramme vorgesehen, kamen jetzt noch Möglichkeiten hinzu, durch die der Eagleplayer nicht nur der steuernde, sondern nunmehr der gesteuerte Teil sein kann. Das heißt, daß die Userprogramme Zugriff auf alle wichtigen Daten des Eagleplayers haben und Kommandos an diesen senden können. Dies eröffnet ganz neue Welten. So ist es kein Problem, eine andere Oberfläche für den Eagleplayer zu schaffen, z.B. mit eigenem Screen im CD-Player-Look. Dazu wäre es nur notwendig, die Oberfläche mit Eingabehandlung zu schreiben und Eingaben an den Eagleplayer weiterzureichen, genauso leicht kann man mal fix die Oberfläche wechseln, statt dessen beispielsweise den Intiutracker "Nostalgielook" verwenden (ist schon in Planung). Der Eagleplayer verwaltet zur Zeit 32 Userprogramme, die beliebig ein- und ausgeschaltet werden können. Auch abspeicherbare Voreinstellungen der Userprogramme mit Fensterposition, Arbeitsmodus (z.B. beim Analyzer) und Information ob das Fenster offen oder geschlossen ist, sind vorgesehen und in der registrierten Version nutzbar. Wer's braucht, kann Userprogramme auch mehr als einmal starten, entweder unter einem anderen Namen ins Verzeichnis 'EP-Userprograms' kopieren oder aber über die Menüfunktion "Add Userprogram" zusätzlich einbinden.

Ein Informationswindow zu den Userprogrammen wird es in Zukunft auch geben.

## 1.54 steuerung der user-programme

### 4.3.2 Steuerung der User-Programme

---

Die Userprogramme können durch das Eagleplayermenü Special/Userprograms ein- und ausgeschaltet werden, indem man den entsprechenden Menüpunkt anwählt. Ist der Haken gesetzt, heißt dies, daß das jeweilige Userprogramm sich im aktiven Zustand befindet.

Wird bei einem der Userprogramme direkt das Close-Gadget (oder der Hide-Menüpunkt) angewählt, so schließt das Userprogramm das Fenster, wird aber nicht aus dem Speicher entfernt, sondern ist durch den Eagleplayer in der oben beschriebenen Form wieder in den Betriebszustand versetzbar.

Wird im Pull-Down-Menu des Userprograms der Punkt "Quit" ausgewählt, wird das Userprogramm sofort aus dem Speicher entfernt.

Bitte beachten Sie: Die Userprogramme werden über die Bildschirmfrequenz gesteuert und brauchen somit bei 50 Hz Bildschirmfrequenz logischerweise 25% Rechenzeit weniger als bei 78Hz Super72 ! Alle aufgeführten Werte in Bezug auf die Geschwindigkeit beziehen sich auf Messungen im 50 Hz DBLPal-Modus.

Jedes Userprogramm hat seine Eigenheiten, die in vollem Umfang zu beschreiben, ein wenig sehr aufwendig wäre.

Bei einigen, besonders rechenzeitaufwendigen Userprogrammen gibt es eine "Slow Mode" Funktion, bei der die Anzeigerate halbiert wird und somit auch deutlich weniger Rechenzeit beansprucht wird.

Folgende bisher fertiggestellte Userprogramme können auch unter Kickstart

---

1.2/1.3 genutzt werden: (mit '\*' gekennzeichnete UPS sind nur mit der registrierten Version zu erhalten)

- Time
- Levelgraph
- Noiseseagle
- \* Levelgraph2
- \* Levelmeter
- \* Analyzer2 (Small Analyzer)
  
- Moduleinfo | diese drei benötigen unter Kick1.3 zusätzlich
- Space-Scope+ | die Gadtools13.library, die nicht dem Eagleplayer-
- Patternsroll\_Kick1.3 | paket beiliegt, da Copyright unbekannt

Die anderen, unten genannten UPS sind nur unter Kick 2.0 aufwärts zu nutzen:

- Analyzer
- DirListviwer
- Quadrscope
- Hifi System V 1 (heißt Kenwood und sieht aus wie Technics)
- M.E.L.G.U. (Mega Eagleplayer List Generating UNIT)
- Patternsroll
- Patternsroll++
- PubScreenManager
- \* Stereoscope
- \* Monoscope
- \* Voicemeter

Ab V1.52 gibt es auch Userprogramme, die OS3.0 verlangen:

- Levelmeter AGA
- Stereoscope AGA

## 1.55 zu den einzelnen bisher verfügbaren userprogrammen

### 4.3.3 Zu den einzelnen bisher verfügbaren Userprogrammen

---

#### Analyzer

- Quasi Spectrum Analyzer, der nach dem üblichen Prinzip arbeitet und nach der Samplerate beim Abspielen agiert. (genauso wie beim Protracker etc.) Auch wenn es einige Leute nicht glauben wollen, der Analyzer besitzt wirklich 21 Modi, wobei zu sagen ist, daß einige sich nur um Kleinigkeiten unterscheiden. Wen Scroller und/oder Bild stören, kann dies über das Pulldownmenü abschalten, in der Konfiguration wird das ganze berücksichtigt, sowie der aktuelle Modus mit abgespeichert.

#### Quadrscope/Stereoscope/Monoscope

- Anzeige der aktuellen abgespielten Samples, entweder getrennt (Quadrscope) oder additiv zusammengefaßt. (Monoscope, Stereoscope) Diese Anzeige erfolgt in Echtzeit, daß heißt, daß auf einem angeschlossenen Oszilloskop dasselbe sichtbar sein müßte.

#### Time

- Anzeige der Abspiel-, sowie der Systemzeit. Das Programm wurde so umgeschrieben, daß der Prozessorzeitaufwand drastisch gesenkt
-

werden konnte.

#### Levelmeter/Levelgraph/Levelgraph2

- "echte" Aussteuerungsanzeigen, die sich ihre Werte aus den Samples holen und somit die wirklich vorhandene Lautstärke anzeigen.

#### Levelmeter AGA

- siehe oben, nur mit besserem Look. Es sollte für bestmögliches Aussehen ein Screen mit mindestens 2 freien Pens (z.B. 16 Farben Workbench) bereitstehen, wird aber nicht zum Betrieb vorausgesetzt. Außerdem zeigt das Programm die aktuelle Spielzeit mit an.

#### Stereoscope AGA

- siehe oben, nur im perfekten "Oszilloskop"-Look. Benötigt eine Menge Rechenzeit durch die großen Datenmengen, die ständig bewegt werden müssen. (weit über 400 KByte pro Sekunde an anfallenden Daten im Normalmodus bei 16 Farben) Kann ein wenig Rechenzeitfreundlicher durch "Slow Mode" geschaltet werden.

BENÖTIGT MINDESTENS EINEN SCREEN MIT 2 FREIEN PENS. (16 FARBEN WORKBENCH)

#### Patternsroll

- Sollte eigentlich jedem, der schon mal einen Tracker in der Hand hatte, bekannt sein. Als kleines Bonbon wurde das Teil gleich so geschrieben, daß es bis 8 Stimmen vorgesehen wurde (Bsp. Oktalyzer)  
Verbraucht auf meinem 50Mhz 68030 Rechner im 8 Stimmen Modus durchschnittlich unter 20% Rechenzeit, bei 4 Stimmen unter 10% !  
Wer heute noch mit einem A500 im Sand spielt, sollte sich um einen echten Computer kümmern !
- Ab Version 1.53 wurde das Teil komplett überarbeitet und wir bieten die bisherige Version, die auch unter Kickstart1.3 läuft, sowie zwei neue Programme, die sich in der Größe beliebig verändern lassen, sowie bis zu 16 Stimmen (Patternsroll++ bis zu 32 mit halb so breitem Text) unterstützen.
- Bei selber Größe wurde die Geschwindigkeit noch einmal um 30% gesteigert
- die automatische Größenveränderung läßt sich bei den neuen Patternsrollern abschalten

#### Voicemeter

- Quasi Aussteuerungsanzeige, die auf die von der Replayroutine eingestellten Lautstärkewerte reagiert. Also wieder derselbe Gag wie bei den schon vorher auf dem Markt befindlichen Programmen. (Tracker) Verschiedene Betriebsmodi möglich, erreichbar über die Taste "1" !! Eine Anwahl mittels Gadgets ist in Vorbereitung.

#### Analyzer2

- Kleine, unter 1.3 nutzbare Version des Analyzers, noch nicht ganz fertig, die endgültige Version folgt in Kürze (irgendwann Anfang '94 -> oder auch nicht...)

#### Noiseeagle

- Wer die Eagleplayeroberfläche überhaupt nicht mag (Hi CPT. BIFAT of TEK) kann dieses Userprogramm starten, das genau wie der gute alte Noiseplayer ein File anfordert und dieses abspielt. Der Unterschied zum Original besteht darin, daß ein wenig mehr Replays unterstützt werden und ich auf die hardwarehackenden Maus/Tastaturabfragen verzichtet habe.

#### Hifi System V1

- Mir gefiel irgendwie mal ein Bild eines HiFi-Turms von Platin, das ich nach Rücksprache mit dem Creator an meine Vorstellungen anpaßte und nun als erste vollwertige neue Eagleplayeroberfläche

präsentiere. Aufgrund dessen, daß unsere Grafiker Protest anmeldeten, stellte ich die Arbeiten ein und bringe dieses Programm als Beigabe zur unregistrierten Version heraus.

Es dient einmal zur direkten Steuerung des Eagleplayers (ich glaube kaum, daß die Funktionen lang und breit erklärt werden müssen) und in zweiter Hinsicht sind die Analyzer integriert worden, in einer etwas farbigeren Form.

Außerdem ist das Levelmeter (die horizontale, LED-förmige Aussteuerungsanzeige) samplegesteuert, daß heißt, daß die Anzeige sich wirklich analog zur tatsächlichen Lautstärke verhält, da die Werte aus den Samples selbst berechnet werden.

M.E.L.G.U

eigene Dokumentation.

Space-Scope

- Nett anzusehender Sampleanzeiger, den ich zum ersten Mal auf dem Atari ST sah. Braucht aufgrund hoher Blitterbeanspruchung und ziemlich vielen Punkten pro Rasterdurchlauf eine Menge Rechenzeit. (besonders auf unbeschleunigten Maschinen, aber wer hat sowas noch)

Wem das Teil zuviel Rechenzeit frißt, der kann die 800 Punkte, die pro Durchlauf dargestellt werden, auf 400 oder 120 verringern. Außerdem kann durch den "Slow Mode" die Rechenzeit durch eine geringere Refreshrate auf etwa die Hälfte gesenkt werden.

(800 Punkte in 1/50sek = 40000 Punkte pro Sekunde P L U S  
200 KByte Daten pro Sekunde, die geschaufelt werden müssen,  
um den Puffer systemkonform ins Fenster darzustellen,  
d.h. ohne alles c.a. 16% der Blitterperformance+20% Rechenzeit  
auf einem 50MHz 030 Computer)

Einige Module, bei denen es besonders gut aussieht:

```
"Beast II ingame"
"Mod.load master loading" by Rebels,
"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
"Mod.Hearts of Darkness" by Paradise
```

```
.
.
.
.
.
.
```

Space-Scope+

- wie oben, nur größenveränderliches Fenster

DirListViewer

- Anzeige des aktuellen Verzeichnisses in einem größenveränderlichen Fenster
- Durch einen Doppelklick wird das selektierte Modul geladen
- Tastaturunterstützung über das rechte Keypad

## 1.56 keine programmierrichtlinien für userprogramme

### 4.3.4 Warum haben wir noch keine Programmierrichtlinien herausgegeben ?

---

Das große Problem besteht darin, daß die Userprogramme schon in der Version 1.0 des Eagleplayers sehr komplex aufgebaut waren und ab der Version 1.10+ ein Haufen neuer Funktionen dazukam. Es sei dazu nur die Möglichkeit erwähnt, externe, vom User ausgewählte Oberflächen zu entwickeln. Unser zweites großes Problem ist das Zeitproblem, mal abgesehen davon, daß dann sowieso keiner durchsehen würde, selbst wenn wir uns große Mühe in der Doc geben würden. ( Wir sehen ja selbst bald keine Sonne mehr :)--== )  
 Wer unbedingt ein Userprogramm programmieren will, der sollte sich bei uns melden.

## 1.57 einbindung des eagleplayers in andere programme

### 4.3.5. Einbindung des Eagleplayers in andere Programme

---

Der Eagleplayer bietet die Möglichkeit über den eigenen UserprogrammPort eine Kommunikation mit völlig eigenständigen Programmen zu vollziehen. Es werden normale Exec\_Messages ausgetauscht, mit der Messagelänge "UM\_Sizeof" und dem Eagleplayerinternen Typ "USM\_ExternalPrg".

Ein Beispiel für diese Möglichkeiten bietet der als Bonusprogramm beigelete Noiseconverter. Er gibt an den Eagleplayer das Kommando ein bestimmtes Module aus dem Speicher zu laden und abzuspielen und umgekehrt.

Es ist auch möglich, ein Extloading zu emulieren, so daß ein Abspielen aus einem Ripper gar kein Problem ist. Alles Klar, Turbo ???

Hier nun ein kleiner Sourcetext zur Demonstration. Er ist voll lauffähig auf ASM-One und Devpac. Wer also mal ein bischen testen will, braucht ihn nur aus der Dok zu pflücken.

(Schauen Sie auch ruhig in die englische Doc. Dort wurde ein anderer Sourcetext mit eingefügt)

```
*****
*      PlayMem, Message an Eagleplayer senden      *
*      -----*
*
* Dieses Sourcecode ist ein Beispiel, um Speicher im Eagleplayer abzu- *
* spielen. Es sollte dabei logischerweise um ein heiles Module handeln. *
* Es werden dabei die Startadressen der jeweiligen Files sowie deren *
* Größe angegeben. Probleme gibt es bei Replayern, die Check1 benutzen. *
* Diese Funktion ist hervorragend für Ripper (Hi Turbo) geeignet. *
*
* Kick2.0+ ist Voraussetzung !!! *
* Geschrieben mit ASM-One/Devpac.      © DEFECT 1993 *
*****
  inidir "include:"
  include "misc/eagleplayer.i"
  include "exec/exec_lib.i"
```

---

```

*----- Eagleplayer-TestProgramm -----*
Prg_Start:  bsr.s PlayEagle
            moveq #0,d0
            rts

*----- Eagleplayer gestartet ?? -----*
PlayEagle:  move.l 4.w,a6
            lea EP_Portname(pc),a1
            jsr _LVOFindport(a6) ;Eagleplayers Port finden
            tst.l d0
            beq.w .NoEagleplayer ;Cancel->Ende des Programms
            move.l d0,EP_Port

*----- Schnell mal einen Port Createn -----*
            jsr _LVOCreateMsgPort(a6)
            move.l d0,TP_Port
            suba.l a1,a1
            jsr _LVOFindTask(a6) ;Task finden
            move.l d0,d6

*----- Message vorbereiten -----*
            lea MyMessage(pc),a1
            clr.l (a1) ;LN_Succ löschen
            clr.l 4(a1) ;LN_Pred löschen
            move.w #$800,8(a1) ;LN_Type & Pri setzen
            move.l #PlayMem.MSG,10(a1) ;LN_Name setzen
            move.l TP_Port(pc),14(a1) ;Portadresse,an die
            ;zurückgesendet wird
            move.w #UM_sizeof-20,18(a1) ;Length festlegen

            move.w #-1,UM_UserNr(a1)
            move.l #USM_Externalprg,UM_Type(a1)
            move.l d7,UM_Userport(a1)
            move.l d6,UM_TaskAdr(a1)
            move.l #-1,UM_Signal(a1)
            move.w #USClass_Command,UM_class(a1)
            move.l #0,UM_Userwindow(a1)
            move.l #UCM_PlayMem,UM_Command(a1)

*----- Argumentstruktur initialisieren -----*
            moveq #EPT_String+30,d0
            movem.l d1-a6,-(Sp)
            move.l 4.w,a6
            move.l #$10001,d1
            jsr _LVOAllocmem(A6)
            movem.l (sp)+,d1-a6
            tst.l d0
            beq .DelMsgPort

*--- Hauptfile in spezieller Struktur übergeben ---*
            move.l d0,a2
            move.l a2,UM_ArgString(a1)

```



```

    move.l #EPT_String+30,EPT_Stringsize(a2)
    clr.l EPT_Next(a2)
    move.l DatenPuffer(pc),EPT_Result1(a2) ;Arg1 (Adr)
    move.l DatenPuffer+4(pc),EPT_Result2(a2) ;Arg2 (Size)

*----- den Filenamen übergeben -----*
* Auf jeden fall sollte eine Kennung übergeben *
* werden, durch die erkannt wird, welcher Replayer *
* genutzt wird. Der entsprechende Player wird dann *
* geladen !! *
*****
    move.l a2,a4
    moveq #30,d1
    lea.l My_Filename(pc),a3
    lea EPT_String(a2),a2
.Copyfilename: move.b (a3)+,(a2)+
               dbeq d1,.Copyfilename

    clr.l UM_Result(a1)
    move.l a1,a2

*-- Nun die Argumentenstrukturen für EXTLoding füllen --*
    lea DatenPuffer(pc),a3
    move.l DatenPufferAnz(pc),d7
    subq.l #2,d7
    blo.s .NoExtLoading

*---- Speicher für nächste Argumentstruktur besorgen ----*
.Fill: move.l 4.w,a6
        move.l #EPT_String,d0
        move.l #$10001,d1
        jsr _LVOAllocMem(a6)
        move.l d0,a1
        tst.l d0
        beq.w .OutOfMem

*----- nächste Argumentstruktur füllen -----*
    move.l d0,(a4)
    move.l d0,a4
    addq.l #8,a3
    move.l (a3),EPT_Result1(a1)
    move.l 4(a3),EPT_Result2(a1)
    move.l #EPT_String,EPT_StringSize(a1)
    dbf d7,.Fill

*----- Message schicken -----*
*- Der Eagleplayer kopiert die entsprechenden Daten in selbst -*
*- Allokierten Speicher, so daß der Speicher hier beim Test -*
*----- nach Replien der MSG weiter genutzt werden kann -----*
.NoExtLoading: lea MyMessage(pc),a1
               move.l 4.w,a6
               move.l EP_Port(pc),a0
               jsr _LVOPutmsg(a6)

*----- Auf Message warten -----*

```

```

.Wait:    move.l  TP_Port(pc),a0
         move.l  4.w,a6
         jsr  _LVOWaitPort(A6)
         move.l  TP_Port(pc),a0
         jsr  _LVOGetMsg(A6)
         tst.l  d0
         beq.s  .Wait
         move.l  d0,a1
         cmp.l  #USM_ExternalPrg,UM_Type(A1) ;Handelt es sich um
         beq.s  .Ok ;unsere MSG
         jsr  _LVOREplyMsg(a6)
         bra.s  .Wait

*--- Zurückgekommene Message auswerten ---*
.Ok:    move.l  UM_ArgString(A1),d0 ;Antwort kopieren und
         beq.s  .NotAnswered
         move.l  d0,a4
         move.l  EPT_Stringsize(a4),d0
         move.l  d0,d1
         sub.l  #EPT_String-1,d1
         blo.s  .FreeNextStr
         lea  EPT_String(a4),a3
         lea  StringTemp(pc),a2
.Copy:   move.b  (a3)+,(a2)+
         dbf  d1,.Copy

*----- Argumentenstrukturen freigeben -----*
*- Achtung: Es können immer mehrere Argumentstrukturen gesendet -*
*- werden. Immer alle freigeben !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! -*
.FreeNextStr: move.l  a4,a1
         move.l  EPT_StringSize(a1),d0
         move.l  (a4),a4
         move.l  4.w,a6
         jsr  _LVOfreeMem(a6)
         moveq  #0,d0
         cmp.l  a4,d0
         bne.s  .FreeNextStr

*--- Ausschrift im TestProgramm "Playing ..." ---*
         lea  StringTemp(pc),a1 ;kann verdammt lang
         bsr  ShowStatus ;sein, Aufpassen !!!!
         bra.s  .DelMsgPort

*-- Ausschrift im TestProgramm "Keine Antwort" --*
.NotAnswered: lea  Stat_NoAnswer(pc),a1
              bsr  ShowStatus
.DelMsgPort:  move.l  TP_Port(pc),d0
              beq.s  .NoPort
              move.l  d0,a0
              move.l  4.w,a6
              jsr  _LVODEleteMsgPort(a6)
.NoPort:     clr.l  TP_Port
              rts

.NoEagleplayer: lea  Stat_NoEagleplayer(pc),a1
                bra  ShowStatus

```

```

*-- Anderen Argumentstrukturen freigeben --*
.OutOfMem:  move.l  MyMessage+UM_ArgString(pc),a4
.FreeStruct:  move.l  EPT_String(a4),d0
             move.l  a4,a1
             move.l  (a4),a4
             move.l  4.w,a6
             jsr  _LVOFreeMem(a6)
             moveq  #0,d0
             cmp.l  d0,a4
             bne.s  .FreeStruct
             lea  Stat_OutOfMemory(pc),a1
             bsr  ShowStatus
             rts

*----- Unterprogramm zum printen von Texten -----*
ShowStatus:  rts

*----- Variablenvereinbarungen -----*
Stat_OutOfMemory:dc.b  "Out of Memory !",0
Stat_NoEagleplayer:dc.b  "No Eagleplayer found !",0
Stat_NoAnswer:  dc.b  "Keine Antwort vom Eagleplayer !",0
StringTemp:  ds.b  100
EP_PortName:  dc.b  "EAGLEPLAYERPORT",0 ;Name des Eagleplayerportes
PlayMem.MSG:  dc.b  "EP_PlayMem",0

             even
MyMessage:  ds.b  UM_SizeOf  ;Messagemem

My_Filename:  dc.b  "MDAT.NoName",0  ;Prefix sollte angegeben sein,
             ;um mögliches Playerloading
             ;when needed (Batch) zu
             ;unterstützen

EP_Port:  dc.l  0  ;PortAdr des EP
TP_Port:  dc.l  0  ;PortAdr des TP

*----- Dies ist nur ein Hilfspuffer zum Initialisieren der UM_Message -----*
*----- Er muß dann entsprechend vorher gefüllt werden -----*
DatenPufferAnz:  dc.l  2  ;Anzahl der Files
DatenPuffer:  dc.l  Data1  ;Adr 1 File
             dc.l  Data1Size  ;Size 1 File
             dc.l  Data2  ;Adr 2 File
             dc.l  Data2Size  ;Size 2 File

*----- Testmodule laden -----*
*-- Kann auch im FastMem liegen, weil der Eagleplayer das Module kopiert. --*
*-----*
             incdir  "xModules:TFMX/"  ;entpackt laden !!!
Data1:  incbin  "MDAT.Turrican-1"
Data1Size = *-Data1

Data2:  incbin  "SMPL.Turrican-1"
Data2Size = *-Data2

```

## 1.58 die playerbatch-datei

### 4.4.1 Die PlayerBatch-Datei

---

Die PlayerBatchDatei ist eine StapelDatei, die benötigt wird, wenn man die "Playerloading when needed"-Funktion nutzen will. Das bedeutet, der benötigte Player wird erst geladen, wenn er benötigt wird. Dabei ist es nötig, jedem Musikformat eine eigene Namenskennung als Präfix zu geben. (wie eigentlich auch üblich, also Protracker "MOD.", SonicArranger "SA." usw.) In dieser Datei wird die Namenkennung sowie der Pfad eines jeden Players festgelegt.

```
Beispiel: FC.*|FC13.*   ="DH0:Eagleplayers/FC 1.3"  
*.sdata   ="DH0:Eagleplayers/MusicMaker8"  
MDAT.*    ="DH0:Eagleplayers/TFMX 1.5"  
MDAT.*    ="DH0:Eagleplayers/TFMX Pro"  
MDAT.*    ="DH0:Eagleplayers/TFMX 7V"
```

Es sind, wie man sieht, mehrere Kennungen möglich!  
Weiterhin muß erwähnt werden, daß wenn Eject Players ausgeschaltet ist, jeder Player nur ein Mal eingeladen wird, egal ob er noch in der Liste ist oder nicht.

Es ist auch möglich, verschiedene Player gleichzeitig über eine Kennung zu laden, wie es hier bei TFMX demonstriert wird.

Eine komplette Eagleplayer-BatchDatei liegt dem Originalpaket bei.

Um nun diese Batch Datei einzubinden rufen Sie die Funktion "Load EP-Batch" aus dem "Special"-Menü -> "Preferences" auf und geben dort den Namen Ihrer Stapeldatei an. Danach aktivieren Sie im selben Menü "Use EP-Batch". Nun sollte der Eagleplayer beim nächsten Start viel schneller nutzbar sein aufgrund erheblich gesunkener Ladezeiten. (Vergessen Sie bloß nicht, die Konfiguration abzuspeichern, bevor Sie den Eagleplayer verlassen !!) Aktivieren Sie die "Eject Players"-Funktion, so werden nach dem Laden des nächsten Moduls alle nicht benutzten externen Player rausgeworfen.

## 1.59 die userprogrammbatch-datei

### 4.4.2 Die UserprogrammBatch-Datei

---

Die UserprogrammBatchDatei ist eine StapelDatei, die benötigt wird, wenn man die "Userprogrammloading when needed"-Funktion nutzen will. Das bedeutet, das benötigte Userprogramm wird erst geladen, wenn es benötigt wird.

In dieser Datei wird die Namenkennung sowie der Pfad eines jeden Userprogramms festgelegt.

---

```
Beispiel: "Analyze.small"   ="DH0:EP-Userprograms/Analyze.small"
          "** Time **"      ="DH0:EP-Userprograms/Time"
```

Beachten Sie bitte, daß der Pfadname unbedingt in Gänsefüßchen stehen muß. Die komplette Userprogramm-BatchDatei liegt dem Originalpaket bei.

Die Aktivierung der UserprogramBatch Datei erfolgt analog zu der für die

externen Player

.

## 1.60 fehlermeldungen

### 5.1 Fehlermeldungen

Hier werden einige Fehlermeldungen aufgezählt die auftreten können. Es kann durchaus passieren, daß mehrere Error's sich teilweise decken, dies hängt vom Player oder der Loadroutine ab. Im allgemeinen werden in den original Deli-Players als Fehler in d0=-1 zurückgegeben. Sollte bei einem dieser Player eine Fehlermeldung geschehen und diese zwischen 0 und Max. Fehlernummer liegen, dann wird der jeweilige Fehler ausgegeben. Bei EaglePlayern erwarten wir korrekte Fehlermeldungen. Durch die internen Funktionen im EaglePlayer werden die korrekten Fehlermeldungen zurückgegeben.

| Meldung                                     | Ursache/Erklärung                                                                                    | Behebung / Tip                                            |
|---------------------------------------------|------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| Unknown Module<br>  nicht                   | Pech, das Modul kennt er<br> <br>  alle Entpacklibraries in<br>  LIBS: sind                          | überprüfen, ob alle exter-<br> nen Player geladen sind,ob |
| Error in File                               | Diskette kaputt<br>  Proggis zu retten (Disksalv)                                                    | versuchen mit diversen                                    |
| Corrupt Module<br>  stand, zu kurz, zu lang | Module nicht im Originalzu-<br>  zu laden                                                            | versuchen ins Soundprogramm                               |
| Audiochannels<br>already used !             | Irgendein Programm benutzt<br>  bereits die Soundkanäle<br>  (andernfalls "Cancel" für<br>  Abbruch) | Dieses Programm entfernen<br>  und dann "RETRY" anwählen  |
| Can't alloc Cia-<br>Interrupt               | Der Cia-Interrupt wird von<br>  einem Programm benutzt,z.B<br>  Musik, Modem<br>  wählen             | Dieses Programm entfernen<br>  und dann erneut Play an-   |
| Error whilst De-<br>crunching               | Datei kaputt oder zu kurz<br>  Cruncher zu neu ?<br>  Neu besorgen                                   | versuchen mit original<br>  Packer zu entpacken           |
| -++- bei Fimp &                             | Gepackte Datei länger als                                                                            | -++-                                                      |

|                                       |                                                                |                                                           |
|---------------------------------------|----------------------------------------------------------------|-----------------------------------------------------------|
| Crunchmania                           | ungepackte                                                     |                                                           |
|                                       |                                                                |                                                           |
| Error adding<br>Player                | File kaputt, nicht erkannt,<br>  Speicher alle, kleinere       | Speicher testen, Datei in<br>  Ordnung? Player entpacken! |
| Versionsnummer                        |                                                                | Bereist geladenen Player                                  |
|                                       | löschen und neu versuchen!                                     |                                                           |
|                                       | Neu besorgen                                                   |                                                           |
|                                       |                                                                |                                                           |
| Extern Loading<br>failed              | zum Musikstück zugehörige<br>  Dateien konnten nicht geladen   | Pfade und Playereinstel-<br>  lungen überprüfen, nach-    |
| werden                                | schauen, ob alle Dateien                                       |                                                           |
|                                       | im richtigen Verzeichnis                                       |                                                           |
|                                       | sind                                                           |                                                           |
|                                       |                                                                |                                                           |
| Open Error                            | Datenträger defekt, Datei gar                                  | ?? nachschauen                                            |
| Read Error                            | nicht vorhanden, Name falsch                                   |                                                           |
| Save Error                            | eingegeben ...                                                 |                                                           |
| Write Error                           |                                                                |                                                           |
|                                       |                                                                |                                                           |
| Module to short                       | Module zu kurz oder gepackte                                   | länger save                                               |
| Datei ist kürzer als ange-<br>  geben |                                                                |                                                           |
|                                       |                                                                |                                                           |
| Not Enough Memory                     | Speicher alle ,kann Modul                                      | alles, was stört, raus-                                   |
| nicht laden                           | schmeissen oder mehr RAM                                       |                                                           |
|                                       | kaufen                                                         |                                                           |
|                                       |                                                                |                                                           |
| Error loading Instr-<br>truments      | Ein oder mehrere Instrumente<br>  konnten nicht geladen werden | nachschauen, ob Sample<br>  existiert! Ist genug Mem      |
| Nicht gefunden, Datenträger           | frei ? Pfad bei                                                |                                                           |
| kaputt, nicht erkannt                 | Playerconfig richtig                                           |                                                           |
|                                       | eingestellt                                                    |                                                           |

## 1.61 fehler des programms

### 5.2 Fehler des Programms

Leider gibt es in diesem Programm auch Fehler, umgangssprachlich "Bugs" genannt, die wir entweder aus Zeitgründen noch nicht beseitigen konnten, oder aber noch nicht im Programm aufspürten.

So zum Beispiel sollen die Userprogramme abstürzen, wenn man auf dem A4000 den About-Requester öffnet, eine Weile offenläßt und dann mit "OK" bestätigt. Der Fehler liegt an ARQ 1.66, einem Requesterverbesserer, der den Absturz provozierte. Am besten rauswerfen und auf eine neue Version warten :(-==

Falls Sie Fehler im Programm entdecken, schicken Sie bitte exakte Bugreports, wenn möglich mit Enforcer getestet, an uns.

Alte Versionen hatten Probleme mit RTPatch. Genauere Fehlerbeschreibungen alter Versionen erhalten Sie in der  
Biographie

.

Der Eagleplayer stürzt ab, wenn man mit VZBE arbeitet, von der VirusZ-Package. Allerdings hängen sich auch andere Programme auf, so daß es nicht unsere Schuld sein sollte. (Neuerdings liegt VZBE dem VirusZ-Paket auch nicht mehr bei)

Nach Meinung eines Users soll der EP Probleme mit NoClick haben. Konnte bisher noch nicht nachvollzogen werden !

Ältere Eagleplayerversionen hatten Probleme mit dem Timing von Sound/Pro-trackermodulen auf dem A4000. Nach Umbau der Routinen wurden umfangreiche Tests in verschiedenen Modi von Pal über DBLPal/DBLNTSC/Productivity bis zu Euro72 und Retina Auflösungen durchgeführt, ohne, daß uns Fehler auf-fielen. Falls dennoch Probleme auftreten sollten, so bitten wir Sie, uns dies umgehend mitzuteilen (mit Angabe des Modules, der Rechnerkonfiguration, des verwendeten Screenmodus etc.)

Manchmal kommt es vor, daß bei Verwendung der "Playerbatch"-Funktion plötzlich keine externen Player mehr geladen werden und die Module nicht mehr erkannt und abgespielt werden. Abhilfe schafft das erneute Aufrufen der "Load EP-Batch" Funktion.

## 1.62 problemecke

### 5.3 Problemecke

-----

Symptom: Plötzlich zeigt der EP die Menüs nicht mehr an, obwohl noch ein paar Hundert KBytes Chip-Speicher frei sind, kann er keine Modules mehr über xx-Kbytes laden, der CLI-Befehl AVAIL produziert einen Recovery Alert.  
Erklärung: Einige Player - besonders Soundtracker und deren Mutanten - löschen 2-4 Bytes am Anfang jedes Samples. Wenn die Module nun zu kurz sind, wird freier Speicher gelöscht und sehr oft der Mem-Header Block am Anfang desselben, wodurch dieser Fehler auftritt. Abhilfe schafft hier nur ein Neustart des Rechners. Wir haben durch Anforderung von 2 Kbyte mehr Speicher (bzw. Modi-fizierung der Playroutinen) versucht, dieses Problem zu beheben, jedoch in letzter Zeit trat der Fehler wieder auf.  
In der Players.Dok wird auf besonders gefährliche Playroutinen am Ende hin-gewiesen.  
AHA ! Es lag daran, daß bei besser XPK ein Sicherheitspuffer angefordert werden muß. Wir haben dieses Problem bei Startrekkermodulen bemerkt und behoben.

Ein großes Problem ist die Einbindung ins Multitasking. Die Userprogramme werden als Task mit der Priorität -5 gestartet. Arbeitet nun ein anderes Programm permanent (z.B. ein Raytracer oder ein Packer), so bleibt für die UserPrgs's keinerlei Rechenzeit übrig. Der Haken dabei ist, daß der Eagleplayer jedes Mal, wenn er neu anfängt (oder aufhört) zu spielen, eine Nachricht an die Userprogramme sendet und warten muß, daß sie wieder ankommt, jedoch die User-programme keine Rechenzeit bekommen...

Diese Prozedur kann mitunter ein oder mehrere Sekunden dauern. Sollte aus irgendeinem Grund ein UserProgramm hängen bleiben, bleibt der ganze Player hängen !

Während der Zeit des Message-Handelns wird die Priorität auf 0 gesetzt. Daher wird dies Problem etwas abgerundet.

---

Weiterhin ist es nicht auszuschließen, daß einige externe/interne Player allergisch auf verschiedene Rechnerkonfigurationen reagieren. Beispiel dafür sind TFMX-7V, Hippel, Hippel 7V, VSS.

Bei Soundsystemen, die die Playroutine im Module haben, kann der Umbau manchmal ein oder mehrere Sekunden dauern, der erforderlich ist, um das Module analyserfähig zu machen. Besonders kraß ist dieser Umbau bei Hippel und Whittaker. Manchmal hat man das Gefühl, der Player hängt. Es kann weiterhin passieren, daß der Umbau nicht den gewünschten Erfolg bringt, weil eine uns unbekannte Playroutine verwendet wurde, die andere Routinen verwendet, als die von uns gesuchten. Sollte so ein Fall auftreten, werden die playerspezifischen Gadgets erlaubt, aber bei Benutzen dieser passiert nichts (Bsp: Promizer).

Es gab in der Testphase Probleme auf einer frühen Version der Domino-Grafikkarte. Dort funktionierten die Modi des Analyzers mit 2 Bitplanes teilweise nicht.

Symtom: Der Eagleplayer stürzt nur noch ab. Einer der Gründe könnte darin liegen, daß der Eagleplayer mit Powerpacker gepackt war und inzwischen wieder entpackt wurde. Dabei wird der Eagleplayer NICHT zu 100% wieder hergestellt. Wird dagegen mit DLD entpackt, ist der Player IO. SCHEIß FEHLER !!!

Die Userprogramme brauchen verdammt viel Zeit, wenn sie alle aktiv sind. Einige CPU-Anzeiger zeigen aber eine viel zu hohe CPU-Belastung an, welche dann immer auf den Eagleplayer zurückgeschoben wird. Ein solcher Kandidat ist z.B. der DOPUS. Er zeigt totalen Mist an. Wenn alle Userprogramme inaktiv sind, wird so gut wie gar keine Zeit verbraucht, außer beim Scrollen. Sonst ist der Player ganz normal im Wait-Status. Nachprüfen kann man dieses Phänomen mit dem Perfmon oder besser noch mit XOper !

## 1.63 einleitung zu den externer players

### 6.1 Die Programierung externer Player

-----

Der Eagleplayer unterstützt sog. externe Player. Das sind Executables, die in einem speziellen Format vorliegen und vom EaglePlayer nachgeladen werden können.

Es handelt sich bei den Kapiteln 6.1-6.5 um die originale Anleitung von Delirium (leicht verändert). Auf die Besonderheiten des Eagleplayers wird in Kapitel 6.6 eingegangen.

## 1.64 das externe player konzept

### 6.1.1 Das externe Player Konzept

-----



Externe Player sind Executables (Objekt Files), die den Replaycode eines Soundsystems enthalten. Am Anfang besitzen sie eine charakteristische Playerstruktur. Diese Struktur wird mit dem Macro aus dem Includefile 'misc/deliplayer.i' erzeugt. Es gibt zwei Arten von externen Playern: normale Player und Custom Module.

## 1.65 schematischer aufbau von externen playern

### 6.1.2 Schematischer Aufbau von externen Playern

---

Normale Player unterscheiden sich von Custom Modulen durch das Vorhandensein einer Check Routine und das Fehlen des DTP\_CustomPlayer Tags. Desweiteren enthalten sie natürlich keine Musikdaten :-)

```
{ Playerheader } Kennzeichnet das Object als Player.
{ TagArray } Beschreibung, was der Player kann.
{ Interfacecode } Playername/Registerumsetzung/Checkcode u.ä.
{ Replaycode } Eigentlicher Musikabspielcode.
{ evtl. Daten } und Daten
```

## 1.66 schematischer aufbau von custom modulen

### 6.1.3 Schematischer Aufbau von Custom Modulen

---

Custom Module sind keine Module im herkömmlichen Sinn, sondern im wesentlichen Player, die ein Spezialmodul beinhalten. Mit diesem Format können Sie auch die ausgefallensten Moduleformate leicht an den EaglePlayer anpassen. Sollten für den Player jedoch mehrere Module existieren, ist es in jedem Fall ratsam, einen richtigen Player zu schreiben.

```
{ Playerheader } Kennzeichnet das Object als Player.
{ TagArray } Beschreibung, was der Player kann.
{ Interfacecode } Routinen zur Registerumsetzung u.ä.
{ Replaycode } Eigentlicher Musikabspielcode.
{ evtl. Daten } und Daten
{ SOUND DATEN } Das eigentliche Modul
```

## 1.67 anpassung von playern

### 6.2 Anpassung von Playern

---

Es ist relativ einfach, eine Replayroutine an den EaglePlayer anzupassen. Alles was Sie tun müssen, ist ein wenig Interfacecode zu schreiben. Dies ist aber halb so wild, denn der EaglePlayer stellt ihnen viele hilfreiche Routinen zur Verfügung. Um ein neues Soundsystem anzupassen,

---

benötigen Sie zum einen den Quellcode oder ein linkbares Objektfile der Replayroutine, zum anderen sollten mindestens ca. 5 Module zum Testen der Funktionstüchtigkeit des Players vorhanden sein.

## 1.68 playerheader

### 6.2.1 Playerheader

---

Das PLAYERHEADER Macro generiert den Header, der das File als externen Player identifiziert. Dieses Macro muß angegeben werden und ganz am Anfang des Players stehen. Als einziger Parameter ist ein Pointer auf ein Tag Array einzutragen, in dem alle Funktionen stehen, die der Player dem EaglePlayer zur Verfügung stellt. Bei allen Funktionsaufrufen außer der Interruptroutine (DTP\_Interrupt) enthält a5 die Base. Globale Variablen können dadurch adressiert werden (siehe 'Misc/Deliplayer.i & 'Misc/Eagleplayer.i').

Da der Eagleplayer vor Aufruf einer Playerroutine (außer bei DTP\_Interrupt) alle Register sichert, dürfen diese verändert werden.

PLAYERHEADER <Tagarray>

Tagarray Pointer auf ein Tag Array, das mit TAG\_DONE abgeschlossen sein muß. Tags, bei denen Ti\_Data NULL ist, werden ignoriert.

## 1.69 modulerkennung

### 6.2.2 Modulerkennung

---

Damit EaglePlayer die einzelnen Module unterscheiden kann, befindet sich in jedem Player eine Erkennungsroutine, die auf ein zugehöriges Modul testet. Diese Routine prüft auf Stellen im Modul, die bei allen Modulen eines Players gleich sind. So z.B. auf 'M.K.' bei Offset \$438 im ProTracker-Player. Bei Sound- und NoiseTracker ist nur das Modul abgespeichert, z.B. bei MarkII hängt vor jedem Modul noch die Replayroutine. Bei so einem Modul müssen Sie auf signifikante Assemblerbefehle testen. Ein Vergleich auf ein oder zwei Sprungbefehle, egal ob bra oder jmp, ist hier nicht ausreichend, da bei vielen Modulen dieses Typs Sprungtabellen am Anfang stehen und der Player möglicherweise das falsche Modul erkennen könnte, was in den meisten Fällen zum Absturz führt. Der Player muß genau eine Checkroutine verwenden. Daraus ergeben sich zwei Grundtypen von Playern.

a) Typ eins Player - in der Regel etwas komplexer  
Bei ihnen ist die Check1 Funktion implementiert.

Vorteil: Es lassen sich auch Player realisieren, die das Modul selbst nachladen.

Nachteil: Gepackte Files werden nicht unterstützt.

---

Dieser Player sollte nur für Härtefälle verwendet werden.  
(z.B: IFF-8SVX Player, der das Sample während dem Spielen nachlädt, ...)

b) Typ zwei Player - die einfachere Variante  
Bei ihnen ist die Check2 Funktion implementiert

Vorteil: Das Modul kann gepackt sein, der Player bemerkt davon nichts.  
Nachteil: Das Modul ist in jedem Fall im Chip-Memory.

Es sollte im Normalfall dieser Playertyp verwendet werden.

Egal ob Typ eins oder zwei, wenn der Player das Modul erkannt hat, muß er in d0.l=0 zurückliefern, wenn nicht d0.l<>0.

## 1.70 interrupts

### 6.2.3 Interrupts

-----

Hier gibt es auch eine Einteilung in zwei verschiedene Typen.

- a) Player, die den Eagleplayer Interrupt verwenden  
Vorteil: Der Player ist vom Videomodus unabhängig.  
Besitzt automatisch die Faster/Slower/Speedregler Funktion.  
Kein Aufwand für den Interrupt (Code + Interruptstruktur).  
Ist kompatibel zum serial.device  
Nachteil: Der Interrupt kommt nicht synchron zum VBlank. (Dies führt nur in den seltensten Fällen zu Problemen.)
- b) Player die ihren eigenen Interrupt erzeugen.  
Vorteil: Es können andere Interruptquellen benutzt werden  
(z.B. AudioIRQ).  
Nachteil: Erhöhter Aufwand, bei VBlank nicht mehr unabhängig vom Videomodus.

Wenn ein eigener Timerinterrupt verwendet wird, sollte die CIAB (wg. OS2.0) und die entsprechenden Resourcefunktionen verwendet werden. Außerdem ist es sehr sinnvoll die Replayroutine nicht direkt im CIA-B TimerInterrupt abspielen zu lassen, sondern im Timerinterrupt einen Soft-Interrupt mittels Cause() zu generieren. In diesem SoftInt kann man dann die eigentliche Replayroutine ablaufen lassen. So hat man den Vorteil, daß man wesentlich kompatibler zum serial.device ist. Dies liegt daran, daß SoftInt eine niedrigere Priorität als der der RBF (Read Buffer Full) Interrupt hat, d.h. erst wird der serielle Port bedient, dann erst die Replayroutine. Es wird davor gewarnt, direkt in die Interruptvektoren zu schreiben! Zur Erinnerung: vom Betriebssystem werden die Funktionen AddIntServer() und SetIntVector() zur Verfügung gestellt!

## 1.71 bedeutung der tags

---

### 6.3 Bedeutung der Tags

---

Außer den SystemTags (TAG\_DONE, TAG\_IGNORE, TAG\_MORE, TAG\_SKIP) dürfen folgende Tags verwendet werden:

DTP\_CustomPlayer (BOOL) - dieser Tag deklariert einen Player als Customplayer.

Bei Verwendung dieses Tags sind folgende Tags dann bedeutungslos: DTP\_PlayerVersion

- DTP\_Check1
- DTP\_Check2
- DTP\_Config
- DTP\_UserConfig

DTP\_RequestDTVersion (WORD) - damit kann man sicherstellen, daß mindestens eine bestimmte Version von Eagleplayer vorhanden ist. Dieser Tag muß angegeben werden, wenn bei den EagleplayerGlobals neue Funktionspointer hinzugekommen sind und diese vom Player benutzt werden. ti\_Data ist dabei die Playerrevision.

DTP\_RequestV37 (BOOL) - wenn dieser Tag vorhanden ist, wird der Player nur noch von der Kick 2.0 Version von Eagleplayer geladen (d.h. dtg\_GadToolsBase ist gültig).

DTP\_PlayerVersion (WORD) - Tag, der die Revisionsnummer des Players enthält. Bei zwei Playern mit dem gleichen Playernamen wird derjenige mit der größeren Revisionsnummer geladen.

DTP\_PlayerName (STRPTR) - ti\_Data enthält den Pointer auf den Namen des Players. Dieser String kann zwar beliebig lang sein aber zur Zeit werden nur die ersten 24 Zeichen angezeigt. Dieser Tag muß existieren!

DTP\_Creator (STRPTR) - Pointer auf den Namen des Autors. Dieser wird im Setupfenster im Playerinfofeld angezeigt. Dieser String kann \$A als Zeilenumbruch enthalten.

DTP\_Check1 (FPTR) - Pointer auf eine Modulerkennungsroutine, die aufgerufen wird, wenn 1024 Bytes des Moduls geladen sind. Wird das Modul erkannt, liefert sie d0=0, ansonsten d0<>0.

DTP\_Check2 (FPTR) - Pointer auf eine Modulerkennungsroutine, die aufgerufen wird, wenn das komplette Modul geladen und evtl. entpackt ist. Wird das Modul erkannt, liefert sie d0=0, ansonsten d0<>0.

DTP\_ExtLoad (FPTR) - Pointer auf eine optionale Laderoutine für Module. Ist kein Fehler aufgetreten, wird d0=0 zurückgegeben, sonst d0<>0. Hinweis: Achten Sie darauf, daß im Fehlerfall alle allocierten Ressourcen (Memory, Locks, ...) wieder freigegeben werden, da dann keine weiteren Playerfunktionen mehr angesprochen werden.

DTP\_Interrupt (FPTR) - Pointer auf eine Interruptroutine, die mittels

---

eines Timerinterrupts standardmäßig alle 1/50 sec aufgerufen wird. Dies ist die Standardmethode, um mit der richtigen Abspielgeschwindigkeit im PAL/NTSC/Productivity Videomodus zu spielen. Wenn keine DTP\_Faster/DTP\_Slower Tags angegeben sind, übernimmt Eagleplayer dies durch Verändern der Interruptfrequenz. Dieser Tag kann auch nicht existieren, dann müssen aber DTP\_StartInt/DTP\_StopInt vorhanden sein !

DTP\_Stop (FPTR) - Pointer auf eine optionale Stoproutine. Wenn dieser Tag nicht vorhanden ist, verfährt Eagleplayer folgendermaßen:

- Interrupt stoppen (DTP\_StopInt)
- Sound Cleanup (DTP\_EndSnd)
- Song initialisieren (DTP\_InitSnd)

Ansonsten hat diese Routine die Aufgabe, einen evtl. spielenden Song anzuhalten und so zu initialisieren, daß dieser beim nächsten Starten des Interrupts von Anfang an zu spielen beginnt.

DTP\_Config (FPTR) - Pointer auf eine optionale Initialisierungsroutine.

Diese wird nur einmal unmittelbar nach dem Laden des Players aufgerufen. Mögliche Anwendungen: Laden einer playerspezifischen Konfigurationsdatei.

DTP\_UserConfig (FPTR) - Pointer auf eine optionale Initialisierungsroutine.

Diese wird nur dann aufgerufen, wenn der User den Player im Setupfenster anwählt und das 'Config' Gadget drückt. Mögliche Anwendungen: Öffnen eines Fensters zum Setzen weiterer Optionen wie z.B. Instrumentenpfad und Abspeichern einer playerspezifischen Konfigurationsdatei.

DTP\_SubSongRange (FPTR) - Dieser Tag sollte angegeben werden, wenn der

Player MultiModule unterstützt. ti\_Data zeigt dabei auf eine Routine, die als Returnwert in d0 die minimale und in d1 die maximale Subsongnummer zurückgeben muß.

Hinweis: Wenn möglich sollte dieser Tag (evtl. zusammen mit DTP\_SubSongTest) anstelle von DTP\_NextSong/DTP\_PrevSong verwendet werden.

DTP\_InitPlayer (FPTR) - Pointer auf eine optionale Initialisierungsroutine,

die aufgerufen wird, wenn ein Modul erfolgreich geladen wurde. Tritt kein Fehler auf, liefert sie d0=0, ansonsten d0<>0. Hier muß die Allocation der Audiokanäle stattfinden! (Eagleplayer stellt dafür eine eigene Routine zur Verfügung) Falls der Player Multi-Module unterstützt, muß hier dtg\_SndNum(a5) auf die erste Subsongnummer gesetzt werden. Falls eine Routine für DTP\_SubSongRange existiert, macht Eagleplayer das automatisch (d.h. die Initialisierung von dtg\_SndNum(a5) kann weggelassen werden).

DTP\_EndPlayer (FPTR) - Pointer auf eine optionale Cleanuproutine, die

aufgerufen wird, wenn das Modul aus dem Speicher entfernt wird. Hier muß die Freigabe der Audiokanäle stattfinden! (Eagleplayer stellt dafür eine eigene Routine zur Verfügung)

DTP\_InitSound (FPTR) - Pointer auf eine optionale Initialisierungsroutine.

---

Diese muß das Modul initialisieren, so daß beim Starten des Interrupts der Song von Anfang an zu spielen beginnt.

DTP\_EndSound (FPTR) - Pointer auf eine optionale Cleanuproutine. Diese kann z.B. die Lautstärkeregister und die Audio-DMA rücksetzen.

DTP\_StartInt (FPTR) - Pointer auf eine Initialisierungsroutine, die existieren muß, wenn DTP\_Interrupt nicht vorhanden ist. In diesem Fall muß hier der Sound gestartet werden.

DTP\_StopInt (FPTR) - Pointer auf eine Cleanuproutine, die existieren muß, wenn DTP\_Interrupt nicht vorhanden ist. In diesem Fall muß hier der Sound gestoppt werden.

DTP\_Volume (FPTR) - Pointer auf eine Funktion, welche die Lautstärke neu setzt. Die Funktion wird aufgerufen, wenn die Volume neu gesetzt wird (Slider, ARexx) und beim Initialisieren des Moduls vor DTP\_InitSnd. Die Mastervolume steht in dtg\_SndVol(a5). Die Mastervolume ist dabei der maximale Volumewert. Die effektive Volume errechnet sich also durch:  $VOL_{eff} = (( dtg\_Volume(a5) * modulevolume ) >> 6)$   
Näheres siehe Beispielsourcen.

DTP\_Balance (FPTR) - Pointer auf eine Funktion, welche die Balance neu setzt. Die Funktion wird aufgerufen, wenn die Balance neu gesetzt wird (Slider, ARexx) und beim Initialisieren des Moduls vor DTP\_InitSnd. Die Balance für die linken Kanäle steht in dtg\_SndLbal(a5), für die rechten Kanäle in dtg\_SndRbal(a5). Hinweis: Alle Player die Balance unterstützen können auch Volume! Man verwendet dann die gleiche Routine zum Setzen der Volume&Balance. Die linke Volume errechnet sich wie folgt:  $(( dtg\_Volume(a5) * dtg\_SndLbal(a5) ) >> 6)$   
Entsprechendes gilt für rechts.

DTP\_Faster (FPTR) - Pointer auf eine Funktion, die den Abspielvorgang beschleunigt.

DTP\_Slower (FPTR) - Pointer auf eine Funktion, die den Abspielvorgang verlangsamt.

DTP\_NextPatt (FPTR) - Pointer auf eine Funktion, die den Patternzeiger um eins erhöht.

DTP\_PrevPatt (FPTR) - Pointer auf eine Funktion, die den Patternzeiger um eins erniedrigt.

DTP\_NextSong (FPTR) - Pointer auf eine Funktion, die Subsongnummer auf den nächsten Subsong setzt und diesen spielt.  
(Falls vorhanden)

DTP\_PrevSong (FPTR) - Pointer auf eine Funktion, die Subsongnummer auf den vorhergehenden Subsong setzt und diesen spielt.  
(Falls vorhanden)

DTP\_SubSongTest (FPTR) - (ab Version 1.35) Dieser Tag wird nur ausgewertet, wenn schon der Tag DTP\_SubSongRange angegeben wurde. ti\_Data

---

zeigt dabei auf eine Routine, die als Returnwert in d0 einen Booleschen Wert zurückliefert. Dieser gibt an, ob der SubSong mit Nummer dtg\_SndNum(a5) gültig ist (d0=0) oder nicht (d0<>0). Dieser Tag ist nur bei den Playern nötig, bei denen nicht jeder SubSong (innerhalb der durch DTP\_SubSongRange festgelegten Grenzen) benutzt ist.

## 1.72 eagleplayer support funktionen

### 6.4 Eagleplayer support Funktionen

---

Eagleplayer stellt zur Erleichterung der Playeranpassung einige Funktionen zur Verfügung. Eine Funktion wird wie folgt aufgerufen:

```
move.l dtg_XXX(a5),a0
jsr (a0)
```

Alle folgenden Funktionen außer dtg\_SongEnd und dtg\_SetTimer verwenden d0/d1/a0/a1 als Scratchregister. In a5 muß bei allen Aufrufen (außer bei dtg\_SongEnd und dtg\_SetTimer) die Base stehen. Derzeit existieren folgende Funktionen:

dtg\_GetListData

SYNOPSIS

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

FUNCTION

Liefert Adresse und Länge eines mit LoadFile() geladenen Files.

INPUTS

number - Nummer eines Files beginnend mit 0 für das vom User selektierte File.

RESULT

memory - ein Pointer auf die Startadresse des Files im Speicher oder NULL im Fehlerfall.  
size - Länge des Files in Bytes bzw. 0 im Fehlerfall.

dtg\_LoadFile

SYNOPSIS

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

FUNCTION

Lädt und entpackt ggf. das angegebene File ins Chip-Memory. (Hinweis: diese Funktion ergänzt automatisch, falls das File mit dem angegebenen Namen nicht geöffnet werden konnte '.pp', '.im' und '.xpk')

INPUTS

---

name - der Filename steht in einem internen Buffer (seine Adresse steht in dtg\_PathArray)

## RESULT

success - alles ok d0.l=0, sonst d0.l<>0.

## dtg\_CopyDir

## SYNOPSIS

dtg\_CopyDir()

## FUNCTION

Kopiert das Directory des von User angewählten Files an das Ende des Strings, auf den dtg\_PathArray(a5) zeigt.

## dtg\_CopyFile

## SYNOPSIS

dtg\_CopyFile()

## FUNCTION

Kopiert den Filenamen des von User angewählten Files an das Ende des Strings, auf den dtg\_PathArray(a5) zeigt.

## dtg\_CopyString

## SYNOPSIS

dtg\_CopyString(string)  
a0

## FUNCTION

Kopiert den String, auf den das Register a0 zeigt, an das Ende des Strings, auf den dtg\_PathArray(a5) zeigt.

## INPUTS

string - der Pointer auf den anzuhängenden String steht  
in a0

## dtg\_AudioAlloc

## SYNOPSIS

success = dtg\_AudioAlloc()

## FUNCTION

Belegt alle Audiokanäle.

## RESULT

success - alles ok d0.l=0, sonst d0.l<>0.

## dtg\_AudioFree

## SYNOPSIS



dtg\_AudioFree()

FUNCTION

Gibt die mit dtg\_AudioAlloc belegten Audiokanäle wieder frei.

dtg\_StartInt

SYNOPSIS

dtg\_StartInt()

FUNCTION

Startet den Soundinterrupt. (Falls er nicht schon läuft.) Falls DTP\_Interrupt existiert, startet Eagleplayer einen Timerinterrupt, ansonsten wird DTP\_StartInt aufgerufen.

dtg\_StopInt

SYNOPSIS

dtg\_StopInt()

FUNCTION

Stoppt den Soundinterrupt. (Falls er nicht schon angehalten ist.) Falls DTP\_Interrupt existiert, stoppt Eagleplayer seinen Timerinterrupt, ansonsten wird DTP\_StopInt aufgerufen.

dtg\_SongEnd

SYNOPSIS

dtg\_SongEnd()

FUNCTION

Signalisiert Eagleplayer, daß das Modul einmal komplett gespielt wurde. Diese Funktion verändert keine Register und darf auch von Interrupts aufgerufen werden.

dtg\_CutSuffix

SYNOPSIS

dtg\_CutSuffix()

FUNCTION

Entfernt am Ende des Strings, auf den dtg\_PathArray(a5) zeigt ggf. die Endung '.pp', '.im' oder '.xpk'

dtg\_SetTimer

SYNOPSIS

dtg\_SetTimer()

FUNCTION

---

Programmiert den CIA-Timer mit dem Wert, der sich in dtg\_Timer(a5) befindet. Diese Funktion verändert keine Register und darf auch von Interrupts aufgerufen werden.

## 1.73 tips zur anpassung

### 6.5 Tips zur Anpassung

---

In der Checkroutine und in den Playern überhaupt darf nur auf Speicher zugegriffen werden, der auch im Bereich des geladenen Moduls liegt !! Wir haben eine riesige Menge an Enforcer-Hits bemerkt, die im wesentlichen auf illegale Speicherzugriffe durch die Checkroutinen zurückzuführen sind, wenn das Modul zu kurz ist.

Wenn die Möglichkeit besteht, sollte der neue Player unbedingt mit Enforcer getestet werden. Dabei sollten alle anderen externen Player gelöscht werden.

Diese kleine Checkliste für sollte für die Player/Custommodul Anpassung erfüllt werden, damit der Player systemkonform ist und einwandfrei funktioniert:

- [ ] ist die Checkroutine präzise genug ?
- [ ] werden die Audiokanäle richtig belegt und freigegeben ?
- [ ] wird aller allocierter Speicher freigegeben ?
- [ ] werden alle Locks wieder freigegeben ?
- [ ] enforcer und mungwall - Test bestanden ?
- [ ] werden alle denkbaren Fehler korrekt abgefangen ?
- [ ] wurde der Player mit 1.3, 2.0 und 3.0 getestet ?
- [ ] spielt der Player in allen Videomodi korrekt ?
- [ ] greift der Player nur auf zugewiesenen Speicher zu
- [ ] werden korrekte Fehlermeldungen zurückgegeben
- [ ] Werden überall mögliche Enforcer-Hits abgefangen ?
- [ ] Arbeitet die Checkroutine bei fremdem Module sicher ?  
Enforcergefahr !!!
- [ ] Wird bei GetInfos nur die Adr der Tagliste übergeben ?

Problemecke:

---

| Symptom | mögliche Ursache | Beseitigung |
|---------|------------------|-------------|
|---------|------------------|-------------|

---

Absturz a5 überschrieben

|                                       |  |                  |
|---------------------------------------|--|------------------|
| Register nicht gesichert              |  |                  |
| falsche oder zu späte Initialisierung |  |                  |
| Modul zu kurz                         |  |                  |
| Modul zu neu für den Player           |  | präziserer Check |

klingschrag Audiodaten nicht im Chipmem    Player ins Chipmem

|                                    |  |                  |
|------------------------------------|--|------------------|
| falsche Initialisierung            |  |                  |
| Modul zu neu für den Player        |  | präziserer Check |
| Player benötigt bestimmte Werte in |  | zusätzliche      |

---

einigen Registern Initialisierung  
 unverträglicher Videomodus

kein Ton Audio-DMA abgeschaltet  
 bei >68000 Player schreibt direkt auf Betriebssystem  
 Interruptvektoren benutzen

fast keine fehlerhaftes Interrupt-Handling bei VBlank-IRQ:  
 freie CPU-Zeit Z-Flag setzen

## 1.74 die besonderheiten und neuerungen des eagleplayers

### 6.6 Die Besonderheiten und Neuerungen des Eagleplayers

---

Der Eagleplayer bietet eine Fülle neuer Möglichkeiten, die im folgenden erläutert werden. Folgende Tags sind für Custom-Module im Eagleplayer wieder relevant, z.B. falls es irgendjemand packen sollte, Jason Page-Module als Custom zu erschaffen:

```
DTP_PlayerName
DTP_Creator
DTP_ExtLoad
```

Zu den besonderen Neuerungen gehören vor allem die neuen Eagleplayer-Tags

die

```
'
ModuleInfo-Funktion
'
Analyzerunterstützung
sowie
neuen Globals
.
```

## 1.75 die neuen tags

### 6.6.1 Die neuen Tags

---

EP\_Get\_ModuleInfo (FPTR) - Diese Funktion hat in A0 den Zeiger auf eine fertig initialisierte Module-Info Tagliste zurückzugeben oder Null für keine TagListe. Weitere Informationen finden Sie im Kapitel 6.6.2 Moduleinfo

EP\_Free\_ModuleInfo (FPTR) - Diese optionale Funktion kann zum Beispiel dazu dienen, Speicher, der für die Moduleinfotagliste allokiert wurde, freizugeben. Es werden keinerlei Returnwerte erwartet.

EP\_Voices (FPTR) - Über diese Funktion werden die Werte für die einzelnen Stimmen in den Bits 0-3 des Datenregisters D0 übergeben. Ist das jeweilige Bit gesetzt, so ist die entsprechende Stimme

---

eingeschaltet. Bit 0 entspricht Kanal 0 , Bit 1 = Kanal 1 usw.

EP\_Structinit (FPTR) - übergibt im Adressregister A0 einen Pointer auf eine UPS\_USER - Struktur, in der der jeweilige Player die Informationen für die Analyzerprogramme (Userprogramme) ablegt oder Null. Diese Strukturadresse wird dann vom Eagleplayer an alle Userprogramme weitergegeben, die sie dann auswerten.

EP\_StructEnd (FPTR) - Optional, die Struktur wird freigegeben, der Speicher, der eventuell allokiert wurde, darf freigegeben werden.

EP\_LoadPlConfig (FPTR) - ruft Routine auf, die die Konfiguration des jeweiligen Players lädt ( vorzugsweise aus 'ENV:Eagleplayer/x.config' )

EP\_SavePlConfig (FPTR) - Routine zum Sichern der Playerkonfiguration nach 'Envarc:Eagleplayer/x.config' (Vorschlag der Autoren)

EP\_GetPositionNr (FPTR) - gibt in D0.1 die aktuelle Patternnummer zurück, die gerade gespielt wird, benötigt für Patternumschaltung (dabei Scrollt der Eagleplayer die jeweilige Position ins Fenster

EP\_SetSpeed (FPTR) - Für Player mit eigenem Timer wird hier die Möglichkeit gegeben, den Speedregler zu nutzen. Dabei wird in D0 ein Wert von -25 bis +25 übergeben, der den aktuellen Stand des Reglers repräsentiert. 0 Stellt dabei den Standardwert dar.

EP\_Flags (LONG) - Die Flags geben an, welche Funktionen der Player grundsätzlich unterstützt. Dieser Tag wurde eingeführt, seitdem wir "modifizierende" Player erschufen, die versuchen, Routinen in Modulen mit Playroutine darinnen so zu modifizieren, daß z.B. Analyzerunterstützung möglich ist. Da sich solche Replays voneinander unterscheiden, kann es sein, daß die eine oder andere Funktion von Modul zu Modul nicht möglich ist. Daher gibt es folgende Flagbits:

EPF\_Songend - Der Player unterstützt Songend

EPF\_Restart - Modul läßt sich mehrfach abspielen (teilweise nicht möglich !)

EPF\_Disable - Player ist nicht erlaubt

EPF\_NextSong - kann nächsten Untersong anwählen

EPF\_PrevSong - kann vorherigen Untersong anwählen

EPF\_NextPatt - kann nächstes Pattern anwählen

EPF\_PrevPatt - kann vorheriges Pattern anwählen

EPF\_Volume - Lautstärkeregelung möglich

EPF\_Balance - Balance möglich

EPF\_Voices - Stimmenbeeinflussung möglich

EPF\_Save - Modul kann gesichert werden

EPF\_Analyzer - Analyzerunterstützung

EPF\_ModuleInfo- Informationen über Modul erhältlich

EPF\_SampleInfo- Informationen über verwendete Sample erhältlich

EPF\_Packable - Module darf gepackt werden

EPF\_VolVoices - Die Lautstärken der Stimmen können unterschiedlich sein (EPG\_VolVoicel...)

Dieser Tag dient vor allem Informationszwecken im PlayerWindow.

EP\_KickVersion (WORD) - Mindeste Kickstartversion (37 für Kick 2.0)

EP\_PlayerVersion (LONG) - Mindeste EaglePlayer-Version  
Wenn der Player nur vom EaglePlayer geladen werden soll, kann bei DTP\_RequestDTVersion eine -1 eingetragen werden.

EP\_EjectPlayer (FPTR) - Falls der Player irgendwelchen Speicher, Filelocks usw. besorgt, kann der an dieser Stelle freigegeben werden, danach wird der Player durch den Eagleplayer entfernt.

EP\_Date (LONG) Datum, an dem der Replayer geschrieben wurde.

EP\_Check3 (FPTR) - Checkroutine, die nach dem Laden der ersten 1000 Byte aufgerufen wird. Sollte das Module erkannt werden, wird es ins Fastram geladen.

## 1.76 moduleinfo

### 6.6.2 Moduleinfo

-----

Für die Moduleinfofunktion stellt der Eagleplayer eine Reihe von Tags zur Verfügung, die Auskunft über das aktuelle Modul ermöglichen. Die Tagliste wird bei Aufruf von "EP\_Moduleinfo" (siehe auch dort im Kapitel 6.6.1) in A0 übergeben. Bitte beachten Sie, daß bisher noch nicht alle Informationen auch angezeigt werden. In der registrierten Version wird es ein Window geben, in dem hoffentlich alle Informationen ausgewertet werden.

Ab Eagleplayer V1.50 gibt es einen Tag, der gleich auf die Tagliste der ModuleInfotags zeigt.

MI\_SongName (STRPTR) - Songname, der mitunter im Modul zu finden ist. Wird eine Null in TI\_Data übergeben, so erscheint bei Moduleinfo ein "Unknown" als Songname. Sehr komfortable Möglichkeit, den richtigen Namen gerippter Module zu erhalten.

MI\_AuthorName (STRPTR) - Name dessen, der den Song schrieb, bei Rückgabe von Null in TI\_Data gibt der Eagleplayer ein "Unknown" aus.

MI\_SubSongs (LONG) - Anzahl der Untersongs im Modul

MI\_Pattern (LONG) - Anzahl der Patterns im Modul

MI\_MaxPattern (LONG) - Maximale Anzahl der Patterns (z.B. Soundtracker: 64)

MI\_Length (LONG) - Länge des Songs (z.B. in Patterns)

MI\_MaxLength (LONG) - Maximale Länge des Songs (z.B. Soundtracker 127)

MI\_Steps (LONG) - Anzahl der Steps (BP Soundmon)

MI\_MaxSteps (LONG) - Max. Anzahl der Steps

MI\_Samples (LONG) - Anzahl der benutzten Samples

MI\_MaxSamples (LONG) - Max. Anzahl der Samples (z.B. Protracker: 31)

MI\_SynthSamples (LONG) - Anzahl der benutzten synthetischen Samples

MI\_MaxSynthSamples (LONG) - Maximale Anzahl der synthetischen Samples

MI\_Songsize (LONG) - Größe des Songs in Bytes

---

MI\_SamplesSize (LONG) - Länge der Samples in Bytes

MI\_ChipSize (LONG) - benutzter Chip-Speicher in Bytes

MI\_OtherSize (LONG) - benutzter Fast-Speicher in Bytes

MI\_Calcsize (LONG) - berechnete Länge des Modules in Bytes

MI\_SpecialInfo (STRPTR) - Zeiger auf Sonderinformationen als Text

MI\_LoadSize (STRPTR) - Anzahl der geladenen Bytes für SoundSysteme, die externe Dateien nachladen

MI\_Unpacked (LONG) - Ungepackte Länge in Bytes (z.B. wie lang ein Propacker-Modul als Protracker wäre)

MI\_UnPackedSystem (LONG) (STRPTR) - gibt an, aus was dieses Format entstand, entweder eine interne Nummer (siehe unten) oder ein String, der den Namen enthält

Folgende Varianten wurden bisher vorgesehen

- MIUS\_OldSoundtracker
- MIUS\_Soundtracker
- MIUS\_Noisetracker
- MIUS\_Protracker

MI\_Prefix (STRPTR) - Zeiger auf ein Präfix für den Namen des Modules, so z.B. 'Mod.' oder 'Mdat.'. So kann man das Modul unter dem richtigen Namen mit einer passenden Kennung abspeichern.

MI\_About (STRPTR) - Zeiger auf einen Informationstext zum Player.

MI\_MaxSubSong (LONG) - Anzahl der maximal möglichen Untersongs bei diesem Soundformat.

MI\_Voices (LONG) - Anzahl der benutzen Stimmen bei diesem Soundformat.

MI\_MaxVoices (LONG) - Anzahl der maximal möglichen Stimmen bei diesem Soundformat.

## 1.77 analyzeransteuerung

### 6.6.3 Analyzeransteuerung

-----

Die Analyzeransteuerung erfolgt mit Hilfe der UPS\_USER - Struktur, die im folgenden erläutert wird. (Übergabe dieser siehe Kapitel 6.6.1 "EP\_Structinit)

Ab Eagleplayer V1.50 kann eine interne UPS\_Structur verwendet werden. Die Adresse steht in EPG\_UPS\_Structure. Sie ist vorinitialisiert und wird bei jeder Volume/Balance/Voice-Änderung automatisch gefüllt (UPS\_DMACon). Dabei muß unbedingt das Flag EPF\_InternalUPSStructure beim Tag EP\_Flags gesetzt werden. Sie muß und darf nicht freigegeben werden.

Achtung. Es kann passieren, daß die Struktur in den nächsten Versionen des Eagleplayers geändert wird, um auch SoundKarten und A5000 zu unterstützen!

---

Die Struktur sieht so aus:

```
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
```

Die Einträge haben folgende Bedeutungen

UPS\_Voice?Adr - Adresse des Samples, das gerade auf dieser Stimme gespielt wird

UPS\_Voice?Len - Länge des Samples, das gerade auf dieser Stimme gespielt wird

---

UPS\_Voice?Per - aktueller Wert der Sampleperiod, spielt eine Schlüsselrolle, wird eine Periode <>0 übergeben, so heißt das im allgemeinen, daß eine neue Note gespielt wird. Die Sampleperiod ist unabdingbar für den Analyzerbetrieb. Können Sie nicht herausfinden, wann eine Note angespielt wird, so setzen Sie die Periode halt immer dann, wenn auf die Audio-Hardware zugegriffen wird (\$DFF0A6/B6/C6/D6), gilt auch für Samplelänge und Adresse

UPS\_Voice?Vol - Lautstärke, die auf die Hardwareregister geschrieben werden soll, Lautstärkeregelung wird dabei nicht berücksichtigt, d.h. wenn der UrWert z.B. 64 ist, die Lautstärke aber nur 32 beträgt, will ich nicht 32, sondern 64 sehen, verstanden ! (Berücksichtigung der Lautstärke ist schon anderweitig vorgesehen, Siehe UVolume)

UPS\_Voice?Note- noch nicht unterstützt

UPS\_Voice?Samplnr - gibt die aktuelle Samplenummer an , noch von keiner Playroutine und keinem Userprogramm unterstützt.

UPS\_Voice?Sampletyp - Sampletyp, noch nicht unterstützt

UPS\_Voice?Repeat - gibt an, ob das Sample nur einfach - oder sich wiederholend gespielt wird , wenn der Wert 0 ist, heißt das Repeat ein, wenn er 1 ist, so ist der Repeat ausgeschaltet

UPS\_DMACon - gibt an, welche Stimmen ein/ausgeschaltet sind, Bit 0 für Kanal 0 ,Bit 1 = Kanal 1 usw. , ist das Bit gesetzt, so ist der Kanal eingeschaltet (Name etwas verwirrend, der Übergabewert sollte sich eigentlich auf die "EP\_Voices"-Funktion ,Kapitel 6.6.1 beziehen)

UPS\_Flags - Flagbits, die angeben , welche Möglichkeiten der UPS\_USER - Struktur der jeweilige Player nutzt.

UPSFL\_Adr - Sampleadresse

UPSFL\_Len - SampleLänge

UPSFL\_Per - Sampleperiod (WICHTIG!)

UPSFL\_Vol - Lautstärke

UPSFL\_Note - Note, noch nicht unterstützt

UPSFL\_SNr - Samplenummer

UPSFL\_STy - Sampletyp, noch nicht unterstützt

UPSFL\_DMACon - welche Stimmen an/aus sind

UPS\_Enabled - gibt an, ob Zugriff auf die Struktur erlaubt ist, 0 heißt ja, eine Angabe <>0 bedeutet, daß die Struktur zur Auswertung gesperrt ist.

Die restlichen Einträge sind für zukünftige Versionen des Eagleplayers vorgesehen.

----- Achtung -----  
Für die derzeitigen Userprogramme wird erwartet, daß mindestens UPSF\_Adr, UPSF\_Len, UPSF\_Per, UPSF\_Dmacon und UPSF\_Vol gesetzt und unterstützt werden. Werden noch die anderen Parameter (UPS\_Voice?Adr, UPS\_Voice?Len, UPS\_Voice?Per,



UPS\_Voice?Vol) gesetzt und UPS\_Enabled nach verlassen der Playroutine "0" ist.

---

## 1.78 die neuen eagleplayer-globals

### 6.6.4 Die neuen Eagleplayer-Globals (ab Eagleplayer V1.10+)

---

Nachdem in der Vergangenheit auf Änderungen in den Globals verzichtet wurde, ist es notwendig geworden, einige Merzzellen bzw. Unterprogramme hinzuzufügen.

Bei den neuen Eagleplayer-Globals werden die Argumente nicht mehr in Registern sondern in Argumentzellen übergeben. Davon sind 8 Stück vorhanden. In der Merzzelle EPG\_ArgN muß immer die Anzahl der Argumente stehen. Sollte ein Unterprogramm mehrere Argumente verlangen, müssen diese auch übergeben werden und vor allem muß EPG\_ArgN immer auf den max. gesetzt werden. Sollte die Parameterübergabe anders von statten gehen, wird darauf hingewiesen.

Die Unterprogramme des Eagleplayers dürfen von jedem Replayer genutzt werden, außer im Interrupt. Wenn nicht anders darauf hingewiesen wird, dürfen Userprogramme die Subroutinen nur nach einen USClass\_LockEP benutzen.

Im folgenden werden die wichtigsten Eagleplayer-Unterprogramme erklärt:

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

Es wird ein Speicherbereich unter Berücksichtigung des Save-Modes gesichert. Diese Funktion ist erst in der registrierten Version möglich. Ist der Savemode -1, wird der im Eagleplayer eingestellte Save-Mode verwendet.

Input: Arg1 = Startadresse  
 Arg2 = Länge des Speicherbereiches  
 Arg3 = Pathadresse  
 Arg4 = SaveMode (-1=Eagleplayereinstellung  
           0=nicht gepackt  
           1=PP-Crunched  
           2=LH-Crunched  
           3=XPK-Crunched  
 Arg5 = Flags  
       Bit 0=0   Anzeige im Playerwindow 0=ja  
       Bit 1=1   Zieldatei immer deprotecten  
       Bit 2=1   Safe Save  
 ArgN = 5

Output: Arg1 = Ergebnis (0=Alles ok)

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

Es wird ein Filerequester unter Berücksichtigung des eingestellten Filereq-Mode geöffnet. Es kann zwischen einer FileSelektion und einer DirSelektion unterschieden werden.

---

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

Output: Arg1 = Ergebnis (0=Cancel oder Systemfehler, sonst 1)

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

Es wird ein Textrequester geöffnet. Übergeben werden muß ein Ascii-Text. Dieser wird dann ausgewertet und das Window wird der Größe des Textes angepaßt. In dem Text können Kennungen übergeben werden, die auf Argumente zeigen, die aus der Argumentenliste entnommen werden. Zudem kann angegeben werden wie viele Gadgets man verwenden will und es können eigene Image-Daten ins Window mit übernommen werden. Auf Kick2.0 ist der Textrequest Publicscreen unterstützt.  
Hinweis: Die Routine weißt noch einen bisher nicht gefundenen Bug auf. Die Routine sollte aber trotzdem in Playern benutzt werden, weil in späteren EPversion der Fehler hoffentlich beseitigt ist.

Input: Arg1 = TextAdresse  
Arg2 = Pointer to Pubscreenname (nur Kick2.0, sonst 0)  
Arg3 = Position on Screen (x.w & y.w)  
Arg4 = Pointer to Gadgetnames  
Arg5 = Poniter to Requestername  
Arg6 = Pointer to ArgumentListe  
Arg7 = Pointer to ImageDatas  
ArgN = 7

Kennungen für Argumente: %s - String  
%d - Zahl in Dezimal angeben

Output: Arg1 = Ergebnis 0=Fehler (z.B Window zu groß)  
sonst Nummer des Gadgets

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

Es wird ein ausführbares Programm geladen. Es wird entpackt, falls dies möglich ist

Input: Arg1 = FilePath  
ArgN = 1

OutPut: Arg1 = Einsprungsadresse des Programms  
d0 = Fehler (0=alles ok)

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

---

Wie DTG\_LoadFile, nur das hier die Memeigenschaften mit angegeben werden.

Input: Arg1 = Memeigenschaften  
ArgN = 1  
DTG\_PathArrayPtr = Path des Files

Output: d0 = Ergebnis (0=alles ok)

-- EPG\_ScrollText --

Scrollt den angegebenen Text ins Statuswindow des Eagleplayers. Wird der Text mit Null abgeschlossen, bleibt er stehen, wird er mit eins abgeschlossen wird er im Loop gescrollt, wird er mit zwei abgeschlossen, wird der Text bis an den linken Rand gescrollt, falls dieser noch nicht erreicht ist.

Input: Arg1 = Textadresse  
ArgN = 1

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

Lädt eine PlayerConfig. Diese Funktion wurde eingeführt um beim Laden der Replayer, wenn kein Env-Verzeichnis existiert, nicht andauernd Cancel zu drücken. Es wird getestet, ob die Config im Env-Verzeichnis liegen soll oder nicht. Ist nun das Verzeichnis nicht vorhanden, wird keine Config geladen. (Funktion noch nicht eingebaut)

Input: Arg1 = ConfigPath  
ArgN = 1

Output Arg1 = Ergebnis

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

Speichert eine PlayerConfig ab. Diese Funktion wurde eingeführt um beim Saven der ReplayerConfiguration, wenn kein Env-Verzeichnis existiert, nicht Cancel zu drücken. Es wird getestet, ob die Config im Env-Verzeichnis liegen soll oder nicht. Ist nun das Verzeichnis nicht vorhanden, wird keine Config geladen. (Funktion noch nicht eingebaut)

Input: Arg1 = ConfigPath  
Arg2 = Startadresse  
Arg3 = Endadresse  
Arg4 = SaveMode (siehe EPG\_SaveMem)

Output: Arg1 = Ergebnis (0=alles ok)

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

Sucht einen Tag in der angegebenen TagListe. Die Funktion ist Kickstart unabhängig und darf auch von Userprogrammen aus ohne USClass\_LockEP benutzt werden.

Input: a0 = Tagliste

---

d0 = Tag

Output: d0 = Wert des Tags

d1 = Tag gefunden (0=nein,dann ist d0 auch 0, 1=ja)

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

Sucht den Autor eines Musicstückes in den angegebenen Grenzen. Die Routine wird normalerweise beim Soundtracker und seinen Mutanten angewendet, sie kann aber auch auf andere Systeme übertragen werden. Es wird in den Samplennamen nach der Kennung "by" bzw. "#" gesucht. Der nächste String ist dann der Autorname. Der Autorname muß nicht copiert werden. Es reicht aus, wenn die Adresse des Autornamens in den ModuleInfo-Tag eingetragen wird.

Input: Arg1 = Start des 1. Samplennamens

Arg2 = Offset zum nächsten Sample

Arg3 = Länge des Samplennamens

Arg4 = Sampleanzahl

ArgN = 4

Output Arg1 = Pointer to Autorstring oder NULL

Arg2 = Länge des Autorstrings oder NULL

----- EPG\_Hexdez -----

Convertiert die Hexzahl in d0 in eine dezimale Ascii-Darstellung. Diese Funktion kann auch von Userprogrammen ohne Probleme genutzt werden.

Input: d0 = Hexzahl

d1 = Flags Bit 0=1 Nullen verstecken

Bit 1=1 Vorzeichen benutzen

a0 = OutPutpuffer

OutPut: -

----- EPG\_TypeText -----

Es wird ein Text ins Mainwindow geprintet. Diese Funktion kann auch von Userprogrammen genutzt werden.

Input: A0 = Adresse ds Testes

OutPut: -

----- EPG\_ModuleChange -----

Ein Module wird nach den Vorgaben umgebaut. Diese Funktion wird benutzt um Module mit Playroutine analyzerfähig und systemkonform zu machen. Vor- und nach Ablaufen der Hauptfunktion wird der Cache, falls vorhanden, gelöscht.

Input: Arg1 = Startadresse der umzubauenden Daten

Arg2 = max Länge

Arg3 = Umbautabelle

Arg4 = 1.b=1 Mehrmals eine Routine umbauen

2.b=1 2. Umbauroutine benutzen

3.b=1 keine Suche nach Werten

```

    4.b=1 Keine Suche nach Jump
    Arg5 = 1.w Kennbyte für Jump
    2.w Kennbyte für Wert
    ArgN = 5
    OutPut: Arg1 = Fehlernummer oder Null
    Arg2 = Anzahl der Umbauten

```

----- EPG\_ModuleRestore -----

Ein Module muß vor dem Saven in den Originalzustand zurück gesetzt werden. Dies erledigt die Funktion ModuleRestore. Vor- und nach Ablaufen der Hauptfunktion wird der Cache, falls vorhanden, gelöscht.

```

Input: Arg1 = Startadresse der umzubauenden Daten
    Arg2 = max Länge
    Arg3 = Umbautabelle
    Arg4 = 1.b=1 Mehrmals eine Routine umbauen
    2.b=1 2. Umbauroutine benutzen
    3.b=1 keine Suche nach Werten
    4.b=1 Keine Suche nach Jump
    Arg5 = 1.w Kennbyte für Jump
    2.w Kennbyte für Wert
    ArgN = 5
    OutPut: Arg1 = Fehlernummer oder Null
    Arg2 = Anzahl der Umbauten

```

Ab der Eagleplayerversion 1.50 gibt es einen Eintrag in den Globals namens "EPG\_NewJumpTab". Dies ist ein Pointer auf noch mehr Unterprogramme die genutzt werden können. Der große Unterschied besteht darin, das diese direct angesprungen werden können, also wie eine Library ! Diese Funktion ist zwar implementiert, allerdings darf sie noch nicht genutzt werden.

Desweiteren sind in den Globals die benutzten Librarybasen festgehalten. Unterlassen Sie es unbedingt, diese zu verändern. Sämtliche neuen Globals sind nur Lese-Globals, außer den Argumentzellen.

## 1.79 einföhrung und beschreibung des arexx-portes

### 7.1 Einföhrung und Beschreibung des Arexx-Portes

Der Eagleplayer unterstötzt nun endlich (ab V1.10) Arexx. Dazu stellt der Eagleplayer eine Reihe von Kommandos zur Verfügung, durch die der Player im Prinzip fernbedienbar wird.

Eine

Anpassung an DOPUS

ist daher überhaupt kein Problem.

## 1.80 arexx-kommandoübersicht

### 7.2 Arexx-Kommandoübersicht

---

Im folgenden sehen sie die Arexx-Kommandos, die der Eagleplayer unterstützt. Ist kein Fehler aufgetreten, wird als Result der ScrollText zurückgegeben, der im Statuswindow scrollen würde, außer wenn natürlich konkret Argumente als Returnwerte, z.B. beim Status, erwartet werden. Ist ein Fehler aufgetreten steht in rx\_Result der Fehlertext.

Der Anwender sollte natürlich testen, ob die einzelnen Funktionen auch nutzbar sind, um sein Arexx-Programm ordentlich anzupassen. Wenn z.B ein Module gesaved werden soll, aber dies der Replayer nicht erlaubt, dann funktioniert das eben nicht.

Die Angaben in eckigen Klammern können gemacht werden, die in spitzen Klammern müssen gemacht werden.

Wenn im folgenden von Filenamen gesprochen wird, dann wird grundsätzlich der ganze Path mit angegeben (falls nicht anders erwähnt).

| Kommando                               | Bedeutung                                                                                                       |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| Iconify                                | Der Eagleplayer wird in den Iconify-Modus versetzt oder verläßt ihn.                                            |
| PubScreen <PubScreen><br>PublicScreen. | Öffnet den Eagleplayer auf dem angegebenen PublicScreen.                                                        |
| AboutEP                                | Gibt einen String über den Eagleplayer aus.                                                                     |
| ScrollText <ScrollText>                | Der Text scrolt in's StatusWindow.                                                                              |
| HelpMode <Zustand>                     | Der HelpModus wird an/ab geschaltet.                                                                            |
| HelpMe <Nr>                            | Gibt den Helpstring für die angegebene Funktion zurück.                                                         |
| DeleteFile [Filename]                  | Es wird die angegebene Datei gelöscht.                                                                          |
| UserPrg <UserPrgNr> <Zustand>          | Schaltet das angegebene Userprogramm an/ab.                                                                     |
| LoadConfig <Filename>                  | Lädt eine neue Eagleplayer-Config.                                                                              |
| SaveConfig <Filename>                  | Speichert die aktuelle Eagleplayer-Config ab.                                                                   |
| AddUserPrg [Filename]                  | Es wird das angegebene Userprogramm geladen.<br>Wurde kein Filename übergeben, wird ein Filerequester geöffnet. |
| AddUserDir [Dirname]                   | Es wird das komplette Dir geladen.                                                                              |

---

Wurde kein Dirname übergeben, wird ein Filerequester geöffnet.

HelpFile <Filename> es wird ein Helpfile geladen.

Font <Fontname> Lädt einen neuen Font für den Eagleplayer.

LoadModule [Filename] Lädt das angegebene Modul ein und spielt es ab, wenn Quickstart auf ON geschaltet ist. Ist kein Filename angegeben, wird ein Filerequester geöffnet.

LoadModule <Dirname> <Filename> <Filename> ...  
Übernimmt die angegebenen Module, wie bei Multi-select, in eine Liste, die u.a. im Pull-Down Menu zu sehen ist. Das Laden des Directories entfällt. Als Filename wird hier nur der Name (kein Path) angegeben !

SaveModule [Filename] Das geladene Module wird nach Filename abgespeichert. Ist kein Name angegeben, dann öffnet sich ein Filerequester.  
(nur registrierte Version)

About Module Gibt Scrolltext, der im StatusWindow angezeigt wird, zurück.

EjectModule Entfernt das Module aus dem Speicher.

Stop Der Player stoppt das Abspielen. Bei Anwählen von PlayModule wird es von Anfang an gespielt.

ReplaySong Stoppt das Abspielen und beginnt mit dem Abspielen am Anfang des Modules.

Play Der Player startet das Abspielen.

Pause Der Player stoppt das Abspielen. Beim Anwählen von Play wird das Abspielen dort weitergeführt, wo es aufgehört hat.

NextSong Es wird der nächste Subsong abgespielt.

PrevSong Es wird der vorherige Subsong abgespielt.

NextModule Es wird das nächste Module in der Liste abgespielt.

PrevModule Es wird das vorherige Module in der Liste abgespielt.

NextPattern Das nächste Pattern wird abgespielt.

PrevPattern Das vorherige Pattern wird abgespielt.

OpenMainWindow Es wird das Hauptwindow geöffnet.

---

---

OpenPlWindow        Es wird das PlayerWindow geöffnet.

CloseMainWindow     Das Mainwindow wird geschlossen.

ClosePlWindow       Das PlayerWindow wird geschlossen.

AddPlayer [Filename]    Es wird der angegebene Replayer geladen. Wurde kein Filename übergeben, wird ein Filerequester geöffnet.

AddPlDir [Dirname]    Es wird das angegebene Verzeichnis mit Replayern geladen. Wurde kein Dirname übergeben, wird ein Filerequester geöffnet.

DeleteAll            Löscht alle externen Replayer.

DeletePlayer <Nummer>    Es wird der angegebene externe Player gelöscht.

Enable <Nummer>        Der angegebene Player wird erlaubt.

Disable <Nummer>       Der angegebene Player wird verboten

LoadPlConfig <Nummer>    Die Config des Players wird geladen.

SavePlConfig <Nummer>    Die Config des Players wird gespeichert.

PlConfig <Nummer>      Die Config des Pllayers wird aufgerufen.

DefaultSpeed         Der Speed wird auf den normalen Speed gesetzt.

SlowerSpeed         Der Speed wird um eine Stufe langsamer.

FasterSpeed         Der Speed wird um eine Stufe schneller.

FadeOut <On/Off>       FadeOut wird an- bzw. ausgeschaltet.

FadeIn <On/Off>        FadeIn wird an- bzw. ausgeschaltet.

Filter <On/Off>        Filter wird an- bzw. ausgeschaltet.

FlashPointer <On/Off>    Flashpointer wird an- bzw. ausgeschaltet.

MasterVolume <On/Off>   Mastervolume wird an- bzw. ausgeschaltet.

Quickstart <On/Off>    Quickstart wird an-bzw. ausgeschaltet.

ScrollInfos <On/Off>    ScrollInfos wird an- bzw. ausgeschaltet.

UseSongName <On/Off>    Use Songname wird an- bzw. ausgeschaltet.

UsePrefix <On/Off>     Use Prefix wird an- bzw. ausgeschaltet.

SaveT <On/Off>        SaveT wird an- bzw. ausgeschaltet.

QuitSurface <On/Off>    Quit Surface wird an- bzw. ausgeschaltet.

---



QuitEagle <On/Off> Quit Eagle wird an- bzw. ausgeschaltet.

AutoPassword <On/Off> Auto Passwort wird an- bzw. ausgeschaltet.

EjectPlayers <On/Off> Eject Players wird an- bzw. ausgeschaltet.

UsePlBatch <On/Off> PlayerBatch wird an- bzw. ausgeschaltet.

LoadPlBatch [Filename] Es wird eine PlayerBatch-Datei geladen.  
Wenn kein Filename angegeben wird, wird ein  
Filerequester geöffnet.

EjectUserPrg <On/Off> Eject Userprograms wird an- bzw. ausgeschaltet.

UseUPrgBatch <On/Off> UserprogrammBatch wird an- bzw. ausgeschaltet.

LoadUPrgBatch [Filename] Es wird eine UserprogrammBatch-Datei geladen.  
Wenn kein Filename angegeben wird, wird ein  
Filerequester geöffnet.

LoadDir <On/Off> LoadDir wird an- bzw. ausgeschaltet.

EPDir <On/Off> EPDir wird an- bzw. ausgeschaltet.

AutoDir <On/Off> AutoDir wird an- bzw. ausgeschaltet.

Notify <On/Off> Notify wird an- bzw. ausgeschaltet.

ScanAlways <On/Off> Scan Always wird an- bzw. ausgeschaltet.

ShowDirNames <On/Off> ShowDirNames wird an- bzw. ausgeschaltet.

HideAll <On/Off> HideAll wird an- bzw. ausgeschaltet.

CheckDifference <Zahl> Die Checkdifferenz wird neu gesetzt.  
Nicht eingebaut !!!

EntriesPerMenu <Zahl> Die Einträge für ein Menü werden neu gesetzt.

Menumem <Zahl> Der interne Dirmem Speicher wird geändert.

NoNewModule Der ProgrammMode wird neu gesetzt  
PrevModule Der ProgrammMode wird neu gesetzt  
NextModule Der ProgrammMode wird neu gesetzt  
RandomModule Der ProgrammMode wird neu gesetzt

RandomSong <On/Off> Randomsong wird an- bzw. ausgeschaltet.

LoadBefore <On/Off> Load Before wird an- bzw. ausgeschaltet.

AutoSubSong <On/Off> AutoSubSong wird an- bzw. ausgeschaltet.

LoadAlways <On/Off> Load Always wird an- bzw. ausgeschaltet.

Songend <On/Off> Songend wird an- bzw. ausgeschaltet.

---

TimeOut <Time>        Es wird die angegebene Spielzeit gesetzt.  
 SmallModule <On/Off>    Small Module wird an- bzw. ausgeschaltet.  
 BigModule <On/Off>     Big Module wird an- bzw. ausgeschaltet.  
 ModuleFault <On/Off>    Module Fault wird an- bzw. ausgeschaltet.  
  
 LittleWin                Schaltet den Iconify-Modus auf Little Window  
  
 AppItem                 Schaltet den Iconify-Modus auf AppItem  
  
 AppIcon                 Schaltet den IConify-Modus auf AppIcon  
  
 Unpacked                Der CrunchMode wird neu gesetzt  
 Pppacked                Der CrunchMode wird neu gesetzt  
 LHpacked                Der CrunchMode wird neu gesetzt  
 XPKpacked               Der CrunchMode wird neu gesetzt  
 CRMPacked               Der CrunchMode wird neu gesetzt  
  
 Overwrite <On/Off>     Overwrite wird an- bzw. ausgeschaltet.  
 AutomaticSave <On/Off>    AutomatikSave wird an- bzw. ausgeschaltet.  
 SSDir <On/Off>         SSDir wird an- bzw. ausgeschaltet.  
 Protect <On/Off>        Protect wird an- bzw. ausgeschaltet.  
 PBits <On/Off>         PBits wird an- bzw. ausgeschaltet.  
 SampleMode <On/Off>     SampleMode wird an- bzw. ausgeschaltet.  
 SaveAsPt <On/Off>       Save As ProTracker wird an- bzw. ausgeschaltet.  
 XPKPackMethod <Mode>    Stellt den XPK-CrunchMode ein.  
  
 NoFilerequester        Der FilerequesterMode wird neu gesetzt  
 Req.Library             Der FilerequesterMode wird neu gesetzt  
 ReqTools.Library       Der FilerequesterMode wird neu gesetzt  
 ASL.Library             Der FilerequesterMode wird neu gesetzt  
  
 Volume <Zahl>          Setzt die Volume auf den angegeben Wert.  
  
 Balance <Zahl>         Setzt die Balance auf den angegeben Wert.  
  
 SetSpeed <Zahl>        Setzt den Speed auf den angegeben Wert  
  
 Voice <Nr> <Zustand>    Schaltet die entsprechende Stimme an bzw. aus.  
  
 Quit                    Der Eagleplayer wird beendet.  
  
 Status <group> <dependent>

Gibt den aktuellen Status zurück. Dabei wird zwischen folgenden Gruppen unterschieden: - Globale Einstellungen (g),  
   - Modulabhängige Einstellungen (m)  
   - Playerabhängige Einstellungen (p)  
   - SampleInfo (s)        (nicht eingebaut !!!)  
   - UserProgramme (u)

Je nach Gruppe müssen evtl. weitere Argumente übergeben werden. Im folgenden ist eine Übersicht aller Status-Kommandos des Eagleplayers zu sehen.

Die globalen Einstellungen:

---

```
-----
status <group> <dependent> ;<group> = g
      ;<dependent> = <cmd>
```

```
      <cmd>      Bedeutung und Returnwert
-----
```

```
ver  Eagleplayer Version      Zahl
dve  Eagleplayer Version      Zahl
dir  Playerverzeichnis        String
udi  Userprogrammverzeichnis   String
fon  derzeitigen Font        String
hep  HelpPfad                 String
ins  Instrumentenverzeichnis  String
sap  SavePfad                 String
pbp  PlBatchPfad              String
ubp  UPrgBatchPfad           String
pke  HotKey                   String
num  Playeranzahl             Zahl
ico  Iconify on               <Yes|no>
mwi  Main-Window offen       <yes|no>
pli  Playerwindow offen      <yes|no>
vol  derzeitige Lautstärke    Zahl
bal  derzeitige Balance       Zahl
spe  derzeitiger Speed        Zahl
ply  Song spielt              <yes|no>
hel  HelpModus                <yes|no>
hef  Helpfile vorhanden      <yes|no>
pub  derzeitiger PubScreen   String
mau  Max. Userprogramme       Zahl
usp  geladene Userprogramme   Zahl
snd  aktuelle SongNummer      Zahl
mil  Anzahl der Module in der Liste Zahl
epb  Eagleplayer Globals      Zahl
vo1  Voice1-Zustand           <yes|no>
vo2  Voice2-Zustand           <yes|no>
vo3  Voice3-Zustand           <yes|no>
vo4  Voice4-Zustand           <yes|no>
eab  EP-Batch-Zustand         <yes|no>
upb  UPrg-Batch-Zustand       <yes|no>
ejp  EjectPlayer-Zustand     <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-Zustand           <yes|no>
fid  Fadein-Zustand           <yes|no>
fad  Fadeout-Zustand          <yes|no>
aps  AutoPasswort-Zustand     <yes|no>
ldd  Load Dir-Zustand         <yes|no>
aud  Auto Dir-Zustand         <yes|no>
sdn  Show Dirnames-Zustand    <yes|no>
hia  HideAll-Zustand          <yes|no>
```

```

uep  Use EPDir-Zustand      <yes|no>
not  Notify-Zustand        <yes|no>
alw  Scan Always-Zustand   <yes|no>
mme  Größe des menuMems    Zahl
chd  Betrag der Checkdifferenz Zahl
epm  Entries per Menu      Zahl
end  Songend-Zustand       <yes|no>
lbf  Load Before-Zustand  <yes|no>
law  Load Always-Zustand  <yes|no>
qst  Quickstart-Zustand   <yes|no>
rns  RandomSong-Zustand   <yes|no>
ass  AutoSubSong          <yes|no>
tim  TimeOut              Zahl
ovw  Overwrite-Zustand    <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  aktueller IconifyMode String
frm  aktuelles Filerequester String

```

der Moduleabhängige Einstellungen:

---

```

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

```

<cmd>      Bedeutung und      Returnwert

---

```

pat  ModulePath           String
dir  Moduleverzeichnis    String
fil  Modulename           String
pnr  Nr des aktiven Players Zahl
num  Position in Modulliste Zahl
min  max Anzahl SubSongs  Zahl
max  max Anzahl SubSongs  Zahl
aut  Authorname           String
son  Songname             String
len  PatternLength        Zahl
dpt  verschiedene Pattern Zahl
stp  Steps                 Zahl
sam  Samples              Zahl
chk  CheckMod (mod ok?)   <yes|no>
pon  PositionsNr          Zahl
icm  aktuelle Playernummer Zahl
dur  berechnete Spielzeit Zahl
siz  gesamte Länge des Moduls Zahl

```

Playerabhängige Einstellungen (p)

---

```
-----
status <group> <dependent> ;<group> = p
      ;<dependent> = <playernum> <cmd>
```

```
      <cmd>      Bedeutung      und Returnwert
-----
```

|     |                      |                                 |
|-----|----------------------|---------------------------------|
| nam | Playername           | String                          |
| cre | Creatorname          | String                          |
| pin | PlayerInfo           | String                          |
| ver | Versionsnummer       | Zahl                            |
| vol | Lautstärke           | veränderbar <yes no>            |
| bal | Balance              | veränderbar <yes no>            |
| jmp | Patternjump          | verfügbar <yes no>              |
| pos | GetPositionsNr       | <yes no>                        |
| spd | Playspeed            | regelbar <yes no>               |
| sub | Subsong              | verfügbar <yes no>              |
| cfg | Player               | konfigurierbar <yes no>         |
| pck | Supports Packed Mods | <yes no>                        |
| upr | Userprogrammunterst. | <yes no>                        |
| csa | Config savbar        | <yes no>                        |
| clo | Config loadbar       | <yes no>                        |
| msa | Module savebar       | <yes no>                        |
| mii | ModuleInfo           | möglich <yes no>                |
| sai | SampleInfo           | möglich <yes no>                |
| msm | Max. Samples         | Zahl                            |
| msy | Max. SynthSamples    | Zahl                            |
| mle | Max. Length          | Zahl                            |
| mpa | Max. Pattern         | Zahl                            |
| mst | Max. Steps           | Zahl                            |
| mso | Max. SubSongs        | Zahl                            |
| rst | Player               | unterstützt Restartbar <yes no> |
| sge | Player               | unterstützt Songende <yes no>   |
| voi | Player               | unterstützt Voices <yes no>     |
| plf | Player               | unterstützt PlayFaster <yes no> |
| mch | Module change        | <yes no>                        |
| mrs | Module restore       | <yes no>                        |
| dat | Creator-Date         | String                          |
| tgl | Tagliste des Players | Zahl                            |

Die userprogrammabhängigen Einstellungen

```
-----
status <group> <dependent> ;<group> = u
      ;<dependent> = <UserNr> <cmd>
```

```
      <cmd>      Bedeutung      und Returnwert
-----
```

|     |                       |        |
|-----|-----------------------|--------|
| nam | Name des UserPrgs     | String |
| abo | About Userprogramm    | String |
| cre | Creator des Userprgs  | String |
| ver | Version des Userprgs  | Zahl   |
| rev | Revision des Userprgs | Zahl   |
| eve | min. EP-Version       | Zahl   |
| dat | Creator-Date          | String |

---

```

kis  Kickstart des UserPrg    Zahl
prz  Prozessname des Userprgs String
win  Window offen           <yes|no>
con  Config des User-Prgs    Zahl
kox  X-Koordinate des Windows Zahl
koy  Y-Koordinate des Windows Zahl

```

## 1.81 anpassung an directory opus

### 7.3 Die Arexxanpassung an Directory Opus

---

In diesem Kapitel soll beschrieben werden, wie Sie Arexx-Scripts, die zur Steuerung des Eagleplayers dienen an den Opus anpassen. Dazu wurden eine Reihe von Scripts mitgeliefert. Unsere Arexx-Scripts sind alle an DOpus4 angepaßt (ArexxPortname des ersten geöffneten Opuses: "DOPUS.1"). Beachten Sie bitte, daß der DOPUS3 einen anderen Arexx-Portnamen verwendet ! Zum Anpassen der ArexxScripts kopieren Sie diese bitte in Ihr assigntes REXX: Verzeichnis. Danach starten Sie das Configurationsprogramm des Opus. Wählen Sie bitte das BUTTON-Gadget an und klicken Sie dann auf eins der freien Gadgets die dann erscheinen. Am Besten ist, Sie öffnen einen eigene Bank (New Bank anklicken), denn Sie können mit den Arexx-Kommandos locker die komplette Bank belegen.

So jetzt erscheint der Button-Edit-Screen. Geben Sie bitte den Namen des Buttons ein ("LoadModule"). Betätigen Sie Gadget "New Entry". Es werden einige Gadgets freigegeben und ein Cursor erscheint. Geben Sie bitte folgende Zeile ein "EP\_LoadModules.dopus" ein. Stellen Sie das linke Gadget auf "Arexx". Betätigen Sie nun "Okay" bis Sie wieder im Opus sind. Selektieren Sie mehrere Module und klicken Sie danach das LoadModule-Gadget.

Dieser Arexx-Script wertet alle angeklickten Namen auf und übergibt diese dem Eagleplayer, der wiederum das erste davon lädt und den Rest in sein Pull-Down-Menu übernimmt. Ist nur ein Module selektiert wird im Eagleplayer das ganze Directory geladen. Falls kein Eagleplayer geladen ist, erledigt dies der Arexx-Script, aber ändern Sie den Pfad des Eagleplayers im Script so, daß der Player auch gefunden wird.

Möchten Sie ein Module per Doppelklick laden, müssen Sie einen Filetype definieren. Gehen Sie dazu bitte auf das Gadget "Filetypes". Nun wählen Sie aus dem PullDown-Menu den Punkt "New" an. Dort wo der Cursor erscheint geben Sie "SoundModule" und im nächsten Stringgadget "Music" ein. Die Eingaben sind zur Orientierung und haben sonst keinerlei Bedeutung. Wählen Sie nun "New Entry" an. Betätigen Sie die Lupe und klicken Sie "Matchname" an. Dort wo der Cursor erscheint geben Sie ein "MOD.\*". Drücken Sie Enter und gehen wieder auf "New". Diesmal klicken Sie "or" an und dann wieder auf "New Entry" gehen. Nochmals "Matchname" betätigen und folgendes eingeben: "CU.\*". Drücken Sie "Okay".

In dem diesem Window drücken Sie auf "New". Es erscheint ein Window in dem Sie das Wort, welches wir vorhin bei File\_Class eingegeben haben("SoundModule"). In dem oberen TeilWindow "Events..." klicken Sie "Double-Click" an. Bei Event Discription geben Sie den Text an, den der Opus bei Aktiviation Ihrer Routine in die Titelleiste des Screens schreiben soll. Also geben Sie ein "Now loading selected Soundmodule ". Betätigen Sie "New Entry" und geben Sie ein

---

"rx rexx:EP\_Loadmodule.dopus". Stellen Sie das linke Gadget auf AmigaDos und betätigen Sie "Okay".

Wenn Sie Ihrem Filetype höchste Priorität geben wollen, müssen Sie den Namen im nun erschienenen Window nach ganz oben "swappen".

Sie haben jetzt eine FileClass geschrieben, d.h. alles was per Doppelklick angewählt wird und mit "MOD." oder "CU." anfängt, wird von Ihrer Class erkannt und an den Eagleplayer weitergeleitet, der falls es möglich ist, das Module lädt und abspielt.

## 1.82 danksagungen

### 8. Danksagungen

-----

Als erstes gilt unser Dank natürlich Delirium, von denen wir das Grundgerüst unserer externen Replayer übernahmen. Auch für die Testmodule, mit denen uns Frank Riffel weiterhalf möchten wir uns bedanken.

Als Zweites müssen wir uns bei den anderen Mitgliedern unserer Gruppe für Unterstützung mit Grafiken (M.U.D.U., Ramon), der Programmierung der Eagle-intros (Crazy Copper) und Anregungen bedanken.

Zum Dritten geht ein besonderer Gruß an Turbo/Infect, der uns neben den aktuellsten Versionen seines Rippers mit Tools/Modules/Sources und Tips aus/weiterhalf.

Nochmals ein Dankeschön an Turbo für die Längenberechnungen von verschiedenen Soundsystemen für die Save-Funktion !!!!

Wir danken allen registrierten Usern für die bezahlte Shareware und die zahlreichen Bugreports, Hinweise und Vorschläge. Bitte versteht, daß nicht gleich alles eingebaut werden kann.

Ein besonderer Dank flattert zu Oliver Borrmann (der 1. registrierte User) für die umfangreichen Vorschläge, Bugreports, Module und das InStereo2.0-Demo, wofür ich nun endlich eine Playroutine habe !!!

Besten Dank auch Manfred Jacobi und Michael Wesp für die zahlreichen Vorschläge, Bugreports und Pics.

Bei Daniel Lars Reuß muß ich mich für die Bugreports in Sachen Protracker und Startrekker bedanken, auf die ich schwerlich allein aufmerksam geworden wäre (wer saved auch Startrekker ohne NT file und will da noch Songend :)--== )

Bei MC Noise of PRIME DESIGN (ja, ich betone das nochmal ausdrücklich, daß MC Noise in PRIME DESIGN und nich woanders is) für den Song unseres Intros.

Dank auch an die Autoren von XPK, deren Paket echt eine Wucht ist, nur der XFH könnte noch verbessert werden (warum packt das dumme Teil doppelt, wenn die Datei schon XPK-gepackt war ??)

Ein weiterer Dank gilt Capitain Bifat (TEK) für seine umwerfenden Vorschläge. (vor allem die neuen Oberflächen).

---

Danke auch an Thomas Schwarz für seinen CRM und die Dokumentation der Library.  
Nur ! Wieso ist auch die LIB registriert ????????

Thomas Winischhofer möchten wir natürlich auch danken, vor allem für seine  
angepaßten Playroutinen sowie kritischen Hinweise zum Eagleplayer

Wir möchten uns auch bei Braintrace Design für den Sonic-Arranger und den  
Instereo2.0 bedanken, die uns bei der Entwicklung unserer Abspielroutinen  
ziemlich weiterhalfen.

Weiterhin möchten wir Michael Sinz für seinen Enforcer danken, der uns auf  
einige bööse Fehler aufmerksam machte.

M. Praschel, R. Theuerkorn und Crazy Copper danken wir dafür, daß wir den  
Player auf ihren A4000 testen konnten.

Dank sei auch an unseren Modemtrader nach Schwerin gesendet, der unseren Player  
immer schön verteilt.

Ein Gruß auch an Crazy Crack of Complex, der unsere eigentlich gut gemeinte  
Kritik in der alten Doc gar nicht so mochte, aber immerhin hat er kapiert, daß  
sich selbstmodifizierender Code doch nicht so gut macht.

Nichts für ungut, aber wir benutzen auch beim TP3 unsere interne, eigene Play-  
routine. Friendship rulz ! Frieden ? :)--==

Ein weniger netter Dank gilt Stefan Stuntz, dafür, daß er mir auf meinen  
seitenlangen Brief mit Bugreports und Anregungen keine Antwort gab, obwohl  
ich registrierter User des MFR bin. Trotzdem ist das Proggy recht gut, mal  
ganz abgesehen von MUI. (Der Typ scheint echt Langeweile zu haben)

Ein netter Gruß auch an Ron Birk, der den einmaligen Playsid programmiert.  
Danke im Vorraus für Playsid-Eagleplayer.

Dank auch allen Beta-Testern, die noch nicht erwähnt wurden, Sorry !!!

Ok, Leute das wars !!! Viel Spaß wünscht Euch das Eagleplayer-Programmierteam.

Eagleeye & Buggs of DEFECT

PS: Der Eagleplayer kann intern SoundTracker und ProTracker abspielen sowie  
einige Customsongs und SoundTracker, die als Songs vorliegen, aber auch  
Propacker1.0 und Prorunner, nicht zu vergessen der unwahrscheinlich oft  
genutzte Wantonpacker und natürlich Noisetracker, aber auch Startrekker und  
AudioSculpture und ProTracker und OldSoundtracker sowie UnicTracker und  
ProTracker sowie einige ProTrackerPacker. Darunter wären Unic-Tracker,  
ProRunner und natürlich nicht zu vergessen der der ... ehm Wantonpacker und  
der Prorunner sowie natürlich die gesamten Soundtracker, die da wären  
Frei nach Helge Schneider !!!!!!!!!!!!!!! :-)))))))))

## 1.83 registration









## 8. Eagleplayer - Biographie

~~~~~

V1.53

- Bug im "The Player 4.x" beseitigt, durch den Abstürze auf 68000er Maschinen hervorgerufen wurden
 - selbiges bei Promizer 2
 - Spielzeitberechnung noch einmal überarbeitet, nun fast 100% richtige Werte, sogar Module wie "MOD.A shade in Waiting" werden richtig behandelt
 - komplett neuer Fastram Replayer für die internen Tracker, der nur noch 1 kByte Chipmem braucht und dabei genauso schnell ist, wie der alte
 - ein Fehler in der Zufallsroutine sorgte für gelegentliche Enforcer-Hits
 - PS3M kam als externer Player dazu, der Screamtracker3, Multitracker-Module und Fast/Taketracker verarbeiten kann mit umfangreicher Konfiguration, Analyzer Unterstützung und Patternscroller Unterstützung (vorerst nur Multitracker/S3M)
 - Moduleinfo/Patternsroll und Space-Scope+ wurden für Kickstart 1.3 umgeschrieben. Zum Betrieb ist allerdings die Gadtools13.library erforderlich. Diese liegt nicht dem Paket dabei, da ich deren Vertriebsbedingungen nicht kenne.
 - es können nun auch mit Stonecracker 4.04 gepackte Module entpackt werden (ohne die xfdmaster.lib zu bemühen)
 - Unterstützung von Hardlinks (bei Nutzung des Pulldown-Menüs)
 - Unterstützung von Dircaching bei Reqtools eingebaut
 - die Kick 2.x+ Versionen der Patternscroller sind nun ziehbar und unterstützen nun bis zu 16 bzw. 32 Stimmen, weiterhin wurde die auch beim Protracker existierende "Blankzero"-Option, die für mehr Übersicht sorgt, implementiert
- Release: Doomsday Party Karlsruhe, 9.10.1994

Aufgrund eines Versehens wurde das "β"-Zeichen nicht entfernt, es handelt sich aber um die voll lauffähige Version.

Bei der Release ohne das Beta-Zeichen (Internet) wurde die Gelegenheit genutzt, um den Eureka-Packer Player zu überarbeiten, außerdem wurde der Vectordean wieder für 68000er Maschinen nutzbar gemacht. Fast- und Taketracker Module werden nun vollständig inclusive Patternscroller und Moduleinfo unterstützt. Die Erweiterung des "Speed"-Kommandos des Protracker Players wurde rückgängig gemacht, da diese - eigentlich zum Beheben von Problemen, durch falsch genutzte Kommandos in Modulen gedachte - Funktion nun mit richtig eingespielten (Extrem)-Modulen Probleme bereite. Wenn nun Probleme mit Modulen wie "MOD.Face another day" (Heatbeat/Virtual Dreams) auftreten, sollte die "Tempo berichtigen" (Fix BPM)-Funktion des Noiseconverters genutzt werden. Diese Probleme erscheinen dann allerdings auch bei anderen Playern, die Protracker voll unterstützen.

V1.52b

Einige kleine Fehler in diversen Userprogrammen wurden entfernt, Trackerpacker 3.0 kam dazu, außerdem wurde in der Spielzeitberechnungsroutine ein Check eingebaut, der eine Endlosschleife mittels Patternrepeat erkennt. Ehm.. Außerdem wurde ein Bug beseitigt, durch den die CRM.library nicht geschlossen wurde. Weiterhin wurde der XANN-Packer Replayer verbessert und funktioniert nun auch mit dem Song von einem Liquid Trackmo. Der Fastram-

player wurde nochmals verbessert. Es wurde die Randomplay Routine noch einmal komplett neu geschrieben, da ab und zu unter bestimmten Bedingungen noch Fehler auftraten. Außerdem sollte die Streuung nun weit besser sein, so daß auch bei großen Modullisten die hinteren Module nicht "vergessen" werden. Als Player kam nun noch der "MellicaPacker" dazu, ein altes Kefrens Format. Ein Fehler, der nur in der registrierten Version auftrat, wenn man bei eingeschaltetem "load before" den Fastram-player ausschaltete, wurde beseitigt.

!!! Achtung Achtung !!! Der Eagleplayer besitzt nun genau 100 unterstützte Musikformate !!! Zeigt mir den Player, der mehr Musikformate abspielen kann !

Release Date: Ende Juni 1994

V1.52

Als erstes wurde eine neue Timingroutine eingebaut, die unabhängig von der Zeilenfrequenz arbeitet. Erfolg: Es wurden bei umfangreichen Tests auf dem A4000/40 in allen Auflösungen keine Probleme mit Protrackermodulen gefunden. Als nächstes wurden einige kleine Fehler in den Userprogrammen ausgebaut, durch die der Speicher nicht komplett freigegeben wurde. (~3.5 k) Es wurden im Eagleplayer einige neue interne Player eingebaut. Dabei handelt es sich um "Noisepacker 2.x", "Noisepacker 3.x", "EUREKA-Packer" "The Player V5.0A/6.0A". (Mehrlänge des Eagleplayers: ~6.5k !!!)

Weltneuheit !! Als erster Soundplayer hat der Eagleplayer eine echte Patternsrollfunktion ! Funktioniert mit allen internen Trackern, sowie Soundtracker2.6-Icetracker (wurde zu einem Player zusammengefaßt und spart 4.5k), IFF-EMOD (Quadracomposer), SonicArranger und Oktalyzer bis 8 Stimmen !!

Die Anzeigefunktion des aktuellen Verzeichnisses im Menü wurde überarbeitet und zeigt nun auf einem normalen Hires Overscan Screen (724*283 Pixel) weit über 500 Einträge an.

Als nächstes kam der "ProTracker FASTRam" Abspieler hinzu, der Protrackermodule ins Fastram lädt, soweit nicht mehr Chip-Pufferspeicher benötigt wird, als im Normalfall. So ist es beispielsweise möglich auf einem Rechner mit nur 1MB Chipmem locker irgendwelche 2MB Module abzuspielen. (wenn entsprechend Fastmemory vorhanden) Es wurde endlich die "Moduleinfo"-Funktion fertiggestellt.

Das Quadrscope wurde um einige neue Modi erweitert und besitzt nun 4 verschiedene Anzeigemöglichkeiten. Außerdem arbeitet es jetzt mit der "Mastervolume"-Funktion des Eagleplayers zusammen. Auch Space-Scope wurde noch einmal verbessert. (teilweise schon mit der V1.51 ausgeliefert)

Der TFMX 7V Replayer wurde neu geschrieben und arbeitet nun auf A4000.

Für das Moduleinfo kam eine Berechnung der Spielzeit von Protracker-SonicArranger- und Oktalyzermodule auf die Sekunde genau dazu. !!! WELTNEUHEIT !!!

(Wer kein registrierter Benutzer des Eagleplayers ist, kann diese Funktion immerhin über den Noiseconverter testen. Dieser ist jetzt auch in der Lage, P50 und P60 zu erkennen und zu entpacken.) Außerdem wurde ein kleines Facelifting der Benutzeroberfläche durchgeführt.

Aus lauter Langeweile wurden auch noch ein paar auf OS3.0 zugeschnittene Userprogramme dazugepackt, die allerdings mindestens einen 16-farbigen Screen voraussetzen.

Das Space-Scope wurde noch ein wenig Geschwindigkeitsoptimiert und arbeitet nun mindestens 20% schneller.

Für die allerliebsten Cracker wurde der Eagleplayer intern noch einmal kräftig überarbeitet und geschützt. Viel Spaß damit !!
Wer es als erstes schafft, den Eagleplayer 100% zu cracken, bekommt eine Registration gratis !
Release Date: 18.04.1994

V1.51 QUICK UPDATE

Es wurde im Eagleplayer ein Fehler ausgebaut, weswegen er nicht auf dem A4000/40 im CopyBack Modus lief. (Eigentlich wurde dieser Fehler schon mal ausgebaut, aber es gab wohl Trouble mit dem Quellcode.) Weiterhin kam der externe Replayer "Aprosys" dazu. Die nächste Änderung betrifft die Installationsdatei, die überarbeitet wurde. Als nächstes kam das Useprogramm "Space-Scope" hinzu, welches allerdings auf unbeschleunigten Maschinen einen Großteil der Rechenzeit beansprucht.

Der Noiseconverter kann nun auch Promizer 2.0 und MEXX-Packer laden.

Mexxpacker ist jetzt ebenfalls ein interner Replayer. Die Random-play Routine wurde ebenfalls überarbeitet. Einige kleine Fehler in der Showdirname Funktion, Stimmenumschaltung und bei Fehlermeldungen wurden entfernt.

Release Date: ab Anfang Januar 1994

V1.50

Der Eagleplayer wurde intern umfangreich umgebaut. Es wurden im Main- und im Playerwindow Tastaturkürzel eingeführt. Die Gadgets des Playerwindows wurden endlich geändert. Die internen Soundtrackercheckroutinen mußten nochmals überarbeitet werden. Die Songerkennung arbeitet jetzt auch besser. Weiterhin sind ein paar Bugs in den Soundtrackerplayroutinen beseitigt. Eine Konvertierungsroutine von etwas älteren Soundtrackermods auf den Protracker-Standard wurde eingebaut. Somit wird teilweises Piepen verhindert.

Die LoadDir-Routine wurde komplett überarbeitet und arbeitet jetzt sauberer. Wenn nun der private Dirmem voll ist, stürzt der Player nicht mehr ab. Er übernimmt dann sovieler Einträge, wie hineinpassen und zeigt diese an. Zudem kann übers Pull_Down-Menu das Directory ausgewählt werden. Beim AppWindow können Files unnd Dirs aus verschiedenen Direktories übergeben werden.

Beim Playerladen ist jetzt Multiselect möglich. Neu ist außerdem, daß die externen Playroutinen erst geladen werden, wenn sie benötigt werden sowie ein Entfernen derselben bei "Eject Module". Somit wird der Speicherverbrauch und das Warten beim Start des Eagleplayer in Grenzen gehalten.

Der Eagleplayer kann nun von fremden Programmen teilweise gesteuert werden (NoiseConverter, Exotic-Ripper <- immer noch nicht fertig, Turbo ist OBERFAUL !).

Von registrierten Usern kann die Funktion "Load Before" genutzt werden. Das Einbinden des Hotkeys funktioniert jetzt besser. Als Default-Hotkey ist nun "Ctrl Alt E" eingetragen. Die Richtlinien für ein Commodity werden jetzt besser befolgt. Es wurde ein "Hide" Commando im Pull-Down-Menu hinzugefügt.

Neu ist auch der umfangreiche Save-Modus. Man kann zwischen Normal und AutomaticSave auswählen. Ein Funktion zum setzen des Save-Dirs existiert auch. Die Module können jetzt gepackt werden.

Dabei kann teilweise zwischen Samplemode und Protection unterschieden werden. Wir haben jetzt den Powerpacker-SampleMode entwickelt (Packt sehr gut). Die Protectionbits für gespeicherte Module können angegeben werden. Der XPK-Packmode kann selbst gewählt werden.

Die Crunchmania SampleDecrunchroutine wurde (über library) eingebaut.

Die interne Decrunch-Routine wurde entfernt, weil keine SampleMode und keine Passwort-Unterstützung vorhanden war. Der Eagleplayer kann jetzt Crm in allen Varianten packen. Beim Packen wird in der Statusanzeige ein Balken gezeichnet, der angibt, wie weit der Packer ist. Der Pubscreename, auf dem der Eagleplayer sich befinden soll, kann jetzt von der Oberfläche aus eingegeben werden.

In der Preference sind die Punkte: Autopassword und QuitEagle hinzugekommen.

Der Eagleplayer arbeitet jetzt endlich mit RTPatch zusammen. Der Eagleplayer scrollt nun auch, wenn die diskfont.library nicht gefunden wird.

Es gibt jetzt ein "Warning Menu". Es kann eingestellt werden, ob zB. ein Warnrequester erscheinen soll, wenn das Module zu kurz, zu lang oder kaputt ist.

Fast alle externen Replayer wurden überarbeitet und sind an den neuen Eagleplayer-Standart angepaßt und somit nicht mehr delitracker-kompatibel. Der MusicMaker-Mods können jetzt auch gepackt werden. Ab dieser Version sind jetzt endlich englische Dokumentationen verfügbar.

Es wurde ein Fehler bei LoadConfig behoben, der vor allem bei Rechnern mit 32 BitRam auftritt.

Es kam ein weiteres SubMenu hinzu: Modules-Directory. Damit läßt sich die Prefs des Pull-Down-Menus einstellen. Weiterhin kann der interne Speicher für den DirPuffer gesetzt werden. Notify wird ebenfalls genutzt.

Ebenfalls eine Neuerung stellen die verschiedenen Filerequester dar. Es wird nur die benötigte Library geöffnet. Die Arp.Library wird trotz Planung nicht implementiert. Der Eagleplayer benötigt jetzt keine Library mehr, da ein FileReqMode "No FileReq" existiert, d.h. der Filename kann jetzt direkt im Statuswindow eingegeben werden.

Es kann eingestellt werden, daß der entsprechende externe Eagleplayer erst geladen wird, wenn das dazugehörige Module angewählt wird. Die Erkennung wird anhand des Prefixes vorgenommen, der in einer EPBatch Datei definiert wird und beim Module im Namen enthalten ist. Diese Funktion ist ebenfalls für Userprogramme verfügbar (r.V).

Userprogramme können nun auch entfernt werden. Die Anzahl der max. ladbaren Userprogramme ist zur Zeit auf 32 begrenzt.

Als Userprogramm ist nun eine Listenverwaltung verfügbar.

Der Eagleplayer kann jetzt auch LHA-Archive entpacken. Dabei wird zur Zeit nur der erste Eintrag geladen.

Release Date: 29.12.93 auf der "THE PARTY III" Herning/Dänemark.

V1.40 Der Eagleplayer unterstützt nun ein eigenes FastDir namens ".EPDIR", dadurch wird erreicht, daß das Dir schneller geladen wird. Weiterhin wurde eine Funktion "SaveT" eingebaut. Sie ermöglicht es, gepackte Module, die eigentlich nicht gepackt werden dürfen, doch geladen werden können. Der Eagleplayer hat nun auch eine Hotkey-Funktion. Bis auf das Hotkey sind alle neue Funktionen nur für registrierte User zugänglich.

Es wurden folgende neue Replayer hinzugefügt: LME, MTH, IFF-SMUS IFF-8SVX, Music-Maker 4+8, Rob Hubbard und SoundControl.

Die sich modifizierenden Replayer werden zum Umbauen Ihrer Playroutinen nun zunehmend die internen Routinen "EPG_ModuleChange" und "EPG_ModuleRestore" nutzen. Weiterhin wurden die Unterprogramme "EPG_Hexdez" und "EPG_TypeText" eingebaut.

Die Funktion "Quit Surface" wurde hinzugefügt. Damit kann bei der

Beendigung eines User-Programms der Eagleplayer beendet werden.
Der Fehler beim Configladen (AppIcon) wurde beseitigt. Ein Bug in der
VolVoice-Routine wurde beseitigt.
Release Date: 20.09.1993

V1.29b Ein gepatchter Eagleplayer, der nicht von uns kommt. Es handelt sich
um einen PP-gepackten und wieder entpackten Eagleplayer. Da der Power-
packer den Code beim Entpacken modifiziert, stürzt dieser Player nur
ab. Außerdem wurden alle "V1.11"-Kennungen durch "V1.29" ersetzt und
eine total schwachsinnige Doc, sowie einige sch.. Module auf eine
3. Disk kopiert.

V1.11 Die erste für registrierte User herausgegebene Eagleplayerversion
mit Keyfile. Registrierte User können jetzt die Save-Funktion,
die HelpFunction sowie die anderen Userprogramme nutzen.
Es wurden 3 neue Replayer geschrieben, Player V5.0A, Syntracker, Major
Tom's Player. In den Replayern SIDMonII, ActionAmics, Noisepacker,
Prorunner V1.0/V2.0, Synth & InStereo wurden Fehler beseitigt. Der
Prorunner Vx.x wurde in NoiseRunner umbenannt. Fast alle ab V1.11
bearbeiteten Player werden nur noch ab Eagleplayer V1.11 und nicht
mehr auf dem Delitracker laufen. Der Grund dafür liegt in der neuen
VolVoiceroutine, die es erlaubt, die Lautstärke jeder einzelnen Stimme
zu steuern. Der Programm-Mode wurde um Randomsong, AutoSubSong und
LoadAlways erweitert. Alle neuen Funktionen werden nur noch für
registrierte User nutzbar sein.
Die Configuration der Userprogramme wurde nochmal überarbeitet. Ein
großer Bug, der an der Intuition.Library liegt, wird nun umgangen. Der
Fehler trat auf, wenn kein Userprogramm geladen wird, dann ist das
Pull-DownMenu "Special-->Userprogramme" nur mit dem Item Add Userprg
belegt. Normalerweise ist eine 2. IText-Strukture angegeben, die unter
dem Text "Add Userprg" eine Wellenlinie zeichnet. Diese wird aber
nicht gezeichnet. Es wird stattdessen im Speicher rumgesaut.
Die Anzahl der in ein Menu aufgenommenen Module wird jetzt von der
Screenhoehe bestimmt. Die Prioritäten der Userprogramme werden nun
richtig gesetzt.

V1.10 Ein Laden und Saven der Config ist nun endlich möglich. Ein Arexx-
Port ist eingebaut mit einem äußerst umfangreichem Befehlssatz.
Es werden demzufolge Beispiel-Arexx-Scripts zur Installation, z.B für
Dopus 4, mitgeliefert.
Der Eagleplayer ist nun ein Commodity. Eine Pubscreen-Unterstützung
ist ebenfalls vorhanden. Zudem kann der Eagleplayer Fonts laden, aber
nur 8*8 und keine proportional Fonts. Der Iconify-Modus wurde auch
fertiggestellt. Nun sind AppItem & AppIcon auch möglich. Eine OnLine
Hilfe ist für registrierte Anwender verfügbar, wenn das File
"Eagleplayer.Help" gefunden wird. Die Tags EP_Date und EP_Check3
wurden implementiert. Nun können z.B ungepackte FTM und Oktalyzer
(8Voices-Mods) ins Fastram geladen werden. Die EP_Globals wurden
umfangreich erweitert. (EPG_LoadFile2, EPG_Loadexecutable,
EPG_Filerequest, EPG_TextRequest, EPG_FindTag, diverse Librarybasen
und wichtige Merzzellen)
Vorsicht: Der Eagleplayer V1.0 testet nicht auf EP_PlayerVersion !!!
Die Eagleplayer-Status-Strukture wurde entfernt, da alle Daten nun
in den EPG_Globals definiert sind. Eine HelpFile-Strukture und eine

GadgetStrukture fürs Textrequester sind im Include-File definiert. Der UserProgrammPort wurde erweitert. Die Message-Class "Command" ist implementiert. Es ist jetzt möglich, eigene Oberflächen zu entwerfen und den Eagleplayer lediglich im Hintergrund als Grundprogramm laufen zu lassen. Die Userprogramme können so den Eagleplayer steuern. Ein Beispiel dieser neuen Möglichkeiten ist das Userprogramm 'NoiseEagle' (ein simpler nachgeahmter Noiseplayer). In den Replacern Actionamics, Sidmon II, Player 4.0a/4.0b/4.1a, Propacker, Oktalyzer wurden Bugs beseitigt. Die Erkennung der Autoren wurde verbessert und kann jetzt von externen Replacern genutzt werden. Ein peinlicher Bug in der SetSpeed-Routine wurde beseitigt (die Parameter wurden falschrum übergeben). Die Anzeige der playertypischen Eigenschaften im Playerwindow funktioniert jetzt richtig. Früher stürzten manchmal die Userprogramme beim Verlassen des Players ab, sollte nicht mehr passieren. Einige Bugs in der LoadModule-Routine und der Change-Voice-Routine wurde removed. Tag-Skip wird jetzt auch genutzt. Der Hippel-Replacer spielt jetzt auch Module, bei denen das Init-Bit nicht gesetzt ist. Ein Bug im JamCracker wurde beseitigt. Die Player Promizer 4.0 & ChaosEngine hinzugefügt. Die CheckRoutinen der interne Soundtracker wurde nochmals überarbeitet. Die Fehlermeldungen im Include-File wurden umbenannt um sie zu vereinheitlichen. Ein Resource-Symbol File zum disassemblieren von Eagleplayern wird mitgeliefert.
Release Date: 10.07.93 auf der Sun'n'Fun Party Hellental

V1.01 Mehrere Bugs in der LoadRoutine und in der OpenWindow-Funktion wurden beseitigt. Die DTG_Globals wurden um DTG_WaitAudioDMA erweitert, so daß der Eagleplayer jetzt Delitracker V1.37 kompatibel sein sollte. Die Programmierrichtlinien der Replacern Anpassung wurden erweitert. Der Eagleplayer wurde erstmals Enforcer getestet und siehe da, es kamen einige Bugs ans Tageslicht. Die internen Checkroutinen der internen Soundtracker wurden nochmals überarbeitet und der Startrekker stürzt hoffentlich auch nicht mehr ab. Zur Eagleplayer-Package gehören nun auch der Promizer 0.1 und der Promizer 2.0. Grouleff wurde endlich an den EP-Standard angepaßt. Fast alle Replacer wurden nochmals bearbeitet und es wurden Enforcer-Hits (ActionAmics, SidmonII, P4xx-Tracker) beseitigt. Weiterhin erkennen jetzt fast alle Replacer SongEnde !
Release Date: 15.05.93

V1.0 Erste freigegebene Demoversion mit vielen Bugs und gesperrten Funktionen. Include-File wird teilweise vom Player noch nicht unterstützt.
Release Date: 24.04.93 auf der Datalive in Dessau

1.87 externe player

Erläuterungen zu den von uns bisher an den Eagleplayer angepaßten Playroutinen

Alle von uns bearbeiteten Player wurden mehrfach getestet. Es kann aber durchaus passieren, daß einige Module, besonders jene mit eingebauter Playroutine, nicht spielen beziehungsweise einige Funktionen des Eagleplayers gesperrt werden, die bei anderen Modulen dieses Soundsystems möglich sind. Bei betroffenen Playern wird dies mit angegeben. Es ist bei solchen gut möglich, daß das Module erkannt wird, aber nicht spielt, das System sogar abstürzt. Solche kritischen Player, bei denen sich die Playroutine von Module zu Module ändert, sind zum Beispiel Hippel und Whittaker.

Falls irgendwelche Probleme mit den Playern auftreten sollten, bitte die

Fehlerbeschreibung

genau formulieren und das problembehaftete Module (wenn alle Module nicht funktionieren reichen 3) mit den Informationen über das verwendete Amigasystem (Kickstartversion, Prozessor, Zusatzhardware wie Grafik- und Soundkarten, Amigamodell) an uns schicken. Wir werden uns bemühen, den Fehler zu beheben und senden dann als Dankeschön die neueste Playroutine zurück.

Besitzen Sie eine Playroutine, die vom Eagleplayer noch nicht unterstützt wird, senden Sie uns diese bitte mit Sourcecode und einigen Modulen zum Test zu, wir werden versuchen diese anzupassen.

In der vorliegenden Version besitzt der Eagleplayer 18
interne Playroutinen

.

Findet er weder das Verzeichnis "Eagleplayers" noch "Deliplayers", so kann so kann er nur ProTracker/Startrekker/OldSoundtracker/Audio-Sculpture & CustomSongs sowie Prorunner,ProPacker,WantonPacker,UnicTracker, GMC und Digital-Illusions abspielen.

Wenn nicht weiter erwähnt, werden Lautstärke, Balance und Songend in der Regel unterstützt, der Player nutzt den internen Timer.

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

ActionAmics: Gerippte PlayRoutine, wird u.a. in dem Game Dynablaster verwendet. Tempo-Bug sowie durch Enforcer entdeckte Bugs wurden removed. ActionAmics nutzt SubSongs, nutzt Patternswitching und unterstützt Volume, Balance, Voices und UserProgramme. SongEnde wird erkannt. ActionAmics ist 99% ein Verwandter von SIDMon II.

ActivisionPro Soundsystem, das vorwiegend in älteren Spielen verwendet wird, auch bekannt unter dem Namen "Wartin Walker".

Die PlayRoutine ist ein einziges Durcheinander (manchmal zwei hintereinander) und mit LoopSchleifen !

Die PlayRoutine befindet sich zudem noch im Module.

Activision Pro unterstützt Analyzer, Volume, Balance, meistens

auch Songende (hängt vom Patternswitching ab).
Die Playroutine müßte Probleme auf höheren Prozessoren haben.

Audio Sculpture Wohl der kommerzielle Nachfolger vom Startrekker. Der AM-Teil der Playroutine wurde kräftig erweitert. Die NT-Files haben allerdings die gleiche Länge wie beim ST1.3 (8336 Bytes). Unterscheiden kann man die beiden Soundsysteme an der Kennung im NT/AS-File. Bei AudioSculpture haben die früheren NT-Files die Kennung ".AS". Der Replayer ist als interne Player eingebaut und kann auch Songs laden. AudioSculpture unterstützt Songende, Patternumschaltung, Volume Balance, UserProgrammansteuerung. Ein Saven des Modules ist nur in der registrierten Version möglich.

ChipTracker Eindeutig ein Noisetrackerformat, arbeitet mit internem Interrupt, unterstützt alle Noisetrackertypischen Fähigkeiten (Patternjump usw.) Das Teil wird überall nur Kris genannt, obwohl es ChipTracker heißt. Wir wissen das, weil wir das Proggy haben. He, Radafi/Effect ! Wie konvertierst Du synthetische Samples des ChipTrackers. Das möchte ich gerne wissen. Ein Saven des Modules ist in der registrierten Version möglich.

Deltamusic1.0: Von Delrium übernommen und kräftig erweitert z.B. Songend, Volume, Balance, Voices, Analyzer ...
Mich wundert nur, daß bei Delta 2.0 nicht auch die Playroutine extern ist, wie hier.

Deltamusic2.0: PlayRoutine im Module. Soundsystem läuft auch auf Kick2.0 und ist sogar PD.
Kritik: Für Synthetische Songs sind die Module zu lang.
Deltamusic unterstützt Analyzer, Volume, Voices... .
SongEnde wird erkannt.
Hinweis:Delta-Module sollten nicht zu kurz gesaved werden, da der Player dann abstürzen droht.

Digital Mugician:
Soundsystem mit coolen Effekten, läuft unter Kick 2.0, ist aber der reinste Hardware-Hack (unterstützt nur df0 !!!).
Playroutine von Delirium übernommen, Ich dürfte aber trotzdem alles alleine gemacht haben. Die Playroutine wurde um 500 Bytes gekürzt und in Fastmem verlagert. Die Anzahl der Subsongs kann

jetzt berechnet werden und Songende wird auch erkannt.
Eine Längenberechnung sowie eine SubSongBerechnung wurde
ebenfalls eingebaut. PatternSwitching ist ebenfalls möglich.
In der registrierten Version können die Module gesaved werden.

DSS Digital Sound Studio, ein Soundtrackerclone als Beigabe zum
GVP Soundsampler. Wenn einer sich nicht unbedingt einen Sampler
kaufen will, lohnt sich das nicht, da der DSS-Tracker so gut
wie keine SpecialFX unterstützt, Protracker ist da besser - und
PD. Ansonsten alle Soundtrackertypischen Anpassungen, die Play-
routine wurde von Delirium übernommen - und trotzdem durfte ich
im Prinzip alles alleine machen, die Typen waren sogar zu faul,
eine Lautstärkeregelung einzubauen - und das bei einem Sound-
tracker.

EMS Eine französische Produktion, leider weiß ich über das Teil so
gut wie gar nichts, die Playroutine habe ich, genau wie mein
einziges Module, von Delirium bekommen. Allerdings gab's im
Deliplayer eine Menge zu tun, da weder Lautstärkeregelung noch
Songend usw. eingebaut waren, und dabei ist die Playroutine
dokumentiert.

FC1.3/1.4 Future Composer V1.0 bis V1.3 Player. Playroutine von Super
Sero of Superions. Bevorzugt in Intros verwendet wegen der
Kürze der Module. Unterstützt werden Lautstärke, Balance,
Songend, Stimmenein-und Ausschalten. Es wird der interne
Interrupt des Eagleplayers verwendet.
Hat irgendeiner eine Idee, wie man Hippel-Module in FC
konvertiert ?

Fred: Und wieder ein Player, wo die Playroutine im Module ist,
(die Leute lernen einfach nicht, daß das Scheiße ist)
Bei mir unterstützt der Fred-Player Volume ...
Eine SubSongRange wurde eingebaut.

FTM Face the Music von Maxon,
das bestklingende 8-Stimmen Musikprogramm für Standard-Amigas,
allerdings auch der schlechteste vorstellbare Support für
Programmierer, keine Playroutine beim Original dabei und dazu
noch ein absolut fies geschützter CLI/WB-Player. Es war eine
Sauarbeit, das Teil als Player zum Laufen zu bringen, jedenfalls
unterstützt der Eagleplayer FTM-Player alles nötige, die inter-

ne Loadroutine des Players wurde entfernt, Module dürfen auch gepackt werden, die Allokierung der Audiokanäle erfolgt über den Eagleplayer und der Hack auf \$70 wurde unter Schweißtriefen ausgebaut. Allerdings arbeitet das Teil immer noch mit eigenem Interrupt.
Ein Saven des Modules ist nur in der registrierten Version möglich.

GMOD MultiPlayer-CustomPlayer
adapted by Delirium. Keine genaueren Informationen vorrätig.
Keine Analyzeransteuerung. Eventuell Volume
(vom Module abhängig).

GMC Ich weiß nicht so recht, was ich von diesem Programm halten soll, keiner arbeitet damit, dafür hatte die Playroutine dringend einige Optimierungen nötig. Sie arbeitet mit internem Interrupt. Ich habe aus Faulheit den Deliplayer übernommen und in Bezug auf die Länge der Reloctabelle erweitert bzw. optimiert, sowie eine neue Checkroutine geschrieben. Die unterstützten Möglichkeiten, siehe Playerinfo-fenster, des Eagleplayers, neuerdings auch Patternumschaltung und Moduleinfo.
Ein Saven des Modules ist nur in der registrierten Version möglich.

Grouleff Ein Soundsystem, das Module mit PlayRoutine im Module saved, die erstaunlich kurz ist. Es wurden alle typischen Anpassungen vorgenommen. Songende wird ebenfalls erkannt.

Hippel Wieder ein Soundsystem, daß die PlayRoutine im Module hat und in dieser Art wohl auch das Bescheuerteste. Da unterscheidet sich wirklich jede Playroutine von der anderen.
Der Hippel-Umbau testet z.Z. die Playroutine des Moduls auf 27 Sub-Routinen, die umgebaut werden müssen, um Hippel analyzerfähig zu machen, ja ihr habt richtig gehört, Hippel ist analyzerfähig (Boah eh) :-) und Songend gibt's als Zuschlag dazu (schabadabadu Mr. Feuerstein) !
Ich bezweifle stark, daß das alle Subroutinen sind. Es gibt bestimmt irgendwo ein Hippel-Module, daß nicht funktioniert.
Hippel ist das Vorbild des Future-Composers !!!
Man erkennt die Ähnlichkeit im Analyzer bei identischen Modulen.
Hippel ist ein sehr altes Soundsystem. Es wurde wie Whittaker vorwiegend in den Jahren 1988-89 benutzt.
Hippel hat Probleme auf dem A4000 !
Hippel sollte jetzt jetzt besser auf schelleren Rechnern

laufen

Hippel-Coso: Bei Hippel-Coso-Modulen handelt es sich um Hippel Module ohne PlayRoutine und das Module beginnt mit "COSO". Der Player versucht ein SampleFile nachzuladen
Es wurde der Delirium-Player verwendet. Die PlayRoutine liegt jetzt im FastMem. Incl. Songend

Icetracker Ein Soundtracker 2.6 Clone, der so gut wie keine Kommandos unterstützt (jedenfalls deutlich weniger als der Noisetracker). Dafür ist das Teil 2.0 kompatibel, was für den Icetracker spricht. Der Eagleplayer unterstützt alle üblichen Funktionen der Soundtracker und seiner Clones. Glücklicherweise habe ich auf der Internet-Package eine Playroutine gefunden, doch -o Spaß, das Teil war teilweise disassembliert. Komisches Teil, muß ich schon sagen. Außerdem verzichteten die Typen sowohl auf den 2. Interrupt, als auch ein DMA-Wait. Wie geht denn das, fragen jetzt versierte Insider ! Ganz einfach, man erstelle 3 Musikroutinen IC_MusicA, IC_MusicB und IC_MusicC und überlasse das Problem dem User, scharf was ?
Ein Saven des Modules ist nur in der registrierten Version möglich.

InStereo/Synth: Es handelt es sich um verwandte Soundsysteme. InStereo war der Vorgänger von Synth. Man kann InStereo-Modules mit dem Synth-Replayer abspielen. Um eventuelle Komplikationen zu vermeiden wurden aber zwei verschiedene Replayer verwendet. Es wurden einige Bugs in den Replayer ausgebaut. Die Player unterstützen ModuleInfo in folgender Form: Songname wird erkannt, Authornamen wird erkannt & eine Längenroutine ist auch eingebaut. Die Anzahl der Subsongs kann berechnet werden und Songend-Erkennung ist auch vorhanden.
Die PlayRoutinen von Synth & InStereo wurden ins FastRAM verlagert.
Vorsicht: Ich habe den Verdacht, die PlayRoutinen gurken irgendwo im Speicher rum !!! (in letzter Zeit weniger :-))))

IFF-SMUS Playroutinen von Delirium übernommen. SMUS-Files können jetzt gepackt werden, wenn SaveT eingeschaltet ist. Weiterhin werden die Instrumente bei EXTLoad geladen und nicht bei InitPlayer, daraus folgt, man kann ein Mod laden, wenn das andere Mod noch spielt. SMUS unterstützt nun auch die Analyzer.

Jamcracker Relativ selten gesehenes Soundformat mit übersichtlicher Playroutine, wird gern mit Synthsamples verwendet. Gegenüber dem Futurecomposer hat das Teil den Vorteil, daß es problemlos unter 2.0 läuft (jedenfalls die mir bekannte Version von XGAG).

LME Gerippte Playroutine aus einem Customsong von Marley/Infect. Wurde eben mal schnell 'n Replayer draus gebastelt.

Major Tom's Player

Playroutine von marley/Infect übernommen und an den Eagleplayer angepaßt. Nur 2 Module vorhanden. Wer kann uns daß Programm zusenden? Der Player unterstützt jetzt Songende, Patternumschaltung, Moduleinfo & Analyser.

Mark-II Eine Soundsystem von Cachet. MII hat einen eigentümlichen Klang und ist, wenn nur ein User-Programm auf ist (bzw. am 4000er mit allen :)), echt lustig anzusehen. OldMark-II wurde ebenfalls implementiert. Songende wird erkannt. Volume,Balance ... Ok.

Martin Walker Wurde entfernt, weil es sich um ActivisionPro-Module handelt.

MCMD Eine gerippte Play-Routine (1 TestModule), die Ähnlichkeiten mit Hippel-PlayRoutinen hat. Auch das Module deutet darauf hin. Ich habe den Player trotzdem erstmal implementiert.

MED Programm von Teijo Kinnunen, entstand wohl aus der Idee, einen Soundtracker unter DOS zu schaffen, kann Soundtrackerfiles laden und saven, in MED umgewandelte Soundtrackerfiles sind zudem kürzer. Dieser Player ist für MED und OctaMED 4-Stimmen Songs. Den Player habe ich von Delirium übernommen, um mich nicht so lange daran aufzuhalten. Bisher habe ich noch keine Möglichkeit gefunden, das Songende zu bestimmen. MED hat eine ekelig lange Playroutine, die einen eigenen CIA-Interrupt einbindet. Auch die Audiokanäle werden vom Player aus besorgt. Der Player besitzt natürlich die Möglichkeiten von Lautstärke- und Balanceregulung, Stimmenein- und Ausschaltung sowie Analyserunterstützung.

Mexx-Packer Eine gerippte PlayRoutine aus einem Mexx-Intro. Die PlayRoutine ist ein einziges durcheinander und selbstmodifizierend wie sau. Dem Coder müßte man das Programmieren verbieten !!!! Der zweite Interrupt wurde entfernt und der selbstmodifizierende Kram ebenfalls. Mexx-Packer unterstützt Songende, Pattern-switching, Volume, Balance, Voices & Analyzer. Ein Saven des Modules ist in der registrierten Version möglich.

ModuleProtector:Modulformat, das nichts so ähnlich sieht, wie dem Propacker V2.1, wenn auch die Patterns nicht gesplitted sind und die eine Playroutine das andere Modul nicht spielt. Verwendet wird das Format gern von LSD (Grapevine#7, Despdir Trackmo). Eine andere Variante haben wir im "Voice Attack II" Musikdemo gefunden. Die 3 Module haben exakt denselben Aufbau, sie lassen sich auch mit der hier genutzten Playroutine abspielen, haben aber teilweise andere Kommandos. Außerdem sind bei der dort verwendeten Playroutine so wenig Kommandos implementiert, daß ich fast behaupten möchte, es handele sich um Soundtracker2.3 oder so was. Die Checkroutine funktioniert halbwegs (jedenfalls mit den uns zur Verfügung stehenden Modulen). Da es aber so viele Varianten gibt, die sich nur in Details unterscheiden, die eine 100%-ige Erkennung ausschließen, sind Probleme durchaus möglich. :(== Interner Interrupt benutzt... Es wird zur Zeit nur ein Mode unterstützt. Der Special-Mode ist noch nicht implementiert. Ein Saven des Modules ist in der registrierten Version möglich.

MON MON heißt Maniacs-of-Noise Player. MON wurde z.B. in Unreal verwendet (teilweise auf festen Adr.), was normalerweise nicht der Fall ist. Bei MON ist die PlayRoutine im Module. Sie wird beim Start umgebaut, sa daß auch MON Analyzer, Volume, Balance und Voices-Regelung hat. Songende wird erkannt. MON hat teilweise Probleme auf Turbokarten. Songende hat Probleme !

MusikAssembler: Bei MusicAssembler ist die PlayRoutine ebenfalls im Module. Ich besitze ein Module von MA, welches rund 3K lang ist, erstaunlich oder (und das mit PlayRoutine) ! Music-Assembler ist ebenfalls Analyzerfähig... . Songende wird teilweise erkannt.

Music-Maker 4/8 Die Playroutine haben wir direct vom Author bekommen, Vielen

Dank an dieser Stelle. Die Eagleplayer-Anpassungen wurden vorgenommen. Soundsystem mit sehr gutem Klang, allerdings teilt sich ein Song in mehrere Dateien auf. Es muß das File mit der Endung *.Sdata geladen werden. Alle Files müssen sich in einem Verzeichnis befinden.
Die Playroutine ist MusicMaker V2.4 kompatibel.

Noisepacker2 Programm zum komprimieren von Noisetrackerfiles, effizienter als Prorunner2.0, aber nur mit Noisetrackerunterstützung. Stammt von Twins of Phenomena, wird gern von Phenomena, Quartz, Andromeda und auch Anarchy genutzt. Der Player kann die tatsächliche Länge des Modules berechnen und die Anzahl der Patterns usw. bestimmen (nützlich für Leute, die per Hand rippen) -> Eagleplayerfunktion "About Module" = [?]
Es wird der interne Interrupt genutzt, die Erkennungsroutine ist so geschrieben worden, daß das Modul ruhig auch zu lang oder zu kurz sein kann.
Ein Saven des Modules ist in der registrierten Version möglich.

Noisepacker3 Remake von NP2, mit eigenem Window, erheblich besserer Packrate, hängt sogar Propacker2.1 ab. Ansonsten siehe Noisepacker2.
Ein Saven des Modules ist in der registrierten Version möglich.

NoiseRunner Dieser Packer kommt von Chaos of Sanity. Der Player hat noch M.K.- Kennung drin, was jeden bisher erschienenen Player abkotzen läßt. Ich versteh sowieso nicht, warum immer noch Module geschützt werden. Wer Ahnung hat, kriegt die eh raus und plagt sich mit der Erstellung neuer Eagleplayer rum (Stimmts Turbo ?) , und wer keine hat, versagt schon bei Noisepackermodulen.
Doch zurück zur Playroutine: Genau wie beim Prorunner 1.0 habe ich meine Protrackerplayroutine angepaßt! Rest siehe dort.
Ein Saven des Modules ist in der registrierten Version möglich.
Ab V1.50 des Eagleplayers ist diese Playroutine intern im Eagleplayer.

NoisetrackerCompressed

Der United Forces Noisetracker in der Version 1.3 D, besitzt die Möglichkeit, Module in gepackter Form zu sichern. Für Leute, die noch mit Noisetracker arbeiten sollten, sicherlich eine sehr gute Alternative zum Noisepacker. Am Eagleplayer genauso nutzbar wie jeder andere Tracker auch, nutzt internen Interrupt, übrigens recht gute Packrate.

Bis jetzt werden die Modules mit Playroutine im Modul noch nicht unterstützt, wird noch nachgereicht.
Ein Saven des Modules ist in der registrierten Version möglich.

MTH Playroutine von Turbo bekommen und angepaßt. Unterstützt Sub-Songs und die meisten Funktionen die ein Eagleplayer haben muß.

OctaMED Player für 5-8-Stimmige Octamedsongs. Wurde von Delirium übernommen und entsprechend erweitert. Der Bug, der in der Playroutine vom Delitracker1.3 auftrat, wurde behoben. Ansonsten siehe MED.
Falls jemand eine Playroutine für 8 Stimmige HQ Wiedergabe besitzen sollte und diese frei kopierbar ist, sollte er sie mir umgehend zusenden, damit auf Turboamigas eine entsprechend bessere, mit dem FTM vergleichbare Qualität erreicht werden kann.

Oktalyzer Urahn aller 8 Stimmigen Musikprogramme und genau so sieht auch die Playroutine aus. Es war mir aber möglich, einen CIA-Timer einzusetzen, so daß unter NTSC und in den Productivity- und VGA-Modi die Abspielgeschwindigkeit trotzdem stimmt.
Der Eagleplayer unterstützt neuerdings Songend , Module-info und Patternumschaltung, außerdem wurden die Probleme mit Monoscope, Stereoscope und Quadrascope beseitigt.
Ab V1.52 unterstützt der Oktalyzerplayer auch den Pattern-scroller und Module-sowie Sampleinfo.

Old Soundtracker
Playroutine für alte Soundtracker (16 Samples). Es wurde die Playroutine des D.O.C. Soundtracker 2.0 verwendet, die von der Unterstützung halbwegs kompatibel zu alten Modulen ist. Es ist durchaus möglich, daß es Probleme mit einigen Modulen aufgrund der verwendeten Kommandos gibt, da jeder, der einen Tracker gecoded, geklaut oder gekloded hat, seine eigenen Kommandos benutzt hat. Da heutzutage sowieso keiner mehr damit arbeitet, wurde auf aufwendige Anpassungen verzichtet. Der Player ist intern im EaglePlayer und unterstützt SongLoading (auch gepackte Songs). Eine Einstellung des Default-InstrumentenPfads ist möglich. Die Samples dürfen gepackt werden (Powerpacker, File Imploder, CrM, LH, XPK, Stc, Pack)
Eine Save-Funktion ist in der registrierten Version möglich.

Player-P40A/P40B/P41A/P50A/P60A

Hierbei handelt es sich wieder um einen ProTracker-Packer, der uns Ripper ein bischen auf's Glatteis führen soll. Im ersten Augenblick liegt das Module nämlich auf festen Adressen, deshalb wurde in die Init-Routine eine EntReLogRoutine eingebaut (Ätsch !!! :-))

Die PlayRoutine war mal wieder selbstmodifizierend und hard-warehacked, aber sehr schnell. Der zweite Interrupt entfernt. Es war eine Sau-Arbeit, diese Playroutinen anzupassen, rippt man die eine P40A, dann spielt das andere P40A-module nicht mehr richtig und so weiter.

Mehrere Enforcer-Bugs wurde removed und die Player sollten jetzt sicherer laufen als früher. Unterstützt teilweise ModuleInfo und Songend.

Bei dem P50A handelt es sich um den besten Packer, den ich jeh gesehen habe. Er schlägt Promizer4.0, Propacker, Prorunner und wie sie alle heißen um Längen und hat dabei weniger Fehler als diese. Dafür stürzt der Packer selber sehr gerne ab. Ein Saven des Modules ist in der registrierten Version möglich. Creator: Jarno Paananen !

Ab Eagleplayer V1.50 sind die Player V4.x, sowie P50/P60 alle intern und die Probleme mit dem Abspielen sind endgültig behoben. Außerdem sollten die Player endlich eine bombensichere Checkroutine besitzen.

Promizer0.1: Der Protracker Optimizer von Frank Hülsmann (Tech). Erste Version des Promizer. PlayRoutine noch nicht im Module und keine SubSongs.

Promizer1.x: Vom Promizer gibt es inzwischen so viele Versionen (verschieden 1.x Versionen), daß man echt 'n Krise beim Anpassen des Players bekommt. Denn jede Version hat eine etwas andere Play-Routine und die will jedes mal angepaßt werden. Bei diese Version sind Subsongs möglich, eine SubsongRange wurde eingebaut.

Promizer2.0 Aktuelle Version des Promizers. Keine SubSong-Unterstützung mehr Wieso ????

Alle Promizer unterstützen PatternSwitching, Songend-Erkennung, Volume, Balance, Voices & Userprogramme.

Promizer4.0 Endlich mal wieder 'n Promizer ohne Playoutine im Module und mit Kennung, aber ziemlich viel von der Prorunner 2.0-Playroutine abgekupfert, Moduleformat vom Propacker 2.1 abgeguckt.

Hinweis:Promizer-Module findet man vor allem in TECH-Demos !!!
(und neuerdings bestimmt auch bei Masque)
99% rippbar !!!

Propacker V1.0a

Wurde von Azatoth of Phenomena geschrieben, packt äußerst mäßig, die Playroutine war selbstmodifizierend, hardwarehackend ohne Ende, einige Kommandos fehlen und obendrein will der Typ auch noch Geld dafür haben.

Für den Eagleplayer wurde der zweite Interrupt entfernt, eine Modulinfofunktion eingebaut, die die korrekte Länge und Packeffizienz angeben kann und zudem die Tempofunktion implementiert, das heißt, auch Module wie "Condom Corruption" (Spaceballs) und "Blow ya Nose Now" (E.O.K.D.P.) arbeiten mit der korrekten Geschwindigkeit.

-----> Aufgrund dessen, daß die Patterns hinter den Samples am Ende des Files liegen, toleriert die Erkennungsroutine keine zu kurzen Modules, weil ein Absturz sehr wahrscheinlich ist. Wenn das Modul also per Hand gerippt wird, lieber ein paar Bytes mehr save, der Player gibt die richtige Länge an. (und kann es in der registrierten Version natürlich auch save !!!)

Es ist eine Config-Möglichkeit gegeben, mit der man angeben kann, ob der Player über Requester ausgibt, wie viel länger das Modul gesaved werden muß (ungefähr), wenn es zu kurz ist, oder nicht.

Die Playroutine benutzt einen eigenen CIA-Timer

Achtung: Der Packer ist in der Lage, doppelte oder ungenutzte Patterns zu entfernen. Es ist also durchaus möglich, daß plötzlich im "About Module"-Modus eine andere Anzahl Patterns angezeigt wird als die des ungepackten Modules und somit auch eine andere Packrate und ursprüngliche Länge.

Achtung2:

Dieses Teil geistert auch noch unter den Namen "Strangeplayer" und "Hannitracker" durch die Gegend, beide mit katastrophaler Checkroutine, die auf die oben beschriebenen Gefahren keinerlei Rücksicht nimmt, nicht mal Lautstärke usw., peinlich ...

Ab V1.50 des Eagleplayers ist diese Playroutine intern im Eagleplayer.

Propacker 2.1 Alternative zum Noisepacker3, packt nur wenig schlechter, unterstützt aber alle Protrackerkommandos.

Die Checkroutine wurde von Turbo of Infect übernommen, allerdings wurden eine Menge Fehler beseitigt. :)--=

Ist im Gegensatz zu den mit "Vxx" betitelten Pro??Packern relativ sicher zu erkennen.

Ein Save des Modules ist in der registrierten Version möglich.

Ab V1.50 des Eagleplayers ist diese Playroutine intern im Eagleplayer.

Propacker 3.0 siehe oben, kleine aber feine Änderungen im Modul (andere Offsetangabe), nicht mit PP2.1 abspielbar, deshalb eigener Player. Ich habe dieses Teil bis jetzt nur auf Parasite-Musik-Disks gefunden. (dort auch die Playroutine gerippt)
Ein Saven des Modules ist in der registrierten Version möglich.
Ab V1.50 des Eagleplayers ist diese Playroutine intern im Eagleplayer.

PropackerVxx.2 Wie PropackerVxx.1, nur die Patterns sind in je 4 Teile gesplittet worden. Einzige Möglichkeit die beiden auseinanderzuhalten. Wird auch ab und zu benutzt. Die Erkennung gestaltet sich ziemlich schwierig. Im Prizip erkenne ich dieses Teil nur, indem ich den Typ 1 (PropackerVxx.1) ausschließe. Wenn also die Patterns in Ordnung sind und es sich nicht um Typ 1 handelt, ist es für den Player Typ 2. Interner Interrupt benutzt
Ein Saven des Modules ist in der registrierten Version möglich.

Alle Propacker unterstützen ab jetzt die Protracker Tempo Funktion !!

ProrunnerV1.0 Von Cosmos of Sanity erschaffene Playroutine, die Protracker-module konvertiert, um Geschwindigkeitsvorteile zu erbringen. Da ich den Prorunner bis vor kurzem noch nicht mein eigen nennen konnte, die aus "Wicked Sensation" (cooles Demo Jungs) gerippte Playroutine irgendwo nicht wollte, nachdem ich den 2. Interrupt (CIAB) gekillt hatte, beging ich einfach den Frevel und baute einen Online-Dekodierer in meine Protrackerroutine ein. Es läuft wunderbar, die Tempofunktion ist implementiert, Moduleinfo, Songend usw. usf. möglich, also wird das auch so bleiben, obwohl der Sinn des Prorunners dadurch total mißachtet wurde (in Bezug auf Speed). Durch die eindeutige Kennung der Module eine mir sehr symphatische Angelegenheit.
Ein Saven des Modules ist in der registrierten Version möglich.

Achtung: Bei "Creator" steht "Cosmos/Amiga Freelancers"

Die Prorunner Playroutine stammt auch von Cosmos, die Freelancers haben insofern damit nichts zu tun, da aber bei mir eine Protrackerroutine verwendet wurde, hielt ich es für angebracht, die F. zu erwähnen (die ja sowieso Ursprung alles Protrackermäßigen sind :))-=O=

Ab V1.50 des Eagleplayers ist diese Playroutine intern im Eagleplayer.

ProrunnerV2.0 Auch von Cosmos kommend, schlägt dieses Teil eigentlich mehrere Fliegen mit einer Klappe:

1. Gibt es eine eindeutige Kennung
2. Gibt es eine schnelle Playroutine
3. Gibt es dazu einen komfortablen Eagleplayer
4. Packt das Teil fast so gut wie Noisepacker2
5. besitzt der Eagleplayer die Tempo-Funktion

und 6. möchte ich jetzt von Cosmos die Werbekosten erstattet kriegen :)---=

Einen kleinen Wermutstropfen birgt allerdings der Prorunner2.0, er ist in Zeiten von Req und Reqtools zu unkomfortabel und er hat einen bösen Bug: beim Packen des Moduls "Condom Corruption" von Spaceballs stürzt er ab. (In einer früheren Ausgabe der Player.dok hatte ich behauptet, er packe "ministry" von TSL nicht. Sorry das war ein Bug von mir.) Ein Saven des Modules ist in der registrierten Version möglich.

Ab V1.50 des Eagleplayers ist diese Playroutine intern im Eagleplayer.

Protracker Was gibts hier noch zu sagen, DER Standard in Sachen Musik, gern benutzt, überall gesehen und oft kopiert. Hier wird die Protracker1.1-Routine angewandt. Da nach meinen Informationen der von Cryptoburners vorgeschlagene Standard noch in der Diskussion befindlich ist, sollte diese Routine mit allen Protrackermodulen klarkommen. Der Protracker-Replayer ist intern im EaglePlayer und unterstützt SongLoading, auch gepackte Songs. Ein Pfad-Einstellung für Default-Instrumente ist ebenfalls möglich. Die Samples dürfen gepackt werden (Powerpacker, File Imploder, CrM, LH, XPK, Stc, Pack) Aus Zeitgründen wurde noch keine Unterscheidung zum Noisetracker eingebaut, ich bin auch nicht so pingelig und habe bis jetzt auch noch keine Probleme mit Noisetrackermodulen gehabt, zumal die von Riffel als "intelligent" bezeichnete Unterscheidung zwischen NT/ST/ProT auch nicht 100%-ig ist und gelegentlich rumspinnt, scheint eher ein Gag zu sein, um die Playerzahl künstlich zu erhöhen ... Ein Saven des Modules ist in der registrierten Version möglich. Es wurden mehrere Bugs und sogar Enforcer-Hits in der original Playroutine entfernt. Die Songerkennung läuft jetzt 100% Alle internen ProTracker-Packer werden mit dieser Playroutine abgespielt.

Psygnosis Special

Entfernt, da es sich bei dem entsprechenden Modul um ein Game Music Creator Modul handelt. Die Leute von Infect scheinen überhaupt nichts mehr zu peilen. Die machen sogar aus GMC irgendwelche Custommodule.

Pumatracker1.0 Remake des legendären Futurecomposers, nicht 2.0 kompatibel, lädt aber dafür auch keine Futurecomposermodule, klingt dennoch überraschend gut (für diejenigen, die damit ein Module hinkriegen sollten). Wurde in ziemlicher Eile aus dem Pumatracker Deliplayer von Turbo of Infect erstellt. Songende

wird noch nicht unterstützt

Rob Hubbard Soundsystem mit Untersongs. Vorsicht: Es ist keine SubSongRange eingebaut, beim Weiterschalten besteht Absturzgefahr !!!
Analyzerfähigkeit wird unterstützt.

Ron Klaren Jawohl, der Eagleplayer unterstützt als erstes Programm seiner Klasse (oder als erstes überhaupt ?) die Ron Klaren Module inklusive Lautstärke, Voices, Balance, Analyzer, SongEnd. Außerdem entfernt der R.K. Player einige Hardwarehacks.

Allerdings ist zum Abspielen dieser Module noch einiges nötig und Programmiererfahrungen sind unerlässlich, da die Module auf festen Adressen liegen und ein special Format verwendet wird. Unser Vorschlag ist es, das Module vollständig zu disassemblieren, so daß ALLE festen Adressen in Labels umgewandelt werden (auch die festen Adressen im Datenteil !!). Danach in den Assembler laden, den Eagleplayer.i includen und das Macro "KlarenHeader" an den Anfang des Moduls setzen. Das Modul muß unbedingt in den Chipmem geladen werden und darf nur eine Sektion besitzen ! Als letztes wird das Modul als Objektfile abgespeichert. (Seka & ASM-One : WO für Write Object)
Ron Klaren-Modules dürfen nicht gepackt werden !!!

Ein Beispiel liegt als disassemblierter Sourcecode bei.
Have Fun !

Beispiel:

```

section 0,Code_C
Klarenheader end-start ;die Zahl ist unbedingt nötig
                ;und gibt die Modullänge als
                ;Objektfile an. !! WICHTIG !!
Start
<hier das disassemblierte Modul einfügen>
End

```

!!!! Achtung !!! Es dürfen keinerlei Änderungen im Module oder der Play-routine vorgenommen werden, da der Eagleplayer ansonsten in arge Bedrängnis kommt ! Es müssen die original Hardwarehacks enthalten sein !!!

Anmerkung:

Die Ron Klaren Module beginnen mit mehreren festen Jumps, gefolgt von mehreren move #xx,d0 bra xx - Befehlen, die ebenfalls nicht gelöscht werden dürfen. Sie dienen zur SubSongBerechnung.

SIDMon1: Soundsystem von 1988, von R.v. Vliet, läuft nur unter Kick1.3, der reinste Hardwarehack, PlayRoutine im Module. Sidmon unterstützt Analyzer..., Songende und NextPattern (unter Schweißtrifen eingebaut)

Wenn der Umbau des Modules nicht erfolgen kann, dann wird das Module normal, wie in jedem anderem Player abgespielt. Die unmöglichen Funktionen (Volume...) werden dann gesperrt.

SIDMon2.0 Sidmon II, ist MIDI-fähig und läuft nur unter Kick 1.3, ist mal wieder der reinste Hardwarehack und ist sehr gewöhnungsbedürftig. Mit Sidmon II können allerdings sehr gute synthetische Songs erstellt werden. Modules sind gegenüber dem Sidmon1.0 ohne Playroutine, was für unsereiner immer positiv ist. Auch die Kennung ist nicht zu übersehen. Der Player unterstützt Pattern-switching, Songende sowie Lautstärke, Balance, Analyzer... , was will man mehr?

Mehrere durch Enforcer entdeckte Fehler wurden beseitigt.

Sonic-Arranger: Dieser Player spielt die Sonic-Arranger Songs(ungepackt) und Sonic-Arranger Module (mit Replay) ab. Zum Abspielen der Modules mit PlayRotoutine wird der Song-Replayer verwendet, daher kann es nicht ausgeschlossen werden, daß es Kompatibilitätsprobleme gibt.

Der Player unterstützt Patternswitching, SubSongs, Volume, Balance, Analyzer und Voices.

Es wurde der Delirium-Player verwendet und verbessert. Die PlayRoutine für SA-Songs liegt jetzt im FastRam.

Es ist höchstwahrscheinlich der Nachfolger von InStereo/Synth.

SonicArranger ist ein kommerzielles Musikprogramm von Brain-Trace-Design. An dieser Stelle vielen Dank für die den Sonic-Arrangers von Carsten Schlote.

SongEnde wurde nach schweißtriefender Arbeit eingebaut.

Weiterhin sind Patternumschaltung und SubSongs möglich.

SoundControl Playroutine von Turbo übernommen. Es wurden alle typischen Anpassungen für Eagleplayer vorgenommen.

SoundFX1.3 Wenn es stimmt, daß dieses Programm kommerziell vertrieben wird/ wurde, dann läufte mir kalt den Rücken runter. Die "Playvoice"-routine wurde so was von offensichtlich vom Soundtracker kopiert, nee Leute, so ja nu nich. Ansonsten wurde alles nötige angepaßt - inclusive Patternumschaltung und Moduleinfo, wie es sich für einen Soundtracker gehört.

Ein Saven des Modules ist in der registrierten Version möglich.

SoundFX2.0 Weiterentwicklung des obigen Systems, unterstützt 31 Samples (AHA !) , ansonsten konnte kaum Neuerungen erkannt werden.
Rest siehe oben.
Ein Saven des Modules ist in der registrierten Version möglich.

Soundtracker2.6 Weiterentwicklung der legendären D.O.C. Soundtracker, 31 Samples, gesplittete Patterns (jede Spur mit eigenem Pattern), aber genausowenig SpecialFX, wie alle alten Soundtracker.
Ein Saven des Modules ist in der registrierten Version möglich.

SoundMaster: PlayRoutine im Module. Sollte alles normale für solche Player unterstützen. Ich bräuchte mal 'n gutes TestModule.

SoundMon: Ein echter C64-Abklatsch und so sieht er auch aus. Läuft aber unter Kick2.0 und arbeitet bedingt (wenn man ST-00:Modules assigned) auch mit Festplatten zusammen. Ich finde es ist echt erstaunlich, wie man mit solch einen Teil überhaupt was hinkriegt, aber die Module sind erstaunlich gut.
SoundMon wird oft in Intros wegen seines guten Klangs und wegen der Kürze der Module verwendet.
PlayRoutine liegt jetzt im FastMem. SongEnde wird erkannt.
Eine Längenberechnung wurde ebenfalls eingebaut.
Ein Saven des Modules ist in der registrierten Version möglich.

Syntracker: Ein schön übersichtliche Playroutine. Der Creator hat aber anscheinend noch nie was von PC-relativen code gehört. Ich konnte die eigentliche Playroutine von 4100 Bytes auf 3400 Bytes kürzen. Habe lediglich ein Testmodule !!! Die Playroutine ist absolut soundtrackertypsich aufgebaut, unterstützt aber trotzdem synthetische Instrumente. Patternumschaltung, Songenderkennung, ModulesInfo und Save sind angepaßt.
Das Hauptprogramm sieht aus wie ein 4 stimmiger Oktalyzer und hat so einige Fehler.
Die Playroutine gurkt im Speicher rum.

Startrekker Es wird die 4-Stimmige AM-Abspielroutine des AudioSculpture verwendet. Startrekker wurde von Exolon of Fairlight geschrieben.
Als besonderes Extra ist das .NT file nicht unbedingt

erforderlich. Probleme gibt es nur dann, wenn es sich um ein Module handelt, das mit AM-Samples arbeitet und das NT-File nicht geladen werden kann (man hört dann halt nichts bei diesen Samples). Ansonsten unterstützt der Startrekkerplayer alle Möglichkeiten der anderen Tracker, inclusive Songend und Moduleinfo. Der Startrekker ist intern im EaglePlayer und unterstützt SongLoading (auch gepackte Songs). Eine Einstellung des Default-Instrumenten-Dirs ist möglich. Die Samples dürfen gepackt werden (Powerpacker,File Imploder,CrM,LH,XPK,Stc,Pack) Ein Saven des Modules ist in der registrierten Version möglich.

TFMX Von Chris Hülsbeck erschaffenes Soundsystem.

Hier in 3 Varianten implementiert:

1. TFMX V1.5
 2. TFMX Pro
 3. TFMX 7 Stimmig (Idee von J.Hippel,Code J.Hippel ? & C.H.)
- TFMX stellt das Nonplusultra unter den professionellen Musikeditoren dar und klingt unter der Hand eines Profis zum schwärmen. Besonders betrachtenswert ist der Verlauf der Samples im Stereoscope, wie Halleffekte simuliert werden, einfach geil. Alle drei wurden an den Eagleplayer angepaßt in Bezug auf Lautstärkeregelung usw. SongEnde & Restart wird erkannt (boah eh).

Einen TFMX mit Playroutine im Modul, wie beim Powerplayer (ich kann einfach nicht glauben, wie man so ein Teil POWER.. nennen kann) behauptet, gibt es nicht !! (von Frank Riffel of DELIRIUM bestätigt) S.Fuhrmann meinte wohl Hippel damit, der arme....

Ab Version 1.52 gibts es 2 neue TFMX Player, einer für Pro und einer für 7V (Danke an TIK of Retire für den 7V !)
Der TFMX pro erkennt nun auch den 7V nicht mehr.

TME The Musical Enlightenment, ein Soundprogramm, für das ich leider nur ein Module habe, so daß ein ausgiebiger Test unmöglich war. Die Playroutine gab's bei Delirium, allerdings wieder fast "nackig", nur Subsongs wurden unterstützt und dabei greift der TME an nur einer, wiederhole einer Stelle auf die Audio-Hardware zu

Jetzt ist der TME einer der wenigen Player, bei dem es nicht ein einziges gesperrtes Gadget im Hauptfenster des Eagleplayers gibt. Die korrekte Länge wird außerdem berechnet und auch richtig gesaved (in der registrierten Version). Ein Saven des Modules ist in der registrierten Version möglich.

Tronic Der Tronic ist es mit hoher Wahrscheinlichkeit KEIN Tracker. Die Playroutine wurde von Marley/IFT gerippt und als Deliplayer rausgebracht. Ich habe dann noch Laustärke/Balance/Voices, sowie

Analyzerunterstützung und Songend eingebaut.
Ein Bug in der Playroutine wurde entfernt.

Unic-Tracker Ich bin mir noch nicht schlüssig, ob es sich hierbei wieder mal um einen Soundkodierer handelt oder Laxity of Kefrens einen eigenen Tracker entwickelte. Jedenfalls war die Playroutine genau wie der Propacker 1.0a ein wüstes Etwas mit allen Schikanen, die man eigentlich vermeidet. Nach etlichen Flüchen (Buggs) präsentieren wir nun einen Unic-Tracker, der ohne selbstmodifizierenden Code auskommt, nur mit dem CIA-Interrupt arbeitet und somit auch unter NTSC läuft. Die Unic-Tracker-Playroutine ist sowieso ziemlich kurios, es sieht so aus, als ob Laxity eine Noisetrackerplayroutine in eine Protrackeroutine umbauen wollte, komisch, komisch.
Ich habe sicherheitshalber erstmal die Protrackerfunktionen "Sampleoffset" und "Tempo" hinzugefügt, so daß die meisten Protrackermodule hinhalten. (eigener Konverter)
Besonderheit des Unic-Trackers : Er kürzt die Patterns um $\frac{1}{4}$!
Ein Saven des Modules ist nur in der registrierten Version möglich.
Ab V1.50 des Eagleplayers ist diese Playroutine intern im Eagleplayer.

Unic-Tracker 2 Im "Desert Dream" verwendetes "Unic"-Format ohne Kennung und Modulname, dafür aber tierisch wilde Playroutine. Ich habe aus Faulheit die alte Unic-Routine umgebaut, fertig !!!
P.S. Die Module liegen im Demo auf \$1532 , Endadresse bei \$152e
Ab V1.50 des Eagleplayers ist diese Playroutine intern im Eagleplayer.

VSS Voodoo Supreme Synthesizer, wurde wahrscheinlich direkt vom Autor angepaßt, bekam von mir noch eine Voices-Routine, einen CIAB-Timer, Analyzerunterstützung und Speedregelung im Eagleplayer verpaßt. Bei dem Demosong Voo4 gefällt mir besonders der Look der Samples, echt cool gemacht.
Problem : auf dem A4000 kein Ton !

Whittaker Womöglich der erste Whittaker bei dem man Balance, Volume und Voices steuern kann. Wer Whittaker kennt, weiß daß bei fast jedem WH-Module eine etwas andere PlayRoutine benutzt wird. Diese Play-Routine wird bei InitPlayer umgebaut Nicht funktionierende Module dieses Playertyps bitte umgehend an uns senden. Ein Bug in der ModuleInternen PlayRoutine, der bei zu kurzen Modulen auftritt, wird abgefangen !!!
Mittlererweile funktioniert der Whittaker mit allen uns zur Verfügung stehenden Modulen, aber wir übernehmen keinerlei

Garantie. Songende wird teilweise erkannt.
Whittaker wurde vor allem in Spielen um 1987-88 verwendet.
Man erkennt das Soundsystem am typischen Klang.

Wizard of Sound

Ich habe - außer beim eigentlichen Programm - noch nie ein Wizard of Sound Modul gefunden, wahrscheinlich weil das Teil niemandem gefallen hat ...
Laßt mal, mir auch nicht, ich habe eine Ewigkeit zugebracht, dieses Programm anzupassen. Mal davon abgesehen, daß es sich um den absoluten Spaghetticode handelt, besaß alleine die kurze Playroutine 2 kB, wiederhole 2 kByte Reloc. Wahnsinn, und das, wo nur die Hälfte der Register benutzt ist.
Auch die Erkennung gestaltet sich schwierig. Ich erkenne das Modul im wesentlichen daran, daß es immer 9000 Bytes lang ist (die Kommandos werden zwar auch gecheckt, aber die Trefferquote ist gering, da so gut wie alle von den 255 Möglichkeiten eines Bytes von WOS genutzt werden).
Songend, Analyzer, Lautstärke usw. sind selbstverständlich.
Die Samples dürfen gepackt werden (Powerpacker, File Imploder, CrunchOMania, LH, XPK, Stc, Pack)
Der Player unterstützt eine Config-Routine, ist in der Lage, sein Instrumentendirectory als Konfiguration zu sichern.
("ENVARC:Eagleplayer/EP-Wos.Cfg")
Das Directory, wo sich die Instrumente befinden, bitte über das [CONFIG]-Gadget abgeben (anlicken, Pfad eingeben)

Danksagungen:

- an Turbo of Infect für einige "exotische" Module, sowie Playroutinen und Konverterprogramme, außerdem für den Exotic-Ripper, der eine ganze Menge an Modulen findet (finden will ?)
- an die Autoren von XPK, die ein außerordentliches Paket rund um Datenkomprimierung als Freeware erschaffen haben, weiter so !
- an alle Soundsystemschafter und Soundkodierer/converter-Schreiber, die eine eindeutige Erkennungsmöglichkeit für die Module ermöglichen
- Thomas Winischhofer für seine Tips und die angepaßten MusicMaker
 - Chaos of Sanity für seinen - eigentlich in einem anderen Zusammenhang gefallenen - Tip, der half, den TFMX 7V auf dem A4000 lauffähig zu machen. "NOP" RULEZ !!
 - TIK of Retire für den TFMX 7V Replayer, der auch mit den T-III Modulen klarkommt

1.88 fehler in den playroutinen

4.2.6 Fehler in den Playroutinen

Bei Tests mit fast allen Replayern kamen verschiedene Fehler ans Tageslicht. Hier ist nun eine Zusammenstellung der möglicherweise fehlerhaften bzw. unsicher laufenden Player

- Action Amics
- ActivisionPro (Cache aus, sonst werden z.T. Noten verschluckt)
- GMOD
- Hippel
- Hippel7V
- Old Whittaker
- Whittaker
 - VSS (Probleme auf A4000)

Generell kann man sagen, alle Soundsysteme mit Playroutine im Module sind in gewisser Weise gefährdet, genauso wie Soundsysteme, die uralt sind. Selbst die Soundtrackerclones sind teilweise unsicher, da der "Standard" oft nicht oder nur teilweise beachtet wird.