

mAirList

Kompakte Sendeautomation für Windows

Handbuch

Torben Weibert

22. Januar 2006



Wichtiger Hinweis

Dies ist eine vorläufige Anleitung für mAirList. Sie bezieht sich auf die Version 1.5. Die Anleitung wird nach Bedarf und vor allem nach Zeit aktualisiert, ergänzt und erweitert. Da sich mAirList aber fortlaufend weiterentwickelt, und nicht immer genug Zeit da ist, auch die Anleitung zu aktualisieren, ist dieses Dokument nicht immer auf dem neuesten Stand. Darauf ist beim Lesen bitte Rücksicht zu nehmen.

Bei offenen Fragen steht das mAirList-Supportforum unter <http://forum.mAirList.de> jederzeit zur Verfügung.

Inhaltsverzeichnis

1. Einführung	5
1.1. Über mAirList	5
1.2. Systemvoraussetzungen	6
1.3. Lizenz	6
1.4. Haftungsausschluss	6
1.5. Über den Autor	6
1.6. In eigener Sache	6
2. Installation	7
3. Konzept	8
3.1. Playlists und Player	8
3.2. Die Cartwall	8
3.3. Die Browser	9
3.4. Faderstart	9
3.5. Datenbankbindung	9
4. Konfiguration	9
4.1. Playlists	10
4.2. Player	11
4.3. Cartwall	12
4.4. Browser	12
4.5. Faderstart	13
4.5.1. Joystick/Gameport	13
4.5.2. Hotkeys	13
4.5.3. MIDI	13
4.6. HotDirs	13
4.7. Logging	14
4.8. Datenbanken	14
4.9. Fernsteuerung	15
4.10. BASS.DLL	16
4.11. Sonstiges	16
5. mAirList benutzen	17
5.1. Das Hauptfenster	17
5.1.1. Die Playlist	17
5.1.2. Die Browser	17
5.1.3. Die Cartwall	18
5.2. Arbeiten mit der Playlist	18
5.2.1. Einfügen von Elementen	18
5.2.2. Verschieben von Elementen	19
5.2.3. Löschen von Elementen	19
5.2.4. Playlist-Optionen	19
5.3. Arbeiten mit Browsern	20
5.3.1. Der Archiv-Browser	20
5.3.2. Der Verzeichnis-Browser	20
5.3.3. Der Verzeichnisbaum-Browser	20
5.3.4. Der Playlist-Browser	20

5.3.5. Der Datenbank-Such-Browser	21
5.3.6. Der Datenbank-Playlist-Browser	21
5.3.7. Der Uhr-Browser	21
5.4. Arbeiten mit den Playern	21
5.4.1. Player laden	21
5.4.2. Spielzeit- und Backtiming-Anzeige	21
5.4.3. Starten und Stoppen von Playern	21
5.4.4. Der erweiterte PFL-Modus	21
5.4.5. Visualisierung der Ramp	23
5.5. Arbeiten mit der Cartwall	23
5.6. Öffnen und Speichern von Dateien	23
5.6.1. Vorlagen	23
5.6.2. Standard-Vorlage	23
5.6.3. Import von Fremdformaten	24
5.6.4. Anhängen von Dateien	24
5.7. Tastaturbedienung	24
5.8. Die Automation	24
A. Faderstart-Befehle	24
B. Logfile-Variablen	25
C. Fenster-Layout anpassen	26
C.1. Aufbau von <code>layout.ini</code>	26
C.2. Beschreibung der Parameter	27
C.2.1. Gemeinsame Parameter aller Elemente	27
C.2.2. Zusätzliche Parameter von <code>[Playlistx]</code>	27
C.2.3. Zusätzliche Parameter von <code>[Playerx_y]</code>	28
C.2.4. Zusätzliche Parameter von <code>[CartWall]</code>	29
C.2.5. Zusätzliche Parameter von <code>[Clock]</code>	29
D. Kommandozeilen-Parameter	29

1. Einführung

1.1. Über mAirList

mAirList ist eine kleine, schnelle, unversell einsetzbare Sendeautomation für Windows 95/98/ME/2000/XP.

Zu den wesentlichen Features von mAirList gehören:

- Beliebig viele Playlists.
- Beliebig viele Player pro Playlist.
- Simultanes Ausspielen auf beliebig vielen Playern gleichzeitig.
- Zusätzlich integrierte Cartwall/Jingleplayer mit beliebig vielen Slots.
- Soundausgabe realisiert mit BASS.DLL (<http://www.un4seen.com>).
Unterstützte Dateiformate: MP3, MP2, WAV, und OGG.
Weitere Formate (WMA, AAC, ...) können durch BASS.DLL-Plugins nachinstalliert werden.
- "Intelligente Playlist" lädt die Player automatisch nach. Wahlweise können aber auch einzelne Stacks für jeden Player verwendet werden ("Split Mode").
- Dateien können sehr einfach durch Drag&Drop aus dem Windows-Explorer oder den eingebauten Browsern eingefügt werden.
- Zu jeder Datei können Cue-, Ramp/Intro-, Outro- und Fade-Out-Punkte definiert und wahlweise in der Playlist oder im ID3-Tag von MP3s gespeichert werden. Die Ramp/Intro-Zeit wird beim Abspielen vom Player graphisch dargestellt.
- Automations-Modus "fährt" die Sendung automatisch unter Beachtung von Cue- und Fade-Out-Punkten.
- Universelle Faderstart-Möglichkeiten per Hotkeys, MIDI oder Gameport. Über zweckentfremdete Joystick-Interfaces können so fast alle Mischpulte mit Faderstart-Ausgang zusammen mit mAirList betrieben werden.
- Anbindung an die "Schwester"-Musikdatenbank eldoDB zum schnellen Zugriff auf das Musik-Archiv und vordefinierte Playlisten.
- Arbeitsumgebung und Playlist kann in XML-basiertem Dateiformat gespeichert und geladen werden.
- Arbeitsoberfläche kann optisch frei gestaltet werden (Position, Farbe und Erscheinungsbild der graphischen Elemente).

mAirList wurde als Sendeautomation für kleine, nicht-kommerzielle Radiosender geschrieben, die sich eine "kommerzielle" Automation schon aus Kostengründen nicht leisten können, die aber auch den Verwaltungsaufwand scheuen, den die kommerziellen Produkte mit ihren komplexen Datenbanken nach sich ziehen.

1.2. Systemvoraussetzungen

mAirList stellt die folgenden (wenigen) Anforderungen an den Computer, auf dem es betrieben werden soll:

- Microsoft Windows 95, 98, ME, 2000 oder XP.
- Von Windows unterstützte Soundkarte. Für den reibungslosen Betrieb mit der BASS.DLL sollte der Soundkarten-Treiber im WDM-Format vorliegen (was aber bei den aktuellen Karten fast ausnahmslos der Fall ist).
- Aktuelle Version von DirectX, mindestens Version 8.1. Die jeweils neueste Version kann auf <http://www.microsoft.com> heruntergeladen werden.
- Für Faderstart-Anbindung an das Mischpult (optional): MIDI- oder Gameport bzw. umgebauter Joystick mit Treiber.
- Für die Datenbankanbindung (optional): Ein von mAirList unterstützter SQL-Server (PostgreSQL, MySQL bis Version 4.0, o.ä.).

Wie performant mAirList ist, hängt von der Performance des Rechners ab. Auf aktuellen Systemen ist das Ausspielen auf allen drei Kanälen gleichzeitig überhaupt kein Problem. Ansonsten gilt: ausprobieren (und evtl. die CPU-Auslastung im Auge behalten).

1.3. Lizenz

mAirList darf von nicht-kommerziellen Radiosendern und -projekten kostenlos eingesetzt werden. Gleiches gilt für Privatpersonen, sofern durch den Einsatz der Software keine Einnahmen erzielt werden.

Anfragen zum kommerziellen Einsatz bitte an den Autor. E-Mail: torben@weibert.de

1.4. Haftungsausschluss

mAirList wurde nach bestem Wissen und Gewissen entwickelt und getestet. Der Autor haftet nicht für Fehler in seinem Programm und eventuelle dadurch verursachte Schäden.

1.5. Über den Autor

Torben Weibert, Jahrgang 1977, ist Diplom-Informatiker und wissenschaftlicher Mitarbeiter der Arbeitsgruppe "Informationssysteme und Sicherheit am Lehrstuhl" am Lehrstuhl Informatik VI der Universität Dortmund. Seit November 2000 ist er beim Dortmunder Campusradio *eldorado** tätig – sowohl als Moderator und Redakteur als auch als "Herrscher" über die Technik.

1.6. In eigener Sache ...

Wenn mir vor drei Jahren jemand gesagt hätte, dass ich einmal ein Handbuch für mAirList schreiben würde, hätte ich nur ungläubig gelacht – denn dass das Programm einmal außerhalb von *eldorado** zum

Einsatz kommen würde, war weder geplant noch absehbar. Eigentlich sollte mAirList nicht einmal bei eldorado* zum allgemeinen Einsatz kommen, denn ich hatte mAirList vornehmlich für **mich** geschrieben, damit ich nicht mehr mit dem alten, umständlichen¹ "Airlist" meine Sendungen fahren musste.

Dass es da was gab, was MP3s spielen konnte und einfaches Drag & Drop in die Playlist beherrschte, sprach sich unter den Moderatoren bald herum, und plötzlich wollte jeder mAirList benutzen. Die Argumente sprachen wohl für mich ;) So wurde mAirList zum offiziellen eldorado-Jingle-Player (die Musik kommt bei uns noch immer von CD). Irgendwann kamen auch die ersten Anfragen von anderen Sendern. Radio Q aus Münster waren glaube ich die ersten. Ob sie mAirList jemals benutzt haben, weiß ich nicht, jedenfalls habe ich damals die alte Anleitung geschrieben.

Der Schritt, die Software auch für andere Sender zugänglich zu machen, lag dann eigentlich auf der Hand. Seitdem hat sich mAirList kontinuierlich weiterentwickelt, etliche Features sind hinzugekommen, Dinge, die bei eldorado* vermutlich nie zum Einsatz kommen werden, die aber für andere Sender wichtig waren, so dass es sich lohnte, sie einzubauen. Nicht zuletzt durch die vielen Ideen und Testberichte, die über die Support-Mailingliste kamen, wuchs mAirList zu einem brauchbaren "Produkt" heran. Allen, die sich daran beteiligt haben, sei hiermit herzlichst gedankt.

Wo und in welcher Anzahl mAirList heute benutzt wird, kann ich höchstens anhand der Web-Statistiken raten. Die Download-Zahlen liegen bei etwa 100 pro Monat, Tendenz steigend. Das soll aber nichts heißen. Natürlich bin ich neugierig. Wer also – wo er schon kein Geld für die Software bezahlen muss – mir jedenfalls mitteilen will, wo mAirList bei ihm im Einsatz ist, darf mir gerne eine kleine Mail schreiben: torben@weibert.de.

2. Installation

mAirList wird als .zip-Datei ausgeliefert. Diese enthält die folgenden Dateien:

- mAirList.exe – die Programmdatei
- mAirListConfig.bat - Batchdatei, die das Hauptprogramm mit dem Parameter `-config` startet, um in den Konfigurationsdialog zu gelangen.
- mAirList.ini – eine (Beispiel-)Konfigurationsdatei
- layout.ini – eine Beispiel-Layout-Konfigurationsdatei
- WHATSNEW.txt – eine Textdatei mit den Neuerungen seit der letzten Version
- mAirList.pdf – diese Anleitung
- BASS.DLL – Bibliothek für die Soundausgabe.

mAirList kann ganz einfach durch Entpacken des Archivs in ein beliebiges Verzeichnis installiert werden. Eine Setup-Routine gibt es nicht. Ebenso wenig benutzt mAirList die Windows-Registry. Alle Konfigurationsdaten sind in der Datei `mAirList.ini` abgelegt. Dadurch kann mAirList auch im Zweifel einfach und ohne Rückstände wieder entfernt werden, indem das Programmverzeichnis wieder gelöscht wird.

¹Bei Airlist musste man für jedes Element erst lange Titel und Autor eintippen, bevor man es in die Playlist einfügen konnte; ein Vorgang, der so langwierig war, dass der Moderator der Morgensendung das meist schon am Vorabend erledigte oder vom Redakteur erledigen ließ – noch heute spricht man bei eldorado* noch manchmal vom "Airlist bauen". Trotzdem bin ich dem Autor des alten Airlist natürlich dankbar, denn mAirList hat sich viel von der alten Software abgeguckt.

Soll mAirList mit der eldoDB-Datenbank zusammen betrieben werden, muss zusätzlich die Datei `libmysql323.dll` aus dem eldoDB-Archiv in das mAirList-Verzeichnis bzw. ein anderes im Systempfad liegendes Verzeichnis kopiert werden.

Vor dem ersten Start muss mAirList mit dem Konfigurationsprogramm konfiguriert werden (siehe Abschnitt 4).

3. Konzept

In diesem Abschnitt wird das grundlegende Konzept von mAirList anhand seiner einzelnen Komponenten erläutert. In Abschnitt 4 wird dann beschrieben, wie die gewünschten Komponenten im Konfigurationsprogramm ausgewählt und eingerichtet werden.

3.1. Playlists und Player

Die wichtigste Komponente von mAirList sind eine oder mehrere Playlists, die auch den größten Platz auf der Benutzeroberfläche einnehmen. mAirList unterstützt beliebig viele Playlists gleichzeitig, wenngleich für die meisten Anwendungsszenarien eine oder zwei Playlists ausreichen sollten.

Die Playlist enthält die Elemente (z. B. Lieder, Jingles oder Beiträge), die in der laufenden Sendung ausgespielt werden sollen. Dazu verfügt jede Playlist über einen oder mehrere Player, die am unteren Bildschirmrand dargestellt werden. Wird mAirList an einem Mischpult eingesetzt, sollte jeder Player einem physikalischen Kanal auf dem Mischpult und damit einem Fader zugeordnet.

Die Elemente aus der Playlist werden nacheinander auf den ihr zugeordneten Playern ausgespielt, entweder automatisch (Automations-Betrieb) oder manuell durch den Moderator (Assist-Betrieb). Zusätzlich können die Elemente über die PFL-Funktion vorgehört werden. Die Ausgabe kann auf einer eigenen Soundkarte oder derselben Soundkarte erfolgen, auf der auch das eigentliche Ausspielen stattfindet. Über die Extra-PFL-Funktion können Elemente zusätzlich über einen vordefinierten Kanal vorgehört werden, auch dann, wenn der Player bereits läuft.

Im Assist-Betrieb kann die Playlist so konfiguriert werden, dass freie Player automatisch mit dem nächsten Element geladen werden. Die Arbeitsweise von mAirList entspricht dann der eines Cartplayers mit zugeordnetem Stack, über den mehrere Elemente nacheinander auf Knopfdruck abgefahren werden können.

3.2. Die Cartwall

Zusätzlich zu den Playern stellt mAirList eine Cartwall mit beliebig vielen Cartplayern (Slots) zur Verfügung. In jeden Player kann ein einzelnes Element geladen werden, zum Beispiel ein Jingle oder Drop. Die Belegung der Cartwall kann als Datei abgespeichert und später wieder geladen werden. Über eine Drop-Down-Menü kann auf häufig benötigte Cartwall-Belegungen schnell zugegriffen werden.

3.3. Die Browser

Um Elemente in die Playlists und die Cartwall einzufügen, stellt mAirList sogenannte Browser zur Verfügung, die sich am rechten Bildschirmrand befinden. Neue Elemente können dann mit der Maus in die gewünschte Playlist bzw. den Cartplayer gezogen werden.

Immer vorhanden ist der Archiv-Browser, in den alle bereits abgespielten Elemente aus den Playlists verschoben werden. Weiterhin lassen sich einblenden: Dateibrowser für Festplatten-Verzeichnisse, Dateibäume sowie vorhandene M3U-Playlisten. Außerdem stehen Browser für den Zugriff auf die eldoDB-Datenbank (siehe Abschnitt 3.5) zur Verfügung.

3.4. Faderstart

Alle Funktionen von mAirList (zum Beispiel die Steuerung der Player) werden über Faderstart-Befehle gesteuert, die mAirList auf verschiedenen Wegen generiert. Zum einen können für alle Funktionen Hotkeys definiert werden und diesen die gewünschten Befehle zugeordnet werden. Andererseits unterstützt mAirList die Kopplung an die Faderstart-Funktion von Mischpulten oder andere Fernsteuerungen über MIDI oder selbstgebaute Gameport-Interfaces.

3.5. Datenbankanbindung

In der Grundkonfiguration werden die Playlists und Cartplayer direkt über die eingebauten Dateibrowser bzw. den Windows-Explorer gefüllt. Optional kann mAirList auch unterschiedliche Datenbanken ansprechen, um auf Musikarchive, Playlisten usw. zuzugreifen. Derzeit werden folgende Datenbanken unterstützt:

- Die (alte) mAirList-Datenbank "eldoDB" (MySQL-basiert)
- "radioDB" von Christoph Krämer (PostgreSQL-basiert)
- Eine lokale iTunes-Installation, wobei die von iTunes erstellte Datei `iTunes Music Library.xml` ausgelesen wird.
- In Planung: Die (neue) mAirList-Datenbank mAirListDB.

4. Konfiguration

Alle Einstellungen von mAirList werden in der Datei `mAirList.ini` im Programmverzeichnis abgelegt. Zum Bearbeiten der Datei dient der Konfigurationsdialog, der erscheint, wenn man `mAirList.exe` mit dem Parameter `-config` (oder über die mitgelieferte Batch datei `mAirListConfig.bat`) startet. Beim Start wird auf der linken Seite eine Reihe von Rubriken angezeigt, rechts können dann die Einstellungen verändert werden. Im folgenden werden die einzelnen Einstellungsmöglichkeiten erläutert.

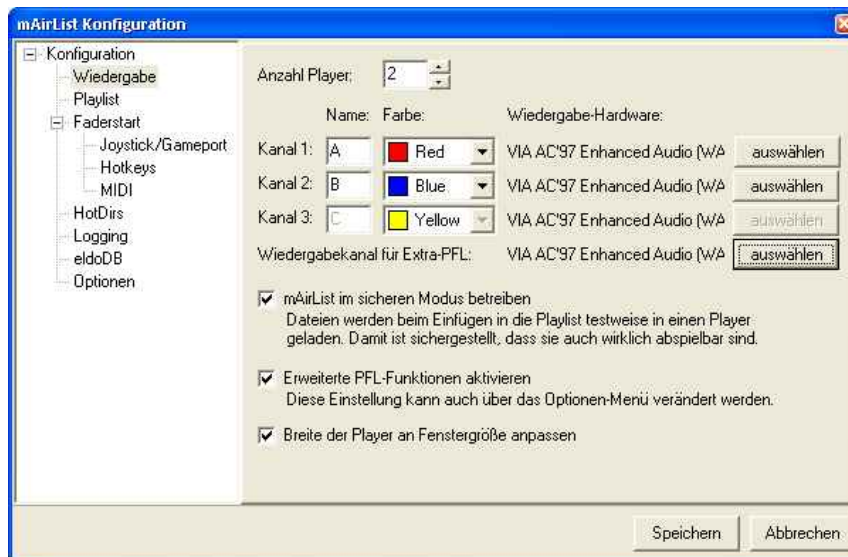


Abbildung 1: Der Konfigurationsdialog

4.1. Playlists

Das Herzstück von mAirList sind eine oder mehrere Playlists, in denen die zu spielenden Elemente geladen werden. Die Anzahl dieser Playlists ist prinzipiell nicht eingeschränkt, wenngleich die meisten Stationen je nach Anwendung mit einer bis drei Playlists auskommen sollten.

Anzahl der Playlists Hier kann ausgewählt werden, wie viele getrennte Playlists auf dem Bildschirm erscheinen sollen. Standardwert ist eine Playlist. Zu jeder Playlist erscheint im Menübaum links ein neuer Menüpunkt. Durch Klicken auf den Menüpunkt gelangt man auf die entsprechende Konfigurationsseite.

Anzahl Player Jede Playlist verfügt über eine (unbegrenzte) Anzahl von Playern, über die die Elemente der Playlist – manuell oder automatisch – ausgespielt werden. Für jeden hier eingestellten Player erscheint wiederum ein neuer Menüpunkt auf der linken Seite des Fensters. Die dahinterliegende Konfigurationsseite und ihre Einstellungen wird in Abschnitt 4.2 beschrieben.

Schriftart Hier kann die Schriftart ausgewählt werden, die für die Playlist verwendet werden soll. Standardwert ist Arial, 16 Pixel.

Toolbar-Position Jede Playlist verfügt über eine Toolbar, über die beispielsweise die Automation gesteuert wird. Hier kann ausgewählt werden, ob die Toolbar über oder unter der Playlist erscheinen soll.

Optionen Folgende Optionen können zu jeder Playlist eingestellt werden:



erweiterte Anzeige	Jeder Playlist-Eintrag wird zweizeilig dargestellt.
Backtiming-Anzeige	In diesem Modus zeigt mAirList automatisch die voraussichtlichen Startzeiten der Elemente in der Playlist an.
Fortschrittsbalken anzeigen	Wenn diese Option aktiviert ist, zeigt mAirList unterhalb eines Elements, das gerade in einem Player abgespielt wird, einen weiteren Fortschrittsbalken an, wie er auch in den Player integriert ist. Dieser Balken ist üblicherweise größer als der des Players und deshalb besser abzulesen.
Kommentare anzeigen	Zu jedem Playlist-Element kann ein Kommentar gepflegt werden, zum Beispiel die Anmoderation zu einem Beitrag. Wenn diese Option aktiviert ist, zeigt mAirList diesen Kommentar in der Playlist an.
Automatisch laden	In diesem Modus lädt mAirList automatisch jeden freien Player mit dem nächsten Playlist-Element.
Automatisch entladen bei STOP	In diesem Modus entfernt mAirList automatisch ein Element aus einem Player, wenn dieser ein STOP-Signal vom Benutzer erhält. Zusammen mit der Funktion "automatisch laden" lässt sich so ein Stack-ähnliches Verhalten realisieren, bei dem aufeinanderfolgende Playlist-Elemente einfach durch das Auslösen von Faderstart-Befehlen abgefahren werden können, ohne die Player zwischendurch manuell neu zu laden.
Automatisch entladen bei EOF	In diesem Modus entlädt mAirList einen Player automatisch, wenn er zuende gelaufen ist.
Automation erlauben	Diese Option muss eingeschaltet sein, damit der Automations-Button in der Toolbar sichtbar wird (und die Automation sich aktivieren lässt).

4.2. Player

Für jeden Player jeder Playlist gibt es eine Konfigurationsseite, auf der die folgenden Parameter eingestellt werden können:

Name und Farbe Jedem Player kann ein Name (ein oder zwei Zeichen, z.B. "A", "B", "C", "1", "2", "3", ...) und eine Farbe zugewiesen werden. Die Farbe wird auch in der Playlist benutzt, um optisch zu kennzeichnen, in welchem Player ein bestimmtes Element geladen ist.

Soundkarte Durch Drücken des Buttons "Auswählen" erscheint ein Dialog, in dem die Soundkarte und das Lautsprecherpaar ausgewählt werden kann, auf der die Ausgabe dieses Players erfolgen soll. Als einzige Ausgabemethode wird derzeit die Verwendung der BASS.DLL unterstützt.

PFL-Soundkarte Hier kann festgelegt werden, auf welcher Soundkarte und welchem Kanal die PFL-Ausgabe des Players erfolgen soll. Ist mAirList an ein echtes Mischpult mit Vorhörfunktion angeschlossen, sollte hier dieselbe Soundkarte wie oben ausgewählt werden. Soll man die PFL-Ausgabe auf einem eigenen Monitorkanal erfolgen, so ist die entsprechende Soundkarte hier auszuwählen.

EOF-Warnung Ist hier eine Zahl größer als null eingestellt, blinkt der Player die entsprechende Anzahl von Sekunden vor Ende des Elementes als Warnsignal.

Optionen Folgende Optionen stehen zur Verfügung:

erweitertes PFL Ist diese Option aktiviert, erscheint im PFL-Modus der erweiterte PFL-Dialog, über den Cue-Punkte gesetzt und verändert werden können.

4.3. Cartwall

mAirList beinhaltet eine Cartwall (Jingleplayer), die aus einer beliebigen Anzahl von Playern besteht. Diese können mit Elementen geladen und über Faderstart, Hotkeys oder Maus abgefahren werden.

Cartwall aktivieren Hier kann die Cartwall ein- oder ausgeschaltet werden.

Cartwall im eigenen Fenster anzeigen Ist diese Option aktiviert, so wird die Cartwall in einem eigenen Fenster angezeigt, das über den Button "Cartwall" im Hauptfenster sowie über Faderstart-Kommandos (s.u.) ein- bzw. ausgeblendet werden kann.

Anzahl Player Hier legt man fest, wie viele Player die Cartwall enthalten soll.

Soundkarte, PFL-Soundkarte Genau wie bei den normalen Playern kann hier die zu verwendende Soundkarte (für das Ausspielen und für das Vorhören) ausgewählt werden.

EOF-Warnung Legt fest, ob und wie viele Sekunden vor Ende eines Elementes der Player als Warnung blinken soll.

Favoriten-Belegungen Die Belegung der Cartwall, also die Namen der gerade in den Playern geladenen Dateien, können in sogenannten "Belegungen" gespeichert werden. Hier können die Belegungen ausgewählt werden, die als Favoriten im Cartwall-Fenster erscheinen sollen.

Abgespielte Elemente in Logfiles schreiben Ist diese Option ausgewählt, werden auch die in der Cartwall abgespielten Elemente in die konfigurierten Logfiles (siehe Abschnitt 4.7) geschrieben.

4.4. Browser

Über die Browser lassen sich Verzeichnisse, Playlisten, Datenbanken usw. anzeigen, über die neue Elemente in die Playlisten bzw. die Cartwall eingefügt werden können. Weiterhin lassen sich zusätzliche Elemente in diesem Bereich darstellen, die auf dieser Seite aktiviert werden können: Eine Digital-Uhr, eine LED-Uhr sowie ein Senderlogo (im JPG-Format).

4.5. Faderstart

mAirList unterstützt derzeit – neben der Mausbedienung – drei Möglichkeiten, die Player und weitere Funktionen zu steuern: Über Hotkeys, über den Gameport bzw. einen Joystick oder Gamepad und per MIDI. Alle diese Methoden können gleichzeitig und unabhängig voneinander betrieben werden. Zu jeder Methode gibt es eine Konfigurationsseite, auf der jedem Ereignis (Taste gedrückt/losgelassen, Joystick-Button gedrückt/losgelassen, MIDI-Event) ein Befehl zugeordnet werden kann. Eine Übersicht über die möglichen Befehle und ihre Bedeutung befindet sich in Anhang A.

4.5.1. Joystick/Gameport

mAirList kann die Buttons von Joysticks als Faderstart-Auslöser benutzen. Damit ist es möglich, einen Joystick als Interface zwischen Rechner und Mischpult zu benutzen, entweder durch Ausschachten eines billigen (USB-)Gamepads (Faderstart-Ausgänge des Mischpultes werden an die ehemaligen Button-Kontakte angelötet) oder durch direktes Anschließen an den Gameport (wobei üblicherweise einige Widerstände eingebaut werden müssen, damit Windows die Schaltung als Joystick erkennt).

Das Konfigurationsprogramm zeigt hier alle in der Systemsteuerung installierten Joysticks an. Zu jedem Button dieser Joysticks kann festgelegt werden, welche Aktion jeweils beim Drücken bzw. beim Loslassen des Buttons ausgeführt werden soll.

4.5.2. Hotkeys

Auf dieser Seite können die o.g. Befehle beliebigen Tasten(kombinationen) zugewiesen werden. Dazu zunächst das Feld "Taste" aktivieren, die gewünschte Tastenkombination drücken, dann eine Aktion aussuchen und auf "Hinzufügen" klicken.

Die Hotkeys sind systemweit gültig, reagieren also auch, wenn mAirList gerade nicht im Vordergrund sein sollte. Allerdings werden die Tasten dann nicht an das Programm im Vordergrund weitergereicht. Es sollten also nur Tasten(kombinationen) verwendet werden, die von anderen Anwendungen nicht benötigt werden. Die PFL-Hotkeys werden nur dann aktiviert, wenn tatsächlich gerade ein erweiterter PFL-Dialog angezeigt wird.

4.5.3. MIDI

Neben Joysticks und Hotkeys bietet mAirList zusätzlich die Fernsteuerung per MIDI an. Dazu kann für jedes MIDI-Event eine Aktion definiert werden. Das Konfigurationsprogramm zeichnet die MIDI-Events automatisch auf und zeigt die drei Parameter Status, Data1 und Data2 an. Für Data2 kann zusätzlich festgelegt werden, ob die Aktion ausgeführt werden soll, wenn Data2 kleiner, größer oder gleich dem angegebenen Wert ist. Damit lassen sich z.B. anschlagdynamische Keyboards zum Fernsteuern verwenden, die je nach Stärke des Anschlages einen anderen Data2-Wert schicken.

4.6. HotDirs

Hier können – Zeile für Zeile – Verzeichnisse angegeben werden, die im "Hinzufügen"-Menü über den Browsern (erreichbar durch Klicken auf den kleinen Pfeil) als Favoriten zur Verfügung stehen.

4.7. Logging

mAirList ist in der Lage, Log-Dateien über alle abgespielten Titel zu schreiben. Diese können dann von anderer Software z.B. für Online-Playlisten ausgewertet werden.

Alle gestarteten Titel in ein Logfile schreiben Ist diese Option aktiviert, schreibt mAirList bei jedem gestarteten Titel einen Logfile-Eintrag. Es muss der Dateiname sowie das Logfile-Format angegeben werden. Dazu gibt es eine Reihe von Logfile-Formatvariablen, die in Anhang B beschrieben sind. Auch der Dateiname darf diese Formatvariablen enthalten. So ist es beispielsweise möglich, an jedem Tag ein neues Logfile zu beginnen.

Alle gestarteten Titel in ein Logfile schreiben Wie oben, allerdings wird der Logfile-Eintrag erst am Ende der Ausspielung geschrieben. Zusätzlich stehen nun die Formatvariablen %d und %e zur Verfügung, die die tatsächliche Länge der Ausspielung angeben. Damit lassen sich zum Beispiel leicht Logfiles für die GEMA-Abrechnung erstellen.

Logging in SQL-Datenbank mAirList kann die Log-Einträge auch direkt in eine SQL-Datenbank schreiben. Es werden unterschiedliche Server-Typen unterstützt, zum Beispiel PostgreSQL und MySQL. Die entsprechende Client-Library muss im mAirList-Verzeichnis bzw. im Systempfad vorhanden sein. Neben den Verbindungs-Parametern können zwei INSERT-Befehle angegeben werden, die beim Start bzw. beim Ende eines Elementes an die Datenbank geschickt werden. Die Befehle dürfen wieder die oben erwähnten Formatvariablen enthalten. Soll nur am Anfang bzw. nur am Ende ein Log-Eintrag geschrieben werden, kann das entsprechend andere Befehls-Feld einfach leer gelassen werden.

4.8. Datenbanken

mAirList besitzt Schnittstellen zu einigen Datenbanken, über die Audiodateien in die Playlisten und die Cartwall übernommen werden können. Auf dieser Konfigurationsseite werden die Datenbank-Verbindungen eingerichtet. Es können mehrere Datenbanken gleichzeitig angesteuert werden. Um eine Datenbank hinzuzufügen, klickt man auf den entsprechenden Button und wählt den gewünschten Datenbank-Typ aus. Danach kann man im unteren Teil des Dialoges die benötigten Parameter eingeben und sie durch Klicken auf den "Speichern"-Button sichern.

Im folgenden werden die unterstützten Datenbank-Typen und ihre Parameter beschrieben.

eldoDB eldoDB ist eine MySQL-basierte Musikdatenbank, die speziell für mAirList entwickelt wurde. Die Verwaltung erfolgt mit dem Musiklisten-Tool, das auf der mAirList-Homepage unter www.mAirList.de erhältlich ist. Die folgenden Parameter müssen eingestellt werden:



Protocol	Protokoll bzw. Servertyp des Datenbank-Servers. Üblicherweise muss hier <code>mysql-3.23</code> angegeben werden. Die Versionsnummer bezieht sich hier auf die verwendete Version der Client-Library, <code>libmysql323.dll</code> , die im mAirList-Verzeichnis oder Systempfad vorhanden sein muss. Es können damit auch Server mit einer höheren Versionsnummer (4.0, 4.1, usw.) angesprochen werden.
Host	Hostname oder IP des Datenbankservers
Database	Name der Datenbank auf dem MySQL-Server
User	Benutzername für die Anmeldung
Password	Passwort für die Anmeldung
BaseDir	Basis-Verzeichnis aller Musik-Dateien, mit Backslash am Ende. Die eldoDB speichert intern nur relative Dateinamen, die dann mit diesem Wert verknüpft werden.

radioDB Die radioDB wurde vom Benutzer *cevou* aus dem mAirList-Forum als Alternative zur eldoDB entwickelt. Es wird ein PostgreSQL-Datenbankserver verwendet, das Verwaltungsprogramm selbst ist Web-basiert. Folgende Parameter stehen zur Verfügung:

Protocol	Protokoll bzw. Servertyp des Datenbank-Servers, üblicherweise <code>postgresql-7.3</code> . Es wird die Client-Library <code>libpq73.dll</code> benötigt.
Host	Hostname oder IP des Datenbankservers
Database	Name der Datenbank auf dem PostgreSQL-Server
User	Benutzername für die Anmeldung
Password	Passwort für die Anmeldung
BaseDir	Basis-Verzeichnis aller Musik-Dateien. Wird allen (als relativ angenommenen) Dateinamen vorangestellt.
StripPath	Falls in der Datenbank absolute Dateinamen gespeichert sein sollten, kann mit diesem Parameter der (falsche) absolute Basispfad angegeben werden, der dann zunächst abgeschnitten und später durch das neue BaseDir ersetzt wird.
PathDelimiter	Zeichen, dass im <code>filename</code> -Feld der Datenbank als Trennzeichen zwischen den Verzeichnisnamen verwendet wird. Wenn der Server unter Linux läuft, kann hier <code>/</code> eingetragen werden. mAirList ersetzt das Zeichen dann automatisch im Dateinamen durch einen Backslash.

iTunes iTunes von Apple pflegt alle Musiktitel auf der lokalen Festplatte in einer XML-Datei. mAirList ist in der Lage, diese XML-Datei auszulesen und auf die List der verfügbaren Titel sowie die Playlists zuzugreifen. Als einziger Parameter muss dazu eingestellt werden:

MusicLibrary	Vollständiger Pfad zur iTunes-XML-Datei, üblicherweise: <code>C:\Dokumente und Einstellungen\<benutzername>\Eigene Dateien\Eigene Musik\iTunes\iTunes Music Library.xml</benutzername></code>
--------------	--

4.9. Fernsteuerung

mAirList besitzt eine RPC-/Web-Services-Schnittstelle, über die eine Instanz über das Netzwerk oder von anderen Programmen auf demselben Rechner ferngesteuert werden kann. Auf dieser Konfigurationsseite kann die Schnittstelle aktiviert und der HTTP-Port (Standard: 9300) eingestellt werden, auf dem mit mAirList kommuniziert werden kann.

Die Schnittstelle wurde mit RemObjects SDK (siehe www.remobjects.com) realisiert. Neben dem



RemObjects-eigenen BIN-Format (URL: `http://<host>:<port>/BIN`) unterstützt mAirList auch SOAP-Aufrufe (über die URL `http://<host>:<port>/SOAP`).

Die Fernsteuerung befindet sich zur Zeit noch in der Entwicklung. Erste Anwendung ist das Programm `mAirListComm.exe` (herunterzuladen auf der mAirList-Homepage), mit dem sich beliebige Faderstart-Befehle (s. Abschnitt 4.5) an eine laufende mAirList-Instanz schicken lassen, entweder über das Netzwerk oder auf dem lokalen Rechner.

4.10. BASS.DLL

Auf dieser Konfigurationsseite können zwei Einstellungen die BASS.DLL betreffend gemacht werden.

Puffergröße Hier kann die Größe des Playback-Buffers (Standard: 500ms) verändert werden, was Abhilfe bei Aussetzern verschaffen kann.

Plugins Die BASS.DLL unterstützt von Haus aus die Audioformate MP3, MP2, WAV und OGG. Weitere Formate (z.B. WMA) können mit Hilfe von Add-On-DLLs nachinstalliert werden. Die Add-Ons gibt es als Zip-Datei zum Download unter `http://www.un4seen.com/bass.html`. Jede Zip-Datei enthält eine DLL-Datei, die in das mAirList-Verzeichnis kopiert und durch einen Eintrag in diesem Feld aktiviert werden muss. Es können mehrere, durch Leerzeichen getrennte Dateinamen angegeben werden. mAirList lädt alle angegebenen Plugins beim Programmstart und gibt eine Fehlermeldung aus, falls eine der DLLs nicht gefunden wurde.

4.11. Sonstiges

Automatisches Fenster-Layout Ist diese Option aktiv, ordnet mAirList die Playlists und Player automatisch im Programmfenster an: Unten nebeneinander die Player aller Playlists, darüber links übereinander alle Playlists sowie eventuell die Cartwall, rechts der Browser. Alternativ kann das Fenster-Layout auch von Hand festgelegt werden, über die Datei `layout.ini` (s. Anhang C).

Alte Browser-Darstellung Hier kann eingestellt werden, ob die Browser in der neuen ("Outlook"-ähnlichen) oder alten Optik erscheinen sollen.

Bei mehreren Monitoren: mAirList starten auf Bildschirm ... Wenn der Rechner mehrere Bildschirme hat, kann hier festgelegt werden, auf welchem der Monitore mAirList angezeigt werden soll. So kann man zum Beispiel zwei Kopien von mAirList (z.B. eine für Musik, eine für Jingles) auf demselben Rechner (mit DualHead-Grafikkarte) betreiben und von vornherein festlegen, welche Kopie auf welchem Bildschirm laufen soll.

Zusätzliche Dateierweiterungen Durch Leerzeichen getrennte List der Dateierweiterungen (ohne führenden Punkt), die mAirList in den Datei-Browsern anzeigen soll, zusätzlich zu `wav`, `mp3`, `mp2` und `ogg`.

PFL OFF bei START Einige Mischpulte mit Faderstart-Funktion sind in der Lage, beim Aktivieren des Kanals gleichzeitig das PFL mit auszuschalten. Diese beiden Signale (PFL OFF, START) kommen in mAirList eventuell in der falschen Reihenfolge an (START vor PFL OFF), so dass mAirList den Player nicht startet, weil er ja noch im PFL-Modus ist. Wenn man diese Option aktiviert, wird jeder Player softwareseitig automatisch aus dem PFL-Modus genommen, wenn er ein START-Signal bekommt. Dies gilt auch für den Betrieb über Hotkeys und MIDI.

Soundkarte für Extra-PFL Die normale Vorhörfunktion (PFL) von mAirList spielt das Element immer auf der Soundkarte aus, die auch für die normale Wiedergabe des Players verwendet wird. Damit ist eine nahtlose Kopplung an die PFL-Funktion von Mischpulten möglich. Zusätzlich kann hier ein weiterer Ausgabekanal definiert werden, auf dem Elemente im sogenannten Extra-PFL-Modus ausgegeben werden, unabhängig davon, ob sie gerade in einem Player geladen sind oder nicht. Diese Funktion erreicht man über das Kontextmenü, das man durch Klicken mit der rechten Maustaste auf ein Element erhält.

Fade-Zeit für die Automation Diese Wert legt fest, wie lange im Automations-Betrieb der letzte Titel ausfaden soll, während der neue bereits läuft.

Meldungen anzeigen in Sprache ... Wenn Dateien zur Übersetzung von mAirList in andere Sprachen installiert sind, können diese hier aktiviert werden.

5. mAirList benutzen

5.1. Das Hauptfenster

Das mAirList-Hauptfenster gliedert sich je nach Konfiguration in drei oder vier Teile:

- die Playlist(en),
- die Browser,
- die Player,
- ggf. die Cartwall.

5.1.1. Die Playlist

In der Playlist sind alle Dateien aufgelistet, die zum Abspielen vorgesehen sind. Im Split Mode ist nicht nur eine Playlist sondern mehrere sichtbar, je nach Konfiguration. Wie genau man mit der Playlist umgeht, steht in Abschnitt 5.2.

5.1.2. Die Browser

Im rechten Bereich des mAirList-Fensters können sogenannte "Browser" eingeblendet werden. Über diese können beispielsweise Verzeichnisse eingeblendet werden, aus denen dann neue Dateien in die

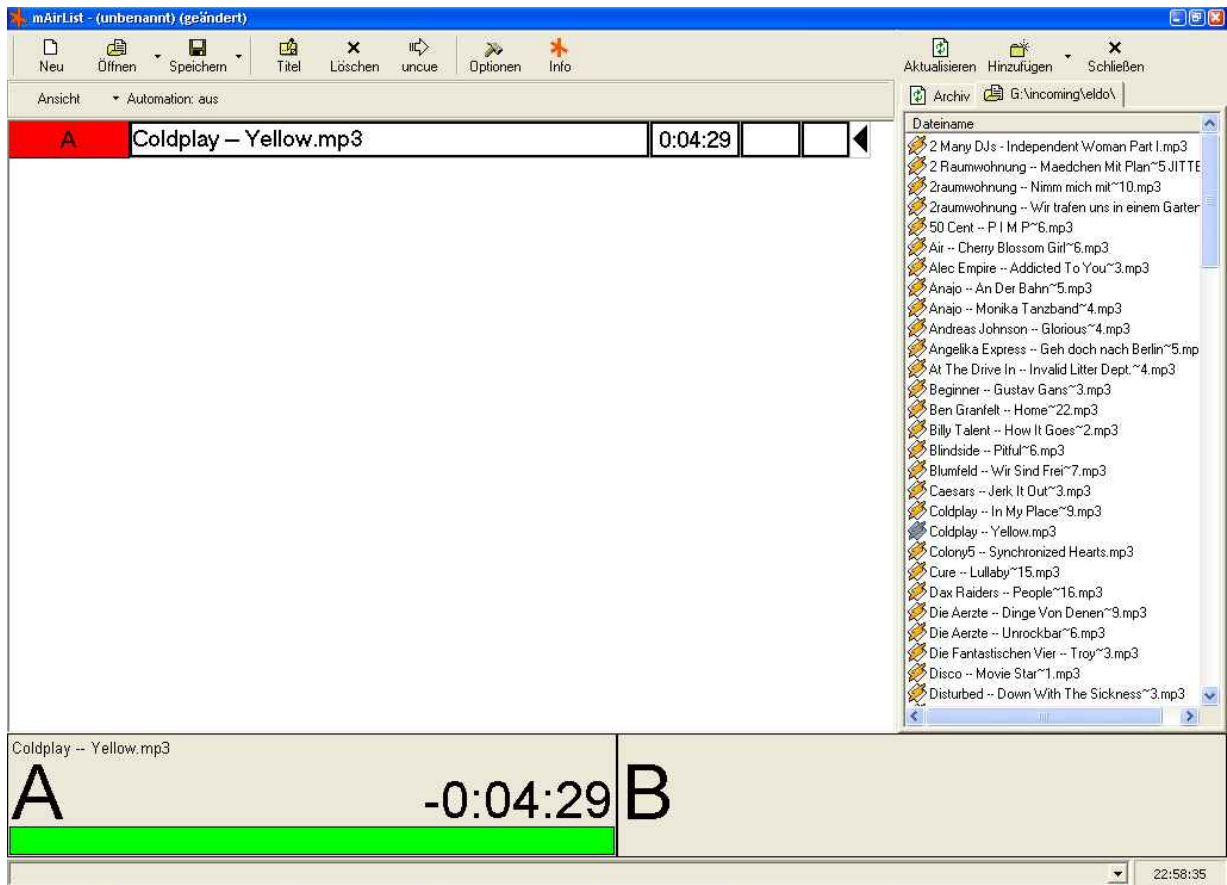


Abbildung 2: Das mAirList-Hauptfenster

Playlist gezogen werden können. Welche Browser zur Verfügung stehen und wie diese funktionieren, wird in Abschnitt 5.3 erklärt.

5.1.3. Die Cartwall

Die Cartwall enthält die Cartplayer, über die beispielsweise Jingles abgefahren werden können. Die Cartwall ist entweder in das Hauptfenster integriert (zwischen den Playlisten und den Playern) oder wird in einem eigenen Fenster dargestellt, das über die konfigurierten Faderstart-Kommandos ein- und ausgeblendet werden kann. Die Bedienung der Cartwall ist in Abschnitt 5.5 beschrieben.

5.2. Arbeiten mit der Playlist

5.2.1. Einfügen von Elementen

Neue Elemente (Dateien) können ganz einfach per Drag & Drop in die Playlist gezogen werden, und zwar entweder aus dem Windows Explorer oder aus einem der Browser. Beim Einfügen in die Playlist

wird die Abspieldauert der Datei ermittelt. Fehlerhafte Dateien werden erst beim Laden in einen Player erkannt.

5.2.2. Verschieben von Elementen

Elemente können einfach mit der Maus von oben nach unten oder umgekehrt verschoben werden. Dazu muss das Element in der ersten Spalte (ganz links) angeklickt werden und dann mit gedrückter Maustaste verschoben werden.

5.2.3. Löschen von Elementen

Es gibt zwei Arten, auf die ein Element aus der Playlist entfernt werden kann:

1. Verschieben in den Archiv-Browser (s. Abschnitt 5.3.1)
2. direktes Löschen

Dazu dienen die beiden Buttons "uncue" und "löschen" in der Symbolleiste.

5.2.4. Playlist-Optionen

Das Kontextmenü einer Playlist (erreichbar über die rechte Maustaste) enthält einige Optionen, die insbesondere die Darstellung der Playlist betreffen.

Eigenschaften Über diese Funktion wird ein Fenster geöffnet, in dem das aktuelle Element bearbeitet werden kann, also insbesondere Titel und Interpret. Außerdem können die eingestellten Cue-Punkte in die Datenbank zurückgeschrieben oder im ID3-Tag einer MP3-Datei gespeichert werden. Ferner kann hier eine Fix-Zeit eingestellt werden, die für die Berechnung des Backtimings herangezogen wird (aber für die Automation derzeit ohne Bedeutung ist).

PFL Hierüber wird die Extra-PFL-Funktion für das bearbeitete Element gestartet. Dazu öffnet sich ein Fenster mit einem zusätzlichen Player und denen aus dem erweiterten PFL-Modus (s. Abschnitt 5.4.4 bekannten Dialogelementen. Die Wiedergabe erfolgt auf der im Konfigurationsprogramm eingestellten Soundkarte (s. Abschnitt 4.11).

Erweiterte Anzeige Schaltet zwischen einfacher und erweiterter Anzeige um (s. Abschnitt 4.1).

Backtiming-Anzeige Diese Funktion aktiviert die Backtiming-Anzeige (s. Abschnitt 4.1).

Player-Fortschritt anzeigen Diese Option aktiviert den Fortschrittsbalken innerhalb der Playlist (s. Abschnitt 4.1).



Player automatisch laden Hiermit kann festgelegt werden, ob mAirList das Laden/Entladen von Playern automatisch vornehmen soll (s. Abschnitt 4.1).

Automation Dieser Menüpunkt aktiviert die Automation für diese Playlist. Mehr dazu steht in Abschnitt 5.8.

5.3. Arbeiten mit Browsern

Die Browser bilden das Herzstück von mAirList, wenn es darum geht, neue Dateien in die Playlist einzufügen. Über die Buttons "Hinzufügen" und "Schließen" können jederzeit neue Browser angezeigt oder ausgeblendet werden. Der Button "Aktualisieren" aktualisiert den Inhalt eines Browsers. Elemente aus den Browsern können per Drag & Drop in die Playlist gezogen werden.

Es gibt verschiedene Arten von Browsern, die im folgenden kurz erklärt werden sollen. Ein einfacher Klick auf den "Hinzufügen"-Button öffnet einen Verzeichnisbrowser, alle anderen Arten von Browsern (außer dem Archiv-Browser, der immer angezeigt wird) können über das Menü aufgerufen werden, das erscheint, wenn man auf den Pfeil neben "Hinzufügen" klickt.

5.3.1. Der Archiv-Browser

Der Archiv-Browser wird immer angezeigt (kann also nicht geschlossen werden). Wenn ein Element abgespielt und automatisch aus der Playlist entfernt wurde, landet es danach im Archiv. Das Archiv enthält also alle bislang abgespielten Dateien. Dadurch ist ein schneller Zugriff möglich, wenn eine Datei versehentlich aus der Playlist gelöscht wurde oder nicht.

5.3.2. Der Verzeichnis-Browser

Der Verzeichnis-Browser listet den Inhalt eines Verzeichnisses auf, wobei nur Sounddateien (MP3, MP2, WAV, OGG) angezeigt werden. Welches Verzeichnis der Browser darstellen soll, legt man beim Hinzufügen des Browsers fest.

5.3.3. Der Verzeichnisbaum-Browser

Dieser Browser arbeitet wie der Verzeichnis-Browser, wobei allerdings ein ganzer Verzeichnisbaum dargestellt wird, ähnlich wie in der linken Hälfte des Windows-Explorers. Verzeichnisse können über die kleinen "+"- und "-"-Buttons auf und zugeklappt werden. Es wird immer der beim Hinzufügen festgelegte Ordner und alle Unterverzeichnisse angezeigt.

5.3.4. Der Playlist-Browser

In diesem Browser können Playlisten im Winamp- (.m3u) bzw. Raudga- (.alb, .mpl) Format angezeigt werden.

5.3.5. Der Datenbank-Such-Browser

Falls aktiv, dient dieser Browser zum Suchen in der eldoDB. Es kann entweder nach Interpret/Titel gesucht oder eine komplette Rotation angezeigt werden.

5.3.6. Der Datenbank-Playlist-Browser

Dieser Browser dient zum Anzeigen einer in der eldoDB angelegten Playlist. Elemente der eldoDB-Playlist, zu denen kein MP3-Dateiname festgelegt wurde, werden automatisch ausgeblendet.

5.3.7. Der Uhr-Browser

Dieser Browser stellt eine große Uhr da. Vielleicht für den einen oder anderen ganz nützlich ;-)

5.4. Arbeiten mit den Playern

Am unteren Fensterrand befinden sich die (bis zu drei) Player, wie im Konfigurationsprogramm eingestellt.

5.4.1. Player laden

Normalerweise ist mAirList so konfiguriert, dass die Player automatisch mit den obersten Elementen in der Playlist geladen werden. Ist diese Option deaktiviert – entweder über das Konfigurationsprogramm (s. Abschnitt 4.1) oder über das Kontextmenü der Playlist – muss man die Elemente von Hand per Drag&Drop in den gewünschten Player ziehen.

5.4.2. Spielzeit- und Backtiming-Anzeige

Im Normalfall zeigt der Player die noch verbleibende Restzeit der Datei an. Durch einen Klick auf die Zeit kann zwischen Restzeit und aktueller Position umgestellt werden. Ein Doppelklick zeigt die Uhrzeit an, zu der die Datei zuende ist (bzw. zuende wäre, würde sie in diesem Moment gestartet).

5.4.3. Starten und Stoppen von Playern

Die Player werden normalerweise über die Hotkeys bzw. den Faderstart angesteuert, wie im Konfigurationsprogramm eingestellt. Die Bedienung mit der Maus ist zur Zeit nicht möglich.

5.4.4. Der erweiterte PFL-Modus

Wenn im Konfigurationsprogramm bzw. im Options-Menü so eingestellt, erscheint im PFL-Modus automatisch ein Fenster, über das die zu einem Element gehörigen Zeiten (Cue, Ramp, Outro, Fadeout) festgelegt werden können.



Abbildung 3: Der erweiterte PFL-Modus

Die Cue-Zeit legt fest, ab welchem Zeitpunkt der Titel abgespielt werden soll. Damit lassen sich z.B. lange Intros von Liedern abschneiden.

Die Ramp kann auf den Zeitpunkt gesetzt werden, an dem der Gesang des Liedes anfängt. Der Moderator hat also zwischen Cue- und Ramp-Punkt Zeit zum Sprechen. Die Ramp wird dem Moderator auch graphisch angezeigt (s.u.).

Outro ist zur Zeit noch ohne Funktion.

Fadeout legt – für die Automation – fest, an welchem Punkt der Titel ausgeblendet und der nächste gestartet werden soll.

PAUSE und PLAY Diese beiden Buttons halten die Wiedergabe an bzw. starten sie wieder.

END MON Dieser Button spielt die letzten 10 Sekunden des Titels ab.

SET Dieser Button setzt die zugehörige Zeit (Cue, Ramp, Outro oder Fadeout) auf den aktuellen Zeitpunkt. Damit kann z.B. ein Cue-Punkt zunächst grob eingestellt werden.

0 Dieser Button setzt die zugehörige Zeit wieder auf Null zurück. Bei Outro und Fadeout bedeutet 0, dass keine Zeit festgelegt wurde.

TEST Dieser Button testet die entsprechende Zeit, der Player springt also zu dem festgelegten Punkt.

+ und - Über diese Buttons kann die Zeit in 1/100-Sekunden-Schritten verstellt werden.

Bedienung mit der Maus Die Zeiten können auch mit dem Mousrad verändert werden, ein Schritt des Mousrades entspricht 1/100 Sekunden. Standardmäßig wird die Cue-Zeit verändert, ansonsten die Zeit, für die zuletzt ein Button gedrückt wurde.

5.4.5. Visualisierung der Ramp

Die Ramp ist im Radio-Jargon der (instrumentale) Beginn eines Liedes, auf den der Moderator noch sprechen kann, ehe der Gesang beginnt. mAirList unterstützt den Moderator dabei, indem die Ramp-Zeit – sofern sie über das erweiterte PFL eingestellt wurde – graphisch visualisiert wird. Innerhalb der Ramp teilt sich der sonst grüne Balken des Players. Die obere Hälfte wird rot und läuft zusammen mit der Ramp langsam runter. Außerdem wird die verbleibende Restzeit der Ramp im Player angezeigt.

5.5. Arbeiten mit der Cartwall

[...]

5.6. Öffnen und Speichern von Dateien

mAirList besitzt ein eigenes XML-basiertes Dateiformat, in dem der Zustand des Programms gespeichert und später wiederhergestellt werden kann. Dazu zählen

- Inhalt der Playlisten,
- Inhalt der Cartwall,
- Art und Inhalt der geöffneten Browser.

Die Endung der mAirList-Dateien lautet “.mp!”, wie “mAirList Playlist”².

5.6.1. Vorlagen

Eine mAirList-Datei kann auch als Vorlage gespeichert werden. Das Format ist dasselbe, allerdings lautet die Endung dann “.mlt” (“mAirList Template”). Wird eine solche Datei später geöffnet, “vergisst” mAirList die Dateinamen, so dass der einfache “Speichern”-Button nicht funktioniert. Vorlagen sind so also gegen das versehentliche Überschreiben gesichert. Soll eine Vorlage überschrieben werden, muss das explizit über “Speichern unter ...” geschehen.

5.6.2. Standard-Vorlage

Beim Start sucht mAirList im Programmverzeichnis nach einer Datei `standard.mlt`. Ist diese vorhanden, wird sie geöffnet. So lassen sich Default-Einstellungen bei jedem Programmstart automatisch laden.

²“mlp” wäre auch denkbar gewesen, hätte mich aber zu sehr an nervende Vertreter erinnert.

5.6.3. Import von Fremdformaten

mAirList kann folgende Dateiformate importieren:

- .xpl (mAirList 1.0)
- .m3u (Winamp)

Der Import geschieht ganz normal über den Öffnen-Dialog.

5.6.4. Anhängen von Dateien

Dieser Befehl verbirgt sich hinter dem kleinen Pfeil neben dem "Öffnen"-Button. Über diese Funktion können Dateien an die aktuelle angehängt werden, das heißt, die Elemente der Playlisten werden an die aktuellen Playlisten angehängt.

5.7. Tastaturbedienung

Die wichtigsten Funktionen von mAirList lassen sich auch über die Tastatur aufrufen:

Strg-O	Playlist öffnen
Strg-A	Playlist anhängen
Strg-S	Playlist speichern
Strg-T	Playlist unter anderem Namen speichern
Strg-E	Element bearbeiten
Entf	Element ins Archiv verschieben
Shift-Entf	Element aus Playlist löschen

Darüber hinaus können mit dem Konfigurationsprogramm Hotkeys für die Player-Funktionen frei definiert werden (s. Abschnitt 4.5).

5.8. Die Automation

[ToDo]

A. Faderstart-Befehle

Zur Zeit stehen die folgenden Faderstart-Befehle zur Verfügung, die Keyboard-, MIDI- sowie Joystick/Gameport-Ereignissen zugeordnet werden können:

- `PLAYER x-y START` – startet den y-ten Player der x-ten Playlist.
- `PLAYER x-y PAUSE` – hält den y-ten Player der x-ten Playlist an. Der Player bleibt an der aktuellen Position stehen, mit `START` kann die Ausgabe fortgesetzt werden. Geht man in den PFL-Modus, springt der Player zurück an den eingestellten Cue-Punkt.

- `PLAYER x-y STOP` – stoppt den y-ten Player der x-ten Playlist. Je nach Konfiguration wird dann entweder das Element aus der Playlist entfernt oder der Player zurück an den Cue-Punkt gefahren.
- `PLAYER x-y START/STOP` – wenn Player y gerade nicht läuft, wird er gestartet, sonst gestoppt.
- `PLAYER x-y START/PAUSE` – wenn Player y gerade nicht läuft, wird er gestartet, sonst angehalten.
- `PLAYER x-y PAUSE/STOP` – wenn Player y gerade läuft, wird er angehalten. Ist er bereits zuende gelaufen (EOF), wird er gestoppt.
- `PLAYER x-y START/PAUSE/STOP` – wenn Player y gerade nicht läuft, wird er gestartet; wenn er läuft, wird er angehalten; ist er am EOF, wird er gestoppt.
- `PLAYER x-y PFL ON, OFF, ON/OFF` – startet oder stoppt die Vorhörfunktion, wie oben.
- `PFL PLAY, PAUSE, END MON` – entspricht den Buttons `PLAY/PAUSE/END MON` im erweiterten-PFL-Dialog
- `PFL CUE PLUS, MINUS, SET, 0, TEST` – entspricht den Buttons `+/-/SET/0/TEST` der CUE-Einstellung im erweiterten PFL-Dialog.
- `PFL RAMP PLUS, MINUS, SET, 0, TEST; PFL OUTRO PLUS, MINUS, SET, 0, TEST; PFL FADEOUT PLUS, MINUS, SET, 0, TEST` – dasselbe für Ramp, Outro und Fadeout.
- `AUTOMATION x ON, OFF, ON/OFF` – aktiviert oder deaktiviert die Automation für Playlist x.
- `AUTOMATION x START, STOP, NEXT` – entspricht den Automations-Buttons von Playlist x.
- `CARTWALL SHOW, HIDE, SHOW/HIDE` – öffnet bzw. schließt das Cartwall-Fenster bzw. wechselt zwischen geöffnet und geschlossen (nur für den Fall, dass die Cartwall in einem eigenen Fenster ist).
- `CARTWALL NEXT SET, PREVIOUS SET` – lädt die nächste bzw. vorherige Belegung aus der Liste der definierten Favoriten.
- `CARTWALL STOP ALL` – stoppt alle Player in der Cartwall.
- `CARTWALL CLEAR ALL` – entlädt alle Player in der Cartwall.
- `CARTWALL x START` – startet Cartplayer x.
- `CARTWALL x STOP` – stoppt Cartplayer x.
- `CARTWALL x START/STOP` – startet Cartplayer x bzw. stoppt ihn, wenn er gerade läuft.
- `CARTWALL x CLEAR` – entlädt Cartplayer x.

B. Logfile-Variablen

Die folgenden Variablen können für Logfile-Einträge verwendet werden:

%Y	aktuelles Jahr (vierstellig)
%M	aktueller Monat (zweistellig)
%D	aktueller Tag (zweistellig)
%h	aktuelle Stunde (zweistellig)
%m	aktuelle Minute (zweistellig)
%s	aktuelle Sekunde (zweistellig)
%1	Dateiname mit vollem Pfad
%2	Dateiname ohne Pfad
%3	Dateiname ohne Pfad und Dateierweiterung
%a	Interpret
%b	Titel
%d	Dauer der Ausspielung (nur bei Stop-Logging) in mAirList-Einheiten
%e	Dauer der Ausspielung (nur bei Stop-Logging) in Sekunden
%c	Cue-In-Position (in mAirList-Einheiten)
%r	Ramp (in mAirList-Einheiten)
%o	Outro-Position (in mAirList-Einheiten)
%f	Fade-Out-Position (in mAirList-Einheiten)
%E	Ende-Typ
%t	Tabstop

Ein mAirList-Einheit entspricht einer 10-Millionstel-Sekunde.

C. Fenster-Layout anpassen

Wenn im Konfigurationsprogramm die Option "Automatisches Layout" ausgeschaltet ist, können die graphischen Bestandteile von mAirList (Player, Playlists, Browser, ...) pixelgenau nach den eigenen Vorstellungen platziert sowie optisch angepasst werden. Dies geschieht über eine weitere Konfigurationsdatei, `layout.ini`. Durch Austausch dieser Datei kann mAirList schnell optisch an verschiedene Gegebenheiten angepasst werden.

C.1. Aufbau von `layout.ini`

Die Datei `layout.ini` ist wie eine herkömmliche Windows-INI-Datei aufgebaut. Zu jedem graphischen Element gibt es einen Abschnitt, der durch eckige Klammern und den Namen des Elements eingeleitet wird, z.B. `Playlist0`. Je nach im Konfigurationsprogramm gewählten Optionen stehen die folgenden Abschnitte zur Verfügung:

- `[Toolbar]` – die Toolbar mit den Buttons "Neu", "Öffnen", "Speichern" usw.
- `[Browser]` – der Datei-Browser, incl. Toolbar "Hinzufügen", "Aktualisieren" und "Schließen".
- `[Cartwall]` – die Cartwall bzw. das Cartwall-Fenster, sofern entsprechend konfiguriert.
- `[Playlist x]` – die x -te Playlist (wobei die erste Playlist die Nummer 0 hat, die zweite die Nummer 1 usw.)
- `[Player x _ y]` – der y -te Player der x -ten Playlist (wobei die erste Player wieder die Nummer 0 hat, der zweite die Nummer 1 usw.)

- [Logo] – das Senderlogo (falls aktiviert).
- [Clock] – die Uhr (falls aktiviert).
- [LEDClock] – die LED-Uhr (falls aktiviert).

C.2. Beschreibung der Parameter

C.2.1. Gemeinsame Parameter aller Elemente

Jedes graphische Element verfügt über die folgenden Parameter:

- Left – X-Koordinate der linken oberen Ecke
- Top – Y-Koordinate der rechten oberen Ecke
- Width – Breite des Elements
- Height – Höhe des Elements

Ist die Cartwall so konfiguriert, dass sie in einem eigenen Fenster erscheint, lassen sich über diese Parameter Größe und Position dieses Cartwall-Fensters festlegen. Ansonsten beeinflussen sie die Position des Cartwall-Bereichs im Hauptfenster.

C.2.2. Zusätzliche Parameter von [Playlist*x*]

Für jede Playlist lassen sich weiterhin die folgenden Werte einstellen:

- FontName – zu verwendende Schriftart.
- FontSize – zu verwendende Schriftgröße.
- FontStyle – zu verwendender Schriftstil. Dieser ergibt sich durch Addition der folgenden Werte:
 - 1 fett
 - 2 kursiv
 - 4 unterstrichen
 - 8 durchgestrichen
- PlayerNameFontName, PlayerNameFontSize, PlayerNameFontStyle – wie oben, Schriftart für den Namen und Zustand des Players in der linken Spalte.
- BacktimingFontName, BacktimingFontSize, BacktimingFontStyle – wie oben, Schriftart für die Backtiming-Anzeige.
- CommentFontName, CommentFontSize, CommentFontStyle – wie oben, Schriftart für den Kommentar des Playlisten-Elementes.
- BackgroundColor – Hintergrundfarbe.
- RowColor – Standardfarbe eines Playlist-Eintrages.

- `SelectedRowColor` – Farbe des ausgewählten Playlist-Eintrages.
- `LoadedRowColor`, `PlayingRowColor`, `PFLRowColor`, `PausedRowColor`, `EOFRowColor` – Farbe eines Playlist-Eintrages in Abhängigkeit vom Zustand: In Player geladend, wird gerade abgespielt, PFL, EOF, angehalten, Fehler.
- `SelectedLoadedRowColor`, `SelectedPlayingRowColor`, `SelectedPFLRowColor`, `SelectedPausedRowColor`, `SelectedEOFRowColor` – wie oben, für Einträge, die zusätzlich noch mit der Maus ausgewählt sind.
- `BorderColor` – Farbe des Rahmens um die Playlisterträge herum.
- `BorderWidth` – Breite des Rahmens.
- `ProgressBarHeight` – Höhe des Fortschrittsbalkens.
- `ProgressBarBorderColor` – Farbe des Fortschrittsbalkens: Rahmen.
- `ProgressBarElapsedColor` – Farbe des Fortschrittsbalkens: abgelaufene Zeit.
- `ProgressBarRemainColor` – Farbe des Fortschrittsbalkens: verbleibende Zeit.
- `ProgressBarRampColor` – Farbe des Fortschrittsbalkens: Ramp.
- `ColWidths` – Spaltenbreiten in Pixel. Als Wert werden sechs, durch Komma getrennte Zahlen erwartet. Wird als Spaltenbreite 0 Pixel angegeben, wird die Spalte ausgeblendet. Beispiel: 150,400,0,0,0,30

Alle Farben können als HTML-Farbcodes (mit führender Raute) angegeben werden, z.B. #F0F0F0.

C.2.3. Zusätzliche Parameter von `[Playerx.y]`

Für jeden Player lassen sich weiterhin die folgenden Werte einstellen:

- `TitleFontName`, `TitleFontSize`, `TitleFontStyle` – wie oben. Definiert die Schriftart, die für den Namen des gerade geladenen Elements verwendet werden soll.
- `NameFontName`, `NameFontSize`, `NameFontStyle` – wie oben. Definiert die Schriftart, die für den Namen des Players verwendet wird.
- `StateFontName`, `StateFontSize`, `StateFontStyle` – wie oben. Definiert die Schriftart, die für den aktuellen Status des Players ("ON AIR", "PFL", ...) verwendet wird.
- `TimeFontName`, `TimeFontSize`, `TimeFontStyle` – wie oben. Definiert die Schriftart, die für die Zeitanzeige des Players verwendet wird.
- `ProgressBarHeight`, `ProgressBarBorderColor`, `ProgressBarElapsedColor`, `ProgressBarRemainColor`, `ProgressBarRampColor` – wie oben.
- `ProgressBarMargin` – definiert den Abstand zwischen Laufbalken und der darüberliegenden Schrift.
- `EmptyColor`, `LoadedColor`, `PlayingColor`, `FadeColor`, `EOFCColor`, `PFLColor`, `PausedColor`, `ErrorColor` – legt die Hintergrundfarbe des Players in den verschiedenen Zuständen fest: Leer, geladen, spielend, ausblendend, EOF, PFL, angehalten, Fehler.

- `FlashEOFColor` – legt die Farbe fest, in der der Player (im Wechsel mit `PlayingColor` oder `FadeColor`) blinkt, um auf das Ende des Elements hinzuweisen (sofern die entsprechende Option aktiviert ist).

C.2.4. Zusätzliche Parameter von `[CartWall]`

Im Abschnitt `[Cartwall]` können dieselben Parameter benutzt werden wie im Abschnitt `[Playerx_y]`, um die Darstellung der Cartplayer zu beeinflussen. Die Einstellungen wirken sich auf alle Player gleichzeitig aus.

C.2.5. Zusätzliche Parameter von `[Clock]`

Über die Parameter `FontName`, `FontSize`, `FontStyle` kann die Schriftart der Uhr festgelegt werden.

D. Kommandozeilen-Parameter

mAirList kennt zur Zeit die folgenden Kommandozeilen-Parameter:

- c `<Dateiname>` Benutzt `<Dateiname>` als Konfigurationsdatei anstelle von `mAirList.ini`
- l `<Dateiname>` Benutzt `<Dateiname>` als Layout-Konfigurationsdatei anstelle von `layout.ini`
- o `<Dateiname>` Öffnet bei Programmstart die Datei `<Dateiname>`