AccessiblePlayer

Jail

COLLABORATORS			
	TITLE : AccessiblePlayer		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Jail	March 28, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AccessiblePlayer

1.1 Root

Welcome to the Amigaguide®d world of AccessiblePlayer:

Please select one of the following topics:

Featurelist	List of cool features
Disclaimer	Legal Information
Requirements	What you need
Introduction	Introduction to APlayer
How to Install	How to install APlayer
ToolTypes/CLI	Tooltypes and CLI arguments
How to use	How to use APlayer
Extra features	Things we didn't know where to put
Programmers	Coders, read this
Thanks to	I want to thank my producer, and mom & dad...
History	The history of APlayer
Contact	How to reach the author
Module Types	Module types supported by APlayer

1.2 features

Special APlayer Features:

- Supports 26 different module formats.

- Recognizes and unpacks 150 different cruncher types (using the Unpack.library).

- Favourite Song System©

- Play samples by the keyboard while listening to the module.

- Save samples can be used to rip your favourite samples, even if you don't own the actual music program.

- Can play modules from Fast memory, saving your sparesome Chip memory.

- App-Icon and App-window.

- Full commodity interface. Depending on your system.

- Supports Lha files, including extract pattern.

- Supports Public Screens.

- Make your own favourite lists of modules with the APML© system.

- Early load system which kills the waiting time between the modules.

- And much, much more...

1.3 disclaimer

Copyright information:

AccessiblePlayer is Public Domain. However parts of the APlayer is copyright by other people (the extern players). Whereever possible we have tried to make sure that no parts of the APlayer violates any copyrights. If you think this is not the case, please contact us immediately!

Although this program is PD, you are welcome to send anything, like money, cannabis, postcards, cars, burgers (preferably McChicken!?!), Kinder Mælkesnitte etc.

1.4 requirements

Hard'n'soft-ware requirements:

You need at least kickstart v2.04 to use this program, but preferably 3.0+ because the listview gadgets looks like hell on 2.0.

Two general libraries are necessary to run the APlayer with all functions. APlayer can't start without the Reqtools.library, but if the commodities.library isn't available, it will just not start as a commodity (This means that you can't use the hotkeys).

If you want the APlayer to unpack your modules, you should use one of the following libraries: Powerpacker.library, Xpkmaster.library or the Unpack.library. All of them is included in this package though.

The APlayer has been successfully tested on the following machines:

Plain A1200, A1200+4MB fastram, A1200+68030+4MB Fastram, A500+2MB Fastram and an A4000(680EC30)+16MB Fastram including test with Enforcer, Mungwall and IOtorture.

1.5 introduction

Introduction:

The AccessiblePlayer is our contribution to the Amiga market of music players. We thought that our favourite player (Smartplay) just wasn't good enough anymore, and since neither D.A.S. Modplayer nor the Delitracker fulfilled our needs we decide to write our own player.

We hope we have reached our goal by making a small, fast and userfriendly player, the AccessiblePlayer.

1.6 installation

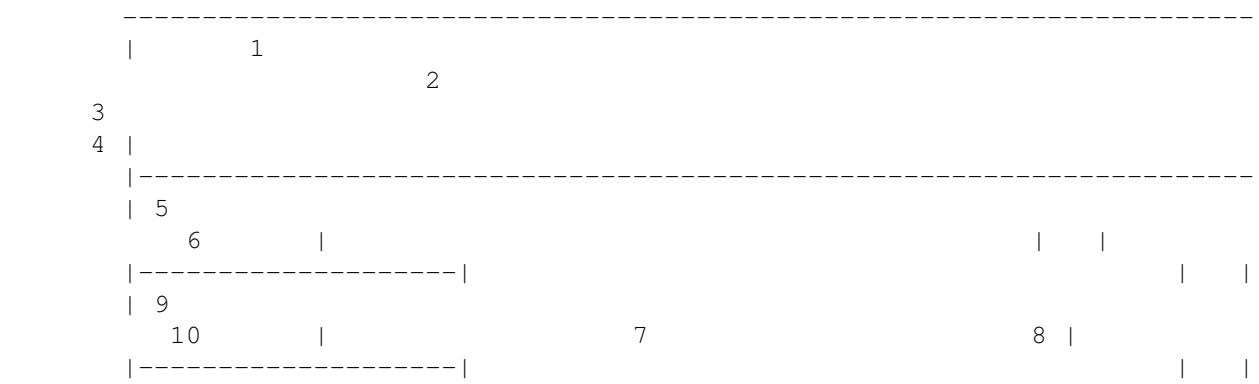
Installation:

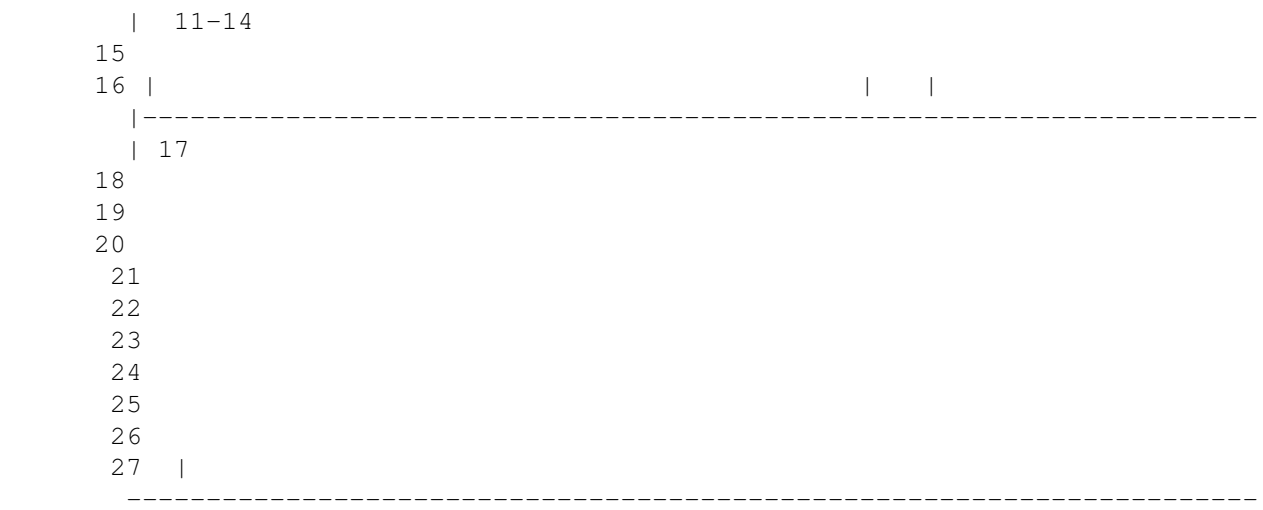
Just use the installer-script included in this package. You can both install AND remove (who would?) the Aplayer with that script.

If you, for some strange reason doesn't like the commodore installer, you can just copy the Libs/Aplayer directory to your libs: and the main program to your favourite path.

1.7 usage

The Accessible Interface:





App-Window: The main window of APlayer can be used for dropping module-icons. The names of the modules will automatically be added to the end of the modulelist. See also App-Icon.

1.8 maincyc

Main window cycle:

This cyclegadget is used to select the state of the status-bar.

There are four different states of this cycle, which are:

Name :
This shows the name of the module, which is found IN the module. e.g it is not necessarily the name of the file, so the two names can be different, that is, if the player can't find a name inside the module, the filename will be displayed in the status bar.

Author :
The player tries to find the name of the author in the module, And if it succeeds it will display it in the statusline. See in the section Module types for info on which players supporting the author name.

Played :
This shows, if supported (see Module types), the actual position and the total songlength. It do also calculate how many % of the module that has been played, and the total listening time.

Type :
When the module has been tested, the module type is displayed in the status-line. This could be anything from the SID-player to the TFMX 7 channel replayer.

1.9 status

@

Status Bar:

This line will display some info on the current module. It will display the name, author, playing time and moduletype. See Main window cycle on how to change the state.

1.10 sampleinfo

Sample info window: (Shortkey s)

Pressing the S-Button will, if samples are supported (see module types) open a new window called the sampleinfo window. From left to right it will display the number of the sample, name, size in bytes and wether it's placed in chip-, or fast-ram.

The sampleinfo window is not only a display/info window, you can actually use it to accompany your favourite modules. You can do it in two ways.

1. Pause the module (in the main window) and use your mouse, or the keypad, to select a sample. Then you can use your keyboard (like in Protracker) to play the sample at different notes, ranging from C-1 to B-3. Like in protracker you can use F1 and F2 to change wether the z-key should be a C-1 (default) or a C-2 note. Note that if a sample has the volume set to 0, then it will be played with a value of 64.

2. Turn off one of the sound-channels and like before choose a sample with the keypad or with the mouse. Then use the keyboard to accompany the module. Sorry no multikeyboard yet...

The last function in the sampleinfo window is the little disk-gadget in the upper right corner of the window. Selecting a sample and pressing the disk-gadget will open a requester which will let you save the sample as an 8SVX soundfile with the samplerate 16574 (in pal) or 16726 (in ntsc) which is the C-3 note.

Instead of using the keypad or the mouse, you can cycle through samples with the cursor up/down keys. Remember that this can only be done if a sample is selected.

If you want to stop the playing sample NOW! then you can hit the "<" which is placed on the right side of the left shift key.

NOTE: Not all players supports the sample info window, and therefore not the sample-save and accompany functions. (see Module types for more info)

1.11 keypad

Numeric Keypad:

This works exactly like in QuadraComposer. Sample 1-16 is chosed from the upper left corner on the numpad to the lower right corner, which is the Enter-key. "0" is used to jump forward 16 samples, and "." to go back to the previous 16 samples.

Note that pressing a keypad key, will trigger the sample at the note C-3. And that is not changeable yet, sorry!

Note that if a sample has the volume set to 0, then it will be played with a value of 64.

1.12 about

About Window: (Shortkey ?)

This will open a window containing some info on the current module:

Module name	: The name of the module.
Author	: The author of the module.
Active Player	: The Player-library which is used now.
Number of tunes	: Number of tunes in the module.
Song Lenght	: How long is this module, songpositions.
Used Patterns	: How many different patterns are used.
Supported/Used samples	: The number of used or supported samples e.g. a Protracker module will always use 31 samples, while a QComposer uses a various number of samples.
Actual Speed	: This is only useful for players supporting cia-speed commands (like Protracker).
Module Size	: The size of the unpacked module.

NOTE: A lot of the above-mentioned parameters is not supported by all the players. See module types for info on which players supporting what, or try it out for yourself.

The "more"-gadget will show you the version number, the arexx-port name and the creditlist for the AccessiblePlayer.

You can use the "cancel"-gadget to go back to the main window again.

1.13 tunes

Tunes:

A module is normally consisting of 1 tune, but players like SID and TFMX supports more tunes in one module. You can choose between the different tunes using the keys 1-9 and 0 for 10. Of course it’s also allowed to use the numeric keyboard. If there is more than 10 tunes, you can use "+" and "-" to skip to the next 10 tunes or to go back to the previous 10 tunes.

The only place where you can't do it is in the sample info window and in the extern player config windows.

1.14 volreset

Volume Reset:

Pressing this gadget will reset the volume to the default volume, saved in your config-file (see configuration).

1.15 volume

Volume Adjusting:

This slider is used to boost or lower the volume of the player. Remember that the actual volume is saved with your configuration file!
(See Configuration)

See also Volume reset

NOTE: Nearly all moduleplayers support this, but a few don't. See in the Module types section for more info.

1.16 listview

Main window module list:

This displays all the modules currently in the module list. If you choose a module it will be highlighted. That is, the playing module is always the one which is highlighted.

1.17 scroll

Modulelist Scroller:

Use this scrollbar to scroll through the list of modules.

1.18 sreset

Speed Reset:

Pressing this gadget will reset the speed to the default value, saved in your config-file. (See configuration).

NOTE: Changing of speed is only available when the player supports cia-timing. (See Module types)

1.19 speed

Adjusting Speed :

This slider is used to fasten or slowing the speed of the module. Remember that the actual speed is saved with your configuration file! (See Configuration)

NOTE: Changing of speed is only available when the player supports cia-timing. (See Module types)

See also Speed reset.

1.20 1

Sound Channels:

These 4 buttons are used to turn off one, some or all of the amiga sound channels. This is used when you want to accompany the module (See the Sampleinfo window).

Ofcourse they can be used for fun too!

NOTE: It's not all moduletypes which supports turning off channels e.g. the 7 channel TFMX-player. See the Module type section.

1.21 loop

Module Loop:

This gadget can toggle between positions:

- No looping, which will automatically skip to the next module, when a module is finished.
- Looping, which will make the actual module start over again when it's finished.

NOTE: This function is only supported by some moduletypes. See the Moduletype section for info.

1.22 shuffle

Modulelist Shuffler:

Pressing this gadget will shuffle all the modules in the list, making the actual module the first. If no modules are played the first module in the shuffled list will be loaded and played.

Of course this function only have a meaning when you have more than 2 modules in the list.

1.23 icn

Iconifying AccessiblePlayer: (Shortkey i)

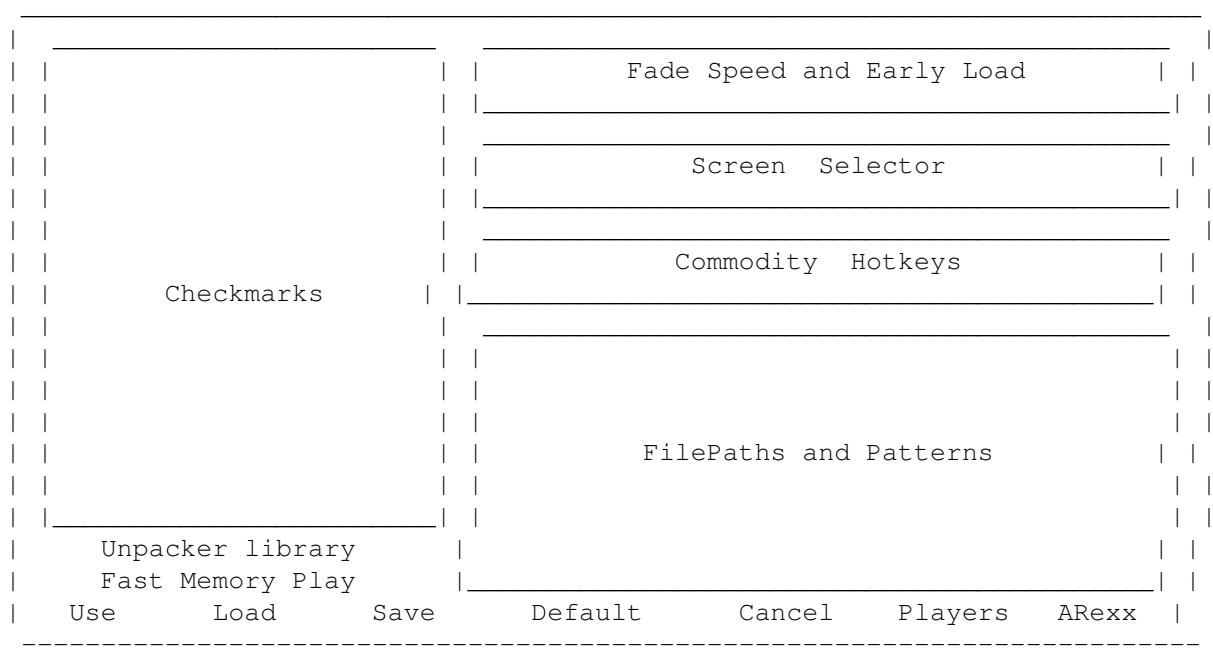
This will close all the open windows belonging to the APlayer, and pop up an APP-ICON. If you want to open the Player again you just double-click the app-icon. Or you can use the popup-hotkey specified in the configuration-window.

1.24 app-icon

AccessiblePlayer App-icon:

When the Player is iconified you can drop module-icons on the app-icon. This is done by opening the drawer in which you are keeping your modules, select one or more (using shift), then move the pointer over the APlayer app-icon and release the mousebutton. The modules will automatically be appended to your actual modulelist. And if it's empty, the player will play the first of the dropped modules.

1.25 cfg



NOTE: If you can't open the configuration window it might be because your workbench screen is to small. The minimum size is 640x254.

1.26 cfg1

Checkmarks:

This box contains 14 checkmarks, controlling some of the functions in the Accessible Player. All the functions are listed below:

```
Load Libraries        - Load libraries at start.
Expunge Libraries     - Kill libraries at end.
Load ARexx library    - Load ARexx at start.
Allocate Channels     - Lock your AudioChannels.
VBlank Interrupt      - VBlank or Cia.
Fade Module At End    - Automatic Fade.
Fade At Pause/Next    - What do you think?
Double Buffering      - More ram -> more modules.
Jump To Loaded Module - Which module to play.
Force Filter Off      - Keeps the Filter off.
Error Requesters      - Warn or not.
Lha Check             - Check for Lha.
Favourite Song System - Your favourite tunes.
Save Windowpositions  - Store your favourite positions.
```

1.27 loadlib

Load Libraries: (Shortkey b)

This gadget will cause APlayer to load all the player libraries AND the selected unpack library, defined in the player-configuration, the first time APlayer is started. This will of course use more memory, but it will also give faster access to all playerlibraries which means quicker moduleload.

It can also seems handy for people without harddisk, because loading of all player libraries at start avoids a lot of diskswapping during module-load.

Default is ON.

1.28 expungelib

Expunge Libraries: (Shortkey i)

With this checkmark you can decide wether the memory used by already loaded player libraries should be released or not, when you quit the program.

If you often quit and reload the APlayer, then this button will help you. As it will prevent the playerlibraries from being loaded everytime you restart the player and load a new kind of module.

Because most people want to have as much memory as possible, this function is by default set to ON.

Default is ON.

1.29 loadarexx

Load ARexx Library: (Shortkey o)

Default is OFF.

1.30 allocchan

Allocate Channels: (Shortkey a)

With this gadget ON you can prevent other music programs from interfering with the audiochannels. If you change the state of this gadget it will only work when you restart the APlayer.

Default is ON.

1.31 vblanki

VBlank Interrupt: (Shortkey v)

This gadget is used to help people who's still using SoundTracker and NoiseTracker modules, containing VBlank speed commands bigger than 1F. These can under normal conditions sound to slow, played by this player. But checking this gadget, will cause the player to interpret all speed commands in the module as VBlank speed commands, which will correct the speed errors.

Default is OFF.

1.32 fadeend

Fade Module At End: (Shortkey e)

This flag will cause the player to automatically fade the module at end. See also Fade Speed.

Default is OFF.

NOTE: This function isn't supported by all moduletypes. See module types.

1.33 fadepause

Fade At Pause/Next: (Shortkey p)

This flag will cause the player to automatically fade the module when the user hits the "next module"- or the "pause"-gadget in the main window. Note that releasing the "pause" button again will fade up the volume

again.

Default is OFF.

NOTE: This function is not supported by all moduletypes. See module types.

1.34 dbuf

Double Buffering: (Shortkey g)

This function will cause the APlayer to load the next module in the list, while the current module still plays. This will normally prevent the silence between two modules. If you don't check this one, you will save some ram, but you will have to wait in silence for the next module to be loaded and started.

See also Early Load for more info on this.

Default is OFF.

1.35 jumpload

Jump To Loaded Module: (Shortkey j)

If there's already some modules in the modulelist and you press the PLAY button, and select a number of files, they will automatically be added to the end of the list. With this function set to ON, APlayer will be forced to load and play the first selected module (in the filerequester).

Default is OFF.

1.36 forcefilter

Force Filter Off: (Shortkey f)

This will force APlayer to keep the audiofilter turned off. This is useful for old noisetracker modules which used one command to turn the filter on/off instead of the new protracker which uses two different to do so.

Default is OFF.

NOTE: This function is not supported by all moduletypes. See module types.

1.37 errreq

Error Requesters: (Shortkey r)

If you don't want the player to warn you when an error arrives, you should check this one. There are three different kinds of error messages:

1. DOS-errors (read/write error, file not found etc.).
2. ModuleLoad errors (Unknown moduletype,out of mem, decrunch error etc.)
3. Channel allocation error.

The Dos-errors will always delete the module from the list (no use for crap files in the list, or..). The second will, if turned off, not do any harm, usually just skip to the next module in the list.

But the last one will, if you have checked the Allocate channels gadget, and if the audiochannels is already occupied by another program, cause the player not to start at all. That's because the error requester stating that the channels are in use, will automatically be cancelled, due to the state of this gadget.

Default is ON.

1.38 lhacheck

Lha Check: (Shortkey h)

This will force APlayer to check all selected files if they are lha-archives. This will, depending on your extract pattern setting, put ALL the filenames from the archive into the modulelist, and then treat them as ordinary files. If you want to speedup the modules-to-list process, and you don't need the lha-support, then leave this gadget OFF. Note that only files in the root directory in the lha-archive will be read.

Default is ON.

1.39 fss

Favourite Song System: (Shortkey t)

You can use this to turn the FSS on or off. If you turn it off, the FSS file will be saved when you close the config window, and the file will be loaded if you turn it on.

See Favourite Song System for some info on the FSS.

Default is OFF.

1.40 windowpos

Save Windowpositions: (Shortkey w)

If this is ON, the positions and sizes of the all windows, except the extern players config windows, are stored in the config-file.

Note that if you try to open a window with coordinates which are too big for the actual screen, then the windowposition and size will be set to default.

Default is OFF.

1.41 cfg2

Unpack Library:

This cyclegadget is used to select which library you want to use for unpacking your modules. There are four possibilities:

1. None : If you do not have any packed modules, use this selection.

2. Powerpacker: This will use the powerpacker.library to depack your modules. This will only allow powerpacked modules to be played.

3. XPK : This will make use of the xpk-libraries for depacking your modules. This means that it can unpack all kinds of xpk-packed files including powerpacked files. Included in this package is SQSH-, SMPL- and Xpkmaster-.library.

4. Unpack : This is a library made by the coder of this program, which can recognize and decrunch approximately 150 different types of crunchers. This includes xpk, and powerpacker.

Default is None.

1.42 cfg3

Use : Pressing this will save the preferencefiles in ENV:APlayer/ and close the configuration window.

Load : Opens a filerequester from which you can select a preference file to be loaded (default is APlayer.prefs).

Save : This will save the preferencefiles in both ENV:APlayer/ and in your ENVARC:APlayer/ directory. It will also close the config window.

Default: This will reset all the values in the configuration window.

Cancel : This will close the config window and use your preferences from the ENV:APlayer/APlayer.prefs file.

Among all the settings saved in the prefs file is the state of the main window cycle, module loop, sound channels and the speed and volume settings.

1.43 cfg4

Fade Speed: Use this slider to choose how fast the fading of a module should be. 1 is the fastest and 4 is the slowest. Note that this will only affect the Fade Module At End.

Early Load: This slider is used in conjuction with the double buffering function. The value (1-9) indicates how many patterns (positions) before the module ends, the next module should be loaded. For harddisk users a value of 1 is enough, but if you use a diskdrive, you will have to set it to 3 or 4, dependent of the modulesizes.

1.44 cfg5

Specific or Default Public Screen:

Set to default, the player will open on the default public screen. This is normally the workbench screen. But with specific you can force APlayer to open on another public screen. You can then use the "?" gadget to select which screen you want it to use. Or you can type the name in the stringgadget, but remember that screen names are CASE-SENSITIVE.

1.45 cfg6

App Popup/Hide Hotkey:

This string contains the hotkey for the APlayer Program. Pressing the defined hotkey will close all open Aplayer windows, and popup an app-icon on the workbench. See app-icon.

Popup/Hide Hotkey:

This is nearly exactly the same as the hotkey above, but this will NOT popup an app-icon on workbench. See app-icon.

Hotkey To Skip Module:

Pressing the defined hotkey will be the same as pressing the "next module" gadget in the main window.

NOTE: Any valid commodity hotkey can be used for these hotkeys.

1.46 cfg7

Paths & Patterns:

Cut Prefixes:
As you maybe have noticed there isn't must space left for the modulenames in the main window. This is usually because most moduletypes is classified via a file prefix eg. "mod." This string is used to define which prefixes APlayer automatically should cut from the list. The format is like this: Prefix1|Prefix2|Prefix3 etc. (notice the "|" between the different prefixes!)
If you don't want anything to be left out, just leave this string empty.

Module Pattern:
Here you can define the filepattern in the filerequester when you load modules. Default is ~(SMPL.#?) which means that all files, except files starting with "smpl.", should be displayed.

Start Scan Path:
In this string you can specify a directory which will be scanned for modules when you start the APlayer. The modules will be shuffled and a random module will be choosed and played. This will ofcourse also cause lha archives to be unarchived. If you don't want this to happen, leave this string empty.

Module Path:
This tells Aplayer where you want it to look for your modules. You can use the diskgadget to the right to choose the modulepath from a file-requester.

APML/FSS Path:
This tells Aplayer where you keep your modulelist files and your Favourite Song System file. You can use the diskgadget to the right to choose the modulepath from a file-requester. Default is in S:

Temp Path:
This is the path which APlayer will use to unpack crunched files and store lha files and so on. You can use the diskgadget to the right to choose the modulepath from a file-requester. Default is T:

Lha File:
Here you should type the complete path+filename to your lha unpacker. You can use the diskgadget to the right to choose the filepath from a file-requester. Default is just "lha" which means that Aplayer just will look in the defaultpath.

Extract Pattern:
This is used for lha files. With this pattern you can tell Aplayer which files in the archive to extract. This is extremely helpful if you have some sort of description text file in all your archives, then you can leave it by using a pattern for instance saying "mod.#?".

1.47 cfg8

Fast Memory Play:

This function is meant for all people (aliens) with a small amount of chip memory free, or just for everyone who just wants to spare some of it. To make this short: this function can play your modules from fast memory, which will save some chip memory, but occupie some more processor time. See for instance Protracker config.
3 different settings are posssible:

Never: Never use fast memory play. (Default).

Always: Always use fast memory play.

When Needed: Only use it when there isn't more chip memory left.

NOTE: see technical description of Fast Memory Play for more info..

1.48 fasttech

Technical Info About The Fast Memory Player:

Well, I will now try to describe how the fast memory player works in the Protracker player. It's very simple. It will allocate 8 buffers,2 for each channel (double buffering), in chip memory with the size you have given through the configuration. When the player start to play a new note, it starts to copy the first part of the sampling into one of the buffers and start the audio interrupt for the channel in which the sample should be played. The audio interrupt will occur when the hardware is finished to copy the hardware register to the intern buffers in the audio chip (anybody said DMA wait?). In my audio interrupt I begin to copy the next part of the sample to the next buffer and set the hardware register to point to it. When the hardware has to loop the (one buffer) sampling, it will start to play the next buffer instead (I have set the hardware registers) and there will come a new audio interrupt in which I will copy the next part etc. Note that my copy routine takes care of loops etc.

If you will try to make you own fastmem player, these should be some good tips on how to do it. I had some big troubles with samples shorter than the buffer size, but it's should be fixed. If you observe some strange things with the fastmem player, then please send me a letter. My address can be found in Contact.

1.49 cfg9

Players Configuration Window:

APlayer supports a lot of different module types from the SID format to Protracker. Some of these players are build in the main program while the rest of them are stored as library files in a special directory, which you define during the installation process. When APlayer has loaded a

file, it has to check which type of module it is. With this window you can decide which player libraries that should be used, and in which order they should be used. The window is split into three boxes, the first one displaying all active player libraries, the second displaing some info on the selected library and the third containing the following gadgets:

Config: Opens a window in which special settings for one player can be changed.

Add: Opens a filerequester from which you can choose the libraries you want to add to the list.

Delete: This will delete the selected library from the list, but NOT from the disk.

Exchange: Select a library, press exchange and select a library to exchange the first library with.

Clear: This will clear the all libraries except for the internals.

Sort: Sorts the list alphabetically.

Toggle: Pressing this will set the selected library in brackets, which means that it would not be used. This can also be done by double-clicking a library in the list. This can avoid the use of the internal players, because they can't be deleted.

1.50 libcfg

Special Player libraries configuration:

This is currently only available for a few players, cause we haven't got any ideas for the rest of the players (if you got any - WRITE NOW!).

The window is build up as a standard amiga preference editor. Which means that all menus are the same, and that the preferences are stored in ENV: and ENVARC:

1.51 procfg

Protracker Configuration:

Here you can set the size of the "Chip Memory Buffer" which is the amount of chip memory used for every channel while playing from fastmemory. The default is 512 which suits most configuration, but if it gives you trouble you can try to change it.

But remember that a smaller buffer makes the processor work harder, because it has to copy the sampledata much more often than with a bigger buffer. But still, a big buffer can give some troubles, so try it out for yourself.

1.52 mfss

Favourite Song System: (Shortkey f)

The Favourite Song System (FSS) is for you who can't remember which modules you like to hear. Ok that was a bit of a joke! If you turn the FSS on in your configuration, APlayer will automatically store the names of all the modules your have heard during your last run of the APlayer. In addition to that it remembers how many times you have heard the same module. All the names are put into a list, from which APlayer takes the 10 most often played modules and put them into the "Favorite Song System Top 10" window. This window displays the top 10 placements and how many times they have been played. The list is saved to disk every time you open the FSS-window.

You can doubleclick an item to place it in the bottom of the module list.

RND One: This will randomly choose a module from the Top-10 and add it to the bottom of the module list.

RND All: Shuffles all the Top-10 modules and put them in the module list.

Reset FSS: Asks if you wants to clear the FSS list from the memory and delete the FSS file on your disk too.

NOTE: After some time the tree-structure of the FSS file can be a bit unstructured, which will cause the tree scan to slow down. To cure this problem you can use the "FSSOptimizer" program in the bonus drawer.

1.53 ml

Module list Editor: (Shortkey m)

This editor is used to create/change/load and save modulelists.

Add: Opens a filerequester from which you can choose the modules you want to add to the module list. The selected modules will either be added in the end of the list or, if you have selected a module in the modulelist, just before that.

Del: Deletes the selected module from the list.

Exg: Select a module, press exchange and pick a module to exchange with the first.

Clear: Clears the module list completely.

Sort: Sorts the module list alphabetically.

Load: Opens a filerequester which lets you open a new module list, deleting the current one.

Append: Exactly the same as above, but this will append the chosen list at the end of the list, or just before the selected module.

Save: Opens a filerequester which lets you save the current module list, with an ".APML"- extension.

NOTE: A nice feature is that if you doubleclick an item in the module list, in the module list editor, the module will be loaded. This can give you a better view of how the list looks.

1.54 modlist

Module list:

The module list is the name for the modules displayed in the main window module list.

1.55 prev

Previous module:

Pressing this button will tell APlayer to restart the current module, except if the current module is still playing the first pattern^1, then APlayer will skip to the previous module in the list.

^{1}NOTE: This function is only supported by some moduletypes.
See the Moduletype section for info.

1.56 rew

Rewind:

This will skip to the previous pattern in the current module. If you press rewind when the first pattern is playing, it will just restart.

NOTE: This function is only supported by some moduletypes.
See the Moduletype section for info.

1.57 play

Play:

This opens a filerequester which lets you add one or more modules to the current list of modules. This will only disable the pause mode if you have the Jump To Loaded Module gadget in the configuration checked.

1.58 ff

Fast Forward:

Use this to skip to the next pattern. If you reach the end it will load and play the next module.

NOTE: This function is only supported by some moduletypes.
See the Moduletype section for info.

1.59 next

Next Module:

Pressing this will tell APlayer to skip to the next module in the list. If you are in pause mode APlayer will unpause and skip to the next module.

1.60 eject

Eject:

Pressing this once will stop the current module and free the memory. Pressing it again clears the module list.

1.61 pause

Pause:

This will simply pause the module playing right now. And play if you press it again.

1.62 extra

Space can be used to toggle the state of the audio filter. However, if you have checked the Force Filter Off gadget in the configuration window, space will not affect anything.

TAB cycles through the APlayer windows.

Escape use this to close the active window. All APlayer windows can be closed, except for the main window.

1.63 modtypes

```
                                                                |F |
                                                                |o |
                                                                |r |
                                                                |c |
                                                    |M |S |P |  |e |
                                                    |a |h |l |  |  |
                                     |M |           |x |o |a |  |F |  |O |
                                     |o |   *       |  |w |y |A |i |  |w |
+-                                |C |d |  |S |  |P |P |  |  |c |l |  |n |
  +-               |F |           |h |u |  |u |  |o |a |S |S |c |t |F |  |
    +-          |R |o |   V |     |a |l |A |b |L |s |t |a |a |o |e |a |C |
      +-  Func. |e |r |P |o |     |n |e |u |s |e |i |t |m |m |m |r |s |o |
        +-      |w |w |a |l |F |  |n |n |t |o |n |t |e |p |p |p |  |t |n |
          +-    |i |a |u |u |a |C |e |a |h |n |g |i |r |l |l |a |O |m |f |
 Players:   +-  |n |r |s |m |d |I |l |m |o |g |t |o |n |e |e |n |f |e |i |
              +-|d |d |e |e |e |A |s |e |r |s |h |n |s |s |s |y |f |m |g |
----------------+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
DigitalMugician |X |X |X |X |X |  |  |X |  |X |X |X |X |  |  |  |X |  |  |
DSS             |X |X |X |X |X |  |X |  |  |  |X |X |X |X |X |  |X |  |  |
Fred            |  |  |X |  |  |  |  |  |X |X |  |  |  |  |  |  |X |  |  |
Future13        |X |X |X |X |  |  |X |  |  |  |X |X |  |  |  |  |X |  |  |
Future14        |X |X |X |X |  |  |X |  |  |  |X |X |  |X |X |  |X |  |  |
HolyNoise       |X |X |X |X |X |  |  |  |  |  |X |X |X |  |  |  |X |  |  |
IFF-8SVX        |  |  |  |X |  |  |  |X |  |  |X |X |  |  |  |  |X |  |X |
JamCracker      |X |X |X |X |X |  |X |  |  |  |X |X |X |X |X |  |X |  |  |
MED             |X |X |X |X |  |  |  |  |  |X |X |X |X |X |X |  |  |  |  |
MON             |  |  |X |  |  |  |  |  |  |X |  |  |  |  |  |  |X |  |  |
MusicAss        |  |  |X |X |  |  |  |  |  |X |  |  |  |  |  |  |X |  |  |
NoisePacker20   |X |X |X |X |X |  |  |  |  |  |X |X |X |  |  |  |X |  |  |
NoisePacker30   |X |X |X |X |X |  |  |  |  |  |X |X |X |  |  |  |X |  |  |
OctaMed         |X |X |X |  |  |  |  |  |  |X |X |X |X |X |X |  |  |  |  |
Oktalyzer       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
Protracker      |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |X |
QuadraComposer  |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |  |  |
RIFF-WAVE       |  |  |  |X |  |  |  |  |  |  |X |X |  |  |  |  |X |  |  |
RonKlaren       |  |  |X |  |  |X |  |  |  |X |  |  |  |  |  |  |X |  |  |
SID             |  |X |X |X |  |  |  |X |X |X |  |  |  |  |  |  |  |  |  |
SoundMon20      |X |X |X |X |  |  |  |X |  |  |X |X |X |  |  |  |X |  |  |
StarTrekker4    |X |X |X |X |X |  |X |X |X |  |X |X |X |X |X |X |X |  |  |
StarTrekker4AM  |X |X |X |X |X |  |X |X |X |  |X |X |X |X |X |X |X |  |  |
Synthesis       |X |X |X |X |  |  |  |X |  |X |X |X |  |  |  |  |X |  |  |
TFMX_7V         |X |X |  |X |  |  |  |  |X |X |X |X |  |  |  |  |  |  |  |
TFMX_Pro        |X |X |X |X |  |  |  |  |X |X |X |X |  |  |  |  |  |  |  |
----------------+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

* For further info see Tunes.

1.64 tooltypes

Tooltypes and CLI arguments:

CX_POPUP (POP): Defines if you want the APlayer to open its window at

start, or start in hide mode. Keywords are YES (default) or NO.

CX_POPKEY (KEY): The hotkey for APlayer. All valid commodity hotkeys will work here. Default is "ctrl alt a".

CX_PRIORITY (PRI): The APlayer task priority. Default is 0.

MODULE: This argument is CLI ONLY! Just type the name (with path) of the module(s) and it will be placed in the modulelist.

MODULELIST (ML): The name of a modulelist file you want APlayer to use. This will overrule modules specified in the CLI argument string. For instance S:example.APML.

CONFIGFILE (CFG): Specifies the config file you want to use. Default is ENV:APlayer/APlayer.prefs

PUBSCREEN (PS): The public screen on which you want the APlayer to open on. Remember that the name is case SenSitiVe. Default is the default public screen e.g Workbench.

LOOP: Specifies the state of the loop gadget in the main window (ON/OFF) or toggles the state (set in your configuration) (TOGGLE).

MODULELISTOPEN (MO): If you want the module list window to open at start, use this argument.

FSSOPEN (FO): Use this argument if you want the FSS window to be opened at start.

SHUFFLE (SH): If you use this argument all the modules (or the modules in the specified modulelist) will be shuffled. Just like if you press the shuffle gadget in the main window.

1.65 programmers

How to code your own players!?

1.66 thanks

Thanks to:

Nico François for the fabulous Reqtools library and the powerpacker library

Urban Dominik Mueller & Bryan Ford for the even more fabulous Xpk master library.

John Hendrikx for the SQSH library

Jorma Oksanen for the SMPL library

Bo Lincoln for the real Quadracomposer player (including the cia and all commands)

Deftronic for Trash'm-One (We hate the enforcer hits!!)

Michael Sinz for Enforcer

Commodore for Mungwall and Segtracker

All the authors of the different players

All our betatesters

Peter Hjelt for the Smartplay which has been a big inspiration source for the Accessible Player design.

Amiga for being the best computer EVER (I.... outside!)

1.67 history

History:

1.00 First public release

1.68 contact

Send bugreports, new players including modules and suggestions to

Thomas Neumann (Tax)
Kongensgade 78
3550 Slangerup
Denmark

Send technotapes and modules to

Jakob Langgaard (Jail)
Krebsen 101
3650 Ølstykke
Denmark

Or write to Asger Høgsted at his E-Mail address
bims@diku.dk