AccessiblePlayer

Jail

COLLABORATORS			
	TITLE : AccessiblePlayer		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Jail	January 19, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 AccessiblePlayer 1

1.1 Root 1
1.2 features 2
1.3 future 3
1.4 disclaimer 4
1.5 shareware 4
1.6 requirements 6
1.7 introduction 6
1.8 installation 6
1.9 listview 7
1.10 usage 7
1.11 maincyc 8
1.12 status 9
1.13 sampleinfo 9
1.14 keypad 10
1.15 about 11
1.16 tunes 11
1.17 volreset 12
1.18 volume 12
1.19 mainlistview 12
1.20 scroll 13
1.21 sreset 13
1.22 speed 13
1.23 1 14
1.24 loop 14
1.25 shuffle 14
1.26 icn 15
1.27 app-icon 15
1.28 cfg 15
1.29 cfg1 16

1.30 loadlib 17
1.31 expungelib 17
1.32 loadarexx 18
1.33 allocchan 18
1.34 vblanki 18
1.35 fadeend 18
1.36 fadepause 19
1.37 dbuf 19
1.38 jumpload 19
1.39 forcefilter 20
1.40 errreq 20
1.41 lhacheck 20
1.42 fss 21
1.43 windowpos 21
1.44 cfg2 21
1.45 cfg3 22
1.46 cfg4 22
1.47 cfg5 23
1.48 cfg6 23
1.49 cfg7 23
1.50 cfg8 24
1.51 cfg9 25
1.52 cfg10 26
1.53 plcyc 27
1.54 libcfg 28
1.55 prefsmenu 28
1.56 mfss 28
1.57 ml 29
1.58 modlist 30
1.59 prev 30
1.60 rew 30
1.61 play 31
1.62 ff 31
1.63 next 31
1.64 eject 32
1.65 pause 32
1.66 keyboard 32
1.67 modtypes 34
1.68 iffcfg 35

1.69 aiffcfg . . . 36
1.70 riffcfg . . . 36
1.71 sidcfg . . . 36
1.72 smuscfg . . . 37
1.73 players . . . 37
1.74 noteplayers . . . 43
1.75 fastcfg . . . 45
1.76 32config . . . 45
1.77 tooltypes . . . 46
1.78 thanks . . . 48
1.79 history . . . 48
1.80 h100 . . . 49
1.81 h101 . . . 49
1.82 h102 . . . 49
1.83 h103 . . . 50
1.84 h104 . . . 50
1.85 h110 . . . 50
1.86 h120 . . . 51
1.87 h121 . . . 52
1.88 h130 . . . 52
1.89 contact . . . 53
1.90 arexxmain . . . 53
1.91 archannel . . . 54
1.92 areject . . . 55
1.93 arloop . . . 56
1.94 armaincycle . . . 57
1.95 armodulelist . . . 58
1.96 arpause . . . 61
1.97 arplay . . . 62
1.98 arrequester . . . 63
1.99 arsample . . . 65
1.100arspeed . . . 66
1.101artechnical . . . 67
1.102arvarious . . . 73
1.103arvolume . . . 73
1.104arwind . . . 74
1.105arwindow . . . 75
1.106index . . . 75

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


            Registration
               How to register


            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


            Keyboard control
```

How to control APlayer with the keyboard

Module Types
Module types supported by APlayer

ARexx
How to use ARexx with APlayer

Programmers Coders, read this

Thanks to
I want to thank my producer, and mom & dad...

History
The history of APlayer

Future
The future for APlayer ..

Contact
How to reach the author

1.2 features

Special APlayer Features:

- Supports 50 different module formats.

- Supports S3M, Fasttracker I & II, Multitracker and more.

- Uses noteplayers. Stereo, 14 bit, surround, fastmem player ...

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

- Favourite Song System© No one else does this!!

- Play samples by the keyboard while listening to the module.
  This now includes a multikeyboard function (press "DEL").
  The number of channels used depends of the module, so if you hear a 16
  channels S3M module you can play in multi mode with 16 CHANNELS!!!!

No one else does this!!

- Save samples can be used to rip your favourite samples, even if you don’t own the actual music program. This also works with S3M, XM a.s.o. No one else does this!!

- 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. We were the first!!

- ARexx interface - And much, much more...

1.3 future

The Future of APlayer:

In this text we will try to give you an idea of what will happen to APlayer in the coming versions.

- Genies (any ideas? then write us NOW!)
- Peakmeters (Martin Wulffeld, DU er velkommen til at lave nogen!)
- Automatic conversion of protracker clones
- Automatic ripping of modules from all kinds of files (e.g. from a demo executable file).
- Even more players
- Your ideas ;-)

- And more, more, more ....

1.4 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!

If you or your company want to use APlayer on a coverdisk or something like that, we would like to know about it, and get a copy of the product.

And since RBF Software wants us to do this, we must write that the OctaMED and MED players are copyrighted by them.

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

NOTE: This is probably the last PD version (=no cost for you). The next version will most likely be shareware, so you have to pay for it.

1.5 shareware

Shareware Payment:

Everything has a price .. even APlayer :) :(

Until now APlayer has been a free-to-use program for all of you. But due to the fact that APlayer now requires a lot of work from the programmer, we have decided to make APlayer shareware. If you do not register your copy of APlayer, there will be a few things to annoy you :) The program is still useable, there aren't any FU**ING requesters everytime you do anything. Only a few useful things have been disabled:

- No ARexx
- No Rewind/Fast forward
- A limited number of modules in the list at a time.

If you register you will also have get your name in the about window :)

IMPORTANT:

If you register, you will only register the MAIN program. All the players and noteplayers are still public domain. We can't make money on other peoples work :)

How To Register:

```
No matter how you register you will have to  fill  out  the  registration
form and send it to us together with your registration.

1. Cash Registration

   You can send the cash direct to the author in two different currencies:

   100 Dkr (Danish Kroner)

   25 US Dollars (Due to the exchange price)

   Send them to:

   Thomas Neumann
   Kongensgade 78
   3550 Slangerup
   Denmark

2. Eurocheck

   If you live in europe you can just use an eurocheck or another type of
   check which will not cost us anything to cash. The amount is:

   10 UK Pound

   25 DM

   or an amount equal to this ..

   Send them to: See above

3. NET registration (Idea taken from the GoldED package)

   This is the easiest way to register. You can  send  an  email  to  the
   email  address  below.  Then  you  can  insert the money direct on the
   author's account (this can be done in any bank).
   You will then recieve an E-mail with an UUencoded keyfile.

   Send the letter to:

   jail@diku.dk

   and transfer the money (100 Dkr) to this account within two weeks (you
   will not get the keyfile before we recieve the money):

   Thomas Neumann

   Account number: 5822 0003091623

   Unibank
   Frederiksborgvej 9
   3650 Ølstykke
   Denmark
```

1.6 requirements

Hard'n'soft-ware requirements:

You need at least kickstart v2.04 to use this program.

Three general libraries are necessary to run the APlayer with all functions. APlayer can not start without the Reqtools.library(v38+) (included) and any version of the Diskfont.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.

To make use of the ARexx part of APlayer you need the Rexxtools.library which is included in the package.

The APlayer has been successfully tested on the following machines:

Plain A1200, A1200+4MB fastram, A1200+68030+4MB Fastram, A4000(680EC30)+ 16MB Fastram and an A4000(68040)+8MB Fastram, including test with Enforcer, Mungwall and IOtorture.

1.7 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.

P.S. Some people think that we are the guys behind Smartplay. That is NOT true. We have never been in contact with the maker of Smartplay and we probably never will.

1.8 installation

Installation:

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

```
Warning: APlayer is pretty complex to  install, so if you want to install
         it by hand, don't blame us if something goes wrong.
         In other words: "Don't try this at home"
```

1.9 listview

```
Listview Control:

Arrow up or down  will move  the  highlightbar  one position up  or down.
If no element in the list is highlighted the highlightbar will appear  at
any cursor press.

Using Shift with the cursor keys, will move the list either one "page" up
or one page down.

Using  control or alt with the cursor keys  will move the highlightbar to
either the top or the bottom.

Return will toggle the current (Note)player’s state (enabled/disabled).
```

1.10 usage

```
                The Accessible Interface:
 ----------------------------------------------------------------------
 | · |  AccessiblePlayer 1.30
             (1/1)
                       |
 |----------------------------------------------------------------------
 |
              Cycle gadget
                              Status bar
              S
              ?
             |
 |----------------------------------------------------------------------
 |
             _ -
              Volume Slider
             |                                         |    |
 |--------------------|                                        |    |
 |
              »>
               Speed Slider
             |
              Main Listview
                            |
              |
             |
 |--------------------|                                        |    |
 |
              1 2 3 4
              []
             ^v^v
             |                                         |    |
 |----------------------------------------------------------------------
 |
              ICN
              CFG
```

```
                 FSS
                 ML
                  |<
                  <<
                   >
                  >>
                  >|
                   ^
                  ||
                |
   -----------------------------------------------------------------------

  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
                .

  The Zoom Gadget can be used to zip the program window.  The  main  window
  will  be  zipped  to the titlebar only. Notice that the placement will be
  saved in the configuration file.
```

1.11 maincyc

```
                 Main window cycle:

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

  There are five 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  support the  author  name.

   Played :
    This  shows,  if supported (see
                Module types
                ), the actual position and
    the  total  songlength.  It  also  calculates  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.

RexxMsg:
This displays info text made by an
Arexx
script.

1.12 status

Status Bar:

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

1.13 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. The samples are always saved in signed IFF-8SVX format.

From left to right it will display the number of the sample, name, size in bytes, type of the sample and whether it’s placed in chip- or fast-memory.

The sample can be of these different types:

XX	- it means that it’s a sample with the sample bit XX.
AM	- Synthetic voices (Amplitude Modulation)
FM	- Precalculated synthetic voices (Frequency Modulation)
HB	- Sample used as a synthetic voice (HyBrid)
AL	- AdLib sound used in S3M modules
??	- Unknown type
--	- No sample

If the type is typed in bold it means that the sample is unsigned.

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 whether 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.

Multichannel Mode:
   Per default  APlayer will only play  the sample in one channel. But if
   you   press  "DEL"   (the  Delete  key)   in  the  sampleinfo  window,
   multichannel mode will be enabled.
   If you are in pause mode, all the channels  which are ON will be used.
   If  you are  using the  accompany  function, the  OFF  turned channels
   will be used to play the samples.
   Please note that the number of channels used is  dependant of the used
   Player.

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. Pressing "return" will do the same.

Instead of using the keypad or the mouse, you can use the
              keyboard
              .

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 support  the sample info window, and therefore not
      the  sample-save  and  accompany  functions. (see
              Module types
               for
      more info)
```

1.14 keypad

```
                Numeric Keypad:

This  works  exactly  like in
              QuadraComposer
              . Sample 1-16 is chosen 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.
```

1.15 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.
  Active NotePlayer     : The actual NotePlayer.

  Number of tunes       : Number of
              tunes
               in the module.
  Song Length           : 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.
  Used Channels         : The number of channels in the actual module.
  Used Mixing rate      : This  will, if a NotePlayer  is mixing, display
                          the actual mixing rate.
  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 "next"-gadget will show you the version number, the  arexx-port  name
and the creditlist for the AccessiblePlayer.

You can use the "cancel"-gadget or <return> to go back to the main window
again.
```

1.16 tunes

```
Tunes:
```

A module normally consists of 1 tune only, 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, you can also use the numeric keyboard. If there are more than 10 tunes, you may 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 external player config windows.
The "(x/x)" in the title bar shows "current tune/total number of tunes".

1.17 volreset

Volume Reset:

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

1.18 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!
You can also use keyboard.

See also Volume reset and Configuration
NOTE: Nearly all moduleplayers support this, but a few don’t. See in the Module types section for more info.

1.19 mainlistview

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.

1.20 scroll

```
                Modulelist Scroller:

Use this scrollbar to scroll through the
              list of modules
              .
```

1.21 sreset

```
                Speed Reset:

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

You can also use
              keyboard
              .

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

1.22 speed

```
                Adjusting Speed :

This slider is used for changing the speed of the module.
Remember that the default speed is saved with your configuration file!
(See
              Configuration
              )

You can also use
              keyboard
              .

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

See also
              Speed reset
              .
```

1.23 1

Sound Channels:

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

Please notice that these buttons indicate the hardware audio channels, which means that if a noteplayer is playing, pressing the channel 1 button can turn of upto 8 channels.

Ofcourse they can be used for fun too!

You can also use
keyboard
.

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

1.24 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.

You can also use
keyboard
.

NOTE: This function is only supported by some moduletypes. See the

Moduletype section
for info.

1.25 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.

You can also use
keyboard
.

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

1.26 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
.

1.27 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.

1.28 cfg

```
|  ______________________________    ______________________________________________  |
| |                              | |
                  Fade Speed and Early Load
                         | |
| |                              | |______________________________________________| |
| |                              |   ______________________________________________  |
| |                              | |
                     Screen  Selector
                           | |
| |                              | |______________________________________________| |
| |                              |   ______________________________________________  |
| |                              | |
                    Commodity  Hotkeys
                           | |
| |
                 Checkmarks
                     | |___________________________________________| |
| |                        |  ___________________________________________  |
| |                        | |                                           | |
| |                        | |                                           | |
| |                        | |                                           | |
| |                        | |
                  FilePaths and Patterns
                        | |
| |                        | |                                           | |
| |________________________| |                                           | |
|
                    Unpacker library
                 |                                            | |
|

                 |____________________________________________| |
|
                  Use      Load      Save      Default      Cancel

                 Players
                 ARexx
                 |
 ---------------------------------------------------------------------------

 NOTE: If you can't open the configuration window it might be because your
       workbench screen is too small.  The minimum size is 640x254.  So if
       you are an NTSC user,  you should use  WB Prefs to make your screen
       at least that big, or interlace it.
```

1.29 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.

                 Make ARexx Port
                  - Creates the APlayer ARexx port.

                 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.

1.30 loadlib

Load Libraries: (Shortkey b)

This gadget will cause APlayer to load the selected unpack library AND all the player libraries, 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 come in handy for people without harddisk, because loading of all player libraries at start avoids a lot of diskswapping during module-load.

Default is ON.

1.31 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.

1.32 loadarexx

Make Arexx Port: (Shortkey o)

This will create an Arexx port named "APLAYER". See the Arexx doc for more info on the Arexx port.

Default is OFF.

1.33 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 the channels will be (de)allocated when the next module is loaded.

That is, the channels will always be allocated as long as there is a module in memory.

Default is ON.

1.34 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.

1.35 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
.

1.36 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
.

1.37 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.

1.38 jumpload

Jump To Loaded Module: (Shortkey j)

If there are modules already 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.

1.39 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
.

1.40 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 four 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. Arexx error (couldn’t find Arexx)
4. 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 play the requested module.

Default is ON.

1.41 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.

1.42 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.

1.43 windowpos

Save Windowpositions: (Shortkey w)

If this is ON, the positions and sizes of all the windows, except the extern players config windows, are stored in the APlayer.win configuration file.
The APlayer.win file will always be loaded when you start the APlayer, but it will only be saved if you have this flag checked.
This means that if you want to return the window positions and sizes to the default values, you have to manually delete the file from ENV: and ENVARC:

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 fitted to the screen if possible, else it will not be opened.

Default is OFF.

1.44 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.
```

1.45 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.
```

1.46 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.
```

1.47 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.

You can use the
              keyboard
               to move around in the list.
```

1.48 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.
```

1.49 cfg7

```
                 Paths & Patterns:

  Cut Prefixes:
   As  you  have  maybe  noticed, there  isn't  much  space  left  for  the
   modulenames in the main window. This is because most moduletypes usually
   are  classified  by a file prefix,  e.g. "mod."  This string  is used to
   define  which prefixes  APlayer should cut  automatically from the list.
   The format is like this: Prefix1|Prefix2|Prefix3  etc. (notice  the  "|"
   (pipe) 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 may 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 chosen and played.  This will of course 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 S:

  Temp Path:
   This is the path which APlayer will use to unpack crunched  files, store
   lha files  and so on. You can use the  diskgadget to the right to choose
   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 will just
   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
   avoid unpacking it by using a pattern, for instance saying "mod.#?".
```

1.50 cfg8

```
No Function right now ..
```

1.51 cfg9

```
                ARexx Config:

In this window you can configure the Arexx part of APlayer.
The window contains 4 things, which is listed below:

1. F1-F10: In these strings you can define  which Arexx macros to execute
           when the actual key is pressed.
           Remember that the qualifier  which is used in conjunction with
           the  key  is  set  using  the cyclegadget (see below). If your
           macros  aren't  in REXX: then you have to write the full path.
           This  can  be  done  easily  by pressing the diskgadget to the
           right of each stringgadget.

Note: It is not allowed to have spaces in filenames or paths. This is due
      to an "error" in Arexx, not in APlayer.

2. Cycle: With this you select 3 different states:

        Shift or Alt: This position contains the names of the macros  to
                      be  executed   when  you  have  shift/alt  pressed
                      together with F1-F10.

        Other: Changes the names of the strings to the following:
               ST: (STart) The macro to be  executed when APlayer starts.
                   Note this is the last thing which is done when APlayer
                   starts.
               ED: (EnD) This macro will be executed when you try to quit
                   APlayer. Note that the only way APlayer can exit then,
                   is by sending a "quit" commando to the Arexx port.
               PB: (PlayButton) This macro is executed when you press the
                   playbutton. Note that it'll  prevent the filerequester
                   from  appearing, which enables  you  to make  your own
                   moduleselector via Arexx.
               PL: (PLay) This  macro is  executed when a module has been
                   successfully recognized and loaded.

3. C: (Clear) This will clear all macro  settings. Note that  it will ask
              you to confirm your choice first.

4. Save  : This  will save the macro  setting file in ENVARC:APlayer/ and
           ENV:APlayer/ with the name APlayer.arexx.
   Use   : This will save the macro setting file in ENV:APlayer/ with the
           name APlayer.arexx
   Cancel: Closes the Arexx window without saving any changes.


Click
             here
              to see how to use the menus in this window.
```

Note that this window is independent of the main configuration window. That is, you can press "use" in the Arexx window and cancel in the main window, and the changes in your macrosettings will be kept.

1.52 cfg10

Players Configuration Window:

APlayer supports a lot of different module types from the SID format to Protracker. These modules are played by using player libraries and sometimes also a noteplayer.

To choose which settings to change you can use the
cycle gadget
in the
top left of the window.

Pressing the ? gadget in the top right of the window will open a list, from which you can choose the NotePlayer you prefer to be used with the selected player. If you don’t want any first priority, just delete the name manually (you may use Amiga-X).
If, at a later time, the selected NotePlayer can’t be loaded, APlayer will give you a requester where you can choose to stop or make APlayer choose a Noteplayer for you.

```
Arrow up/down: These will move the  marked item to  either  the  position
               above or below.
               Pressing shift in  conjunction  with the gadget, will move
               the selected item to the top or the bottom of the list.
```

Config
: Opens a window in which special settings for the selected player may be changed.

```
Show:     No function yet...

Add:      Opens  a  filerequester from which you can choose the libraries
          (players) 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.


Save:     Saves  the  APlayer.libs  file  in  ENVARC:APlayer/  and  in
          ENV:APlayer/ directories. After the saving it closes the window
```

and uses the settings. If the actual player/noteplayer is deleted from the list and you press save, the playing module will be ejected.

Use: Saves the APlayer.libs file in the ENV:APlayer/ directory. After the saving it closes the window and uses the settings. If the current player/noteplayer is deleted from the list and you press save, the playing module will be ejected.

Cancel: Closes the window and restores the settings.

To toggle an entry in the list, you can just double-click the entry. This will prevent the use of the specific entry.

Click
here
to see how to use the menus in this window.

To move around the list you can use the
keyboard

1.53 plcyc

List Selector:

Players
A player recognizes and plays the module. Some feeds the hardware directly, some passes the data to a noteplayer.

When a player is selected and if it uses a noteplayer, you can press the "?" button in the top right of the window. This will open a window with a list containing all the NotePlayers which can be used by the player. Hereby you can select the NotePlayer you prefer should be used with the player. If by some reason the NotePlayer isn’t available when the player tries to use it, it will scan through the NotePlayers until it finds a useable one.

NotePlayers
The noteplayer recieves the sound data from the player and makes your computer play the sound. Some of the noteplayers has the ability to mix several channels into the four audio channels of the amiga.

Some (Note)Players are built into the main program while the rest of them are stored as files in "LIBS:APlayer" or "LIBS:APlayer/NotePlayers". When APlayer has loaded a file, it has to check which type of module it is. With this window you can decide which (Note)Players that should be used, and in which order they should be used (prioritized list).

When a (Note)Player is chosen, the version number is displayed in a little box in the top left of the right part of the window.

1.54 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 built up as a standard amiga preference editor. Which means that the preferences are stored in ENV: and ENVARC:

Click
here
to see how to use the menus in this window.

1.55 prefsmenu

Preference Menu:

Project:

```
 Open    : Opens a file  requester from which you  can load a new  config
           file.

 Save as: Opens  a file requester  from which you  can save  your actual
           settings.

 Quit    : This is the same as pressing cancel.
```

Edit:

```
 Reset To Default: This will reset the settings to the default values.

 Last Saved      : Sets your  configuration  settings to the settings of
                   the last saved configuration file.

 Restore         : The same as "Cancel", except that  it won't close the
                   window.
```

1.56 mfss

Favourite Song System: (Shortkey f)

The Favourite Song System (FSS) is for you who can't remember which modules you like to hear. Okay, 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 at 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.

Delete:  Deletes the marked module from the fss list. The changes will be
         saved to your fss file the  next time you either quit the player
         or open the fss window.

Reset:   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.
```

1.57 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.


Arrow up/down: These will move the marked module to  either  the position
               above or below.
               Pressing shift in  conjunction  with the gadget, will move
               the selected module to the top or the bottom of the list.

The number of modules in the list can always be seen in the right side of
the window titlebar.

To move around in the list you can use the
              keyboard
              .

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.
```

1.58 modlist

```
                Module list:

The module list is the name for the modules displayed in the

              main window module list
              .
```

1.59 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.

You can also use
              keyboard
              .

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

1.60 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.

You can also use
keyboard
.

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

1.61 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.

You can also use
keyboard
.

1.62 ff

Fast Forward:

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

You can also use
keyboard
.

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

1.63 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.

You can also use
              keyboard
              .
```

1.64 eject

```
                Eject:

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

You can also use
              keyboard
              .
```

1.65 pause

```
                Pause:

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

You can also use
              keyboard
              .
```

1.66 keyboard

```
                These keys can be used in all windows when they are activated:
---------------------------------------------------------------

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. If you press escape in the
                      main window,  one of the other windows will
```

```
                     close for each keypress.

These keys can only be used when the main window is activated:
--------------------------------------------------------------

`  Back apostrophe : Jump to random module and play it

~  Tilt            : Shuffle

Del                : Eject

Shift Return       : Reset volume

Alt Return         : Reset Speed

Arrow up           : Like pressing the play button

Arrow down         : Pause

Arrow left         : Rewind

Arrow right        : Forward

Shift Arrow left   : Loads next module

Shift Arrow right  : Loads previous module

Alt Arrow up       : Increase speed

Alt Arrow down     : Decrease speed

Alt Arrow left     : Increase volume

Alt Arrow right    : Decrease volume

[ or (             : Toggle channel 1    \
                                          |
] or )             : Toggle channel 2     |
                                          » Top row on numpad
/                  : Toggle channel 3     |
                                          |
*                  : Toggle channel 4    /

Backspace          : Module loop on/off

<                  : Cycle the maincycle gadget backwards

>                  : Cycle the maincycle gadget forwards

For  control  keys  in  the special  windows,  see the  sections  on  the
different windows.
```

1.67 modtypes

```
                                                                                |F ↩
                                                                                  ↩
                                                                                | ↩
                                                                |o |
                                                                |r |
                                                                |c |
                                                    |M |S |P |  |e |
                                                    |a |h |l |  |  |
                                     |M |           |x |o |a |  |F |  |O |N |
                                     |o |    *      |  |w |y |A |i |  |w |o |
+-                                |C |d |  |S |  |P |P |  |  |c |l |  |n |t |
  +-               |F |           |h |u |  |u |  |o |a |S |S |c |t |F |  |e |
    +-          |R |o |  |V |     |a |l |A |b |L |s |t |a |a |o |e |a |C |P |
      +-  Func. |e |r |P |o |  |T |n |e |u |s |e |i |t |m |m |m |r |s |o |l |
        +-      |w |w |a |l |F |e |n |n |t |o |n |t |e |p |p |p |  |t |n |a |
          +-    |i |a |u |u |a |m |e |a |h |n |g |i |r |l |l |a |O |m |f |y |
 Players:   +-  |n |r |s |m |d |p |l |m |o |g |t |o |n |e |e |n |f |e |i |e |
              +-|d |d |e |e |e |o |s |e |r |s |h |n |s |s |s |y |f |m |g |r |
----------------+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
ActionAmics     |X |X |X |X |  |X |X |  |X |X |X |X |  |X |X |X |X |  |  |  |
DeliCustom      |  |  |X |  |  |X |  |  |  |X |  |  |  |  |  |  |X |  |  |  |
DigitalMugician |X |X |X |X |X |X |  |X |  |X |X |X |X |  |  |  |X |  |  |  |
DSS             |X |X |X |X |X |X |X |  |X |  |X |X |X |X |X |  |X |  |  |  |
FastTracker     |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |  |X |
FastTracker20   |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |  |X |
Fred            |  |  |X |X |  |X |X |  |  |X |  |  |  |  |  |  |X |  |  |  |
FTM             |  |  |X |X |  |  |  |X |X |  |X |X |  |  |  |  |  |X |  |  |
Future13        |X |X |X |X |  |X |X |  |  |  |X |X |  |  |  |  |X |  |  |  |
Future14        |X |X |X |X |  |X |X |  |X |  |X |X |  |X |X |  |X |  |  |  |
Hippel          |  |  |X |  |  |X |  |  |X |X |  |  |  |  |  |  |X |  |  |  |
Hippel-COSO     |  |  |X |X |  |X |  |  |X |X |X |X |  |  |  |  |X |  |  |  |
IFF-8SVX        |  |  |  |X |  |X |  |X |X |  |X |X |  |  |  |  |X |  |
                X
                |  |
IFF-AIFF        |  |  |  |X |  |X |  |X |X |  |X |X |  |  |  |  |X |  |
                X
                |  |
IFF-SMUS        |  |  |X |X |  |X |X |X |X |  |  |  |  |  |  |  |X |  |
                X
                |  |
InStereo!       |X |X |X |X |  |X |X |X |  |X |X |X |  |  |  |  |X |  |  |  |
JamCracker      |X |X |X |X |X |X |X |  |X |  |X |X |X |X |X |  |X |  |  |  |
MajorTom        |X |X |X |X |X |X |  |  |  |  |X |X |X |  |  |  |X |  |  |  |
MED             |X |X |X |X |  |  |  |  |X |X |X |X |X |X |X |  |  |  |  |  |
MON             |  |  |X |  |  |X |  |  |  |X |  |  |  |  |  |  |X |  |  |  |
MultiTracker    |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |  |X |
MusicAss        |  |  |X |X |  |X |X |  |  |X |  |  |  |  |  |  |X |  |  |  |
NoisePacker20   |X |X |X |X |X |X |  |  |  |  |X |X |X |  |  |  |X |  |  |  |
NoisePacker30   |X |X |X |X |X |X |  |  |  |  |X |X |X |  |  |  |X |  |  |  |
NoiseTracker    |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |  |X |
OctaMed         |X |X |X |  |  |  |  |  |X |X |X |X |X |X |X |  |  |  |  |  |
Oktalyzer       |X |X |  |X |  |  |  |  |  |  |X |X |X |X |  |  |  |  |  |  |
ProTracker      |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |  |X |
PumaTracker     |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 |  |  |  |  |X |  |
                X
                |  |
RonKlaren       |  |  |X |  |  |X |  |  |  |X |  |  |  |  |  |  |X |  |  |  |
ScreamTracker30 |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |  |X |
SID             |X |X |X |  |  |  |X |X |X |X |  |  |  |  |  |  |  |  |
                X
                |  |
SidMon10        |X |X |X |X |  |X |  |  |  |  |X |X |  |  |  |  |X |  |  |  |
SidMon20        |X |X |X |X |X |X |X |  |  |  |X |X |  |  |  |  |X |  |  |  |
SonicArranger   |X |X |X |X |  |X |X |  |X |X |X |X |  |X |X |X |X |  |  |  |
SoundControl    |X |X |X |X |  |X |X |X |  |  |X |X |  |  |  |  |X |  |  |  |
SoundFX13       |X |X |X |X |X |X |X |  |X |  |X |X |X |X |X |X |X |  |  |  |
SoundFX20       |X |X |X |X |X |X |X |  |X |  |X |X |X |X |X |X |X |  |  |  |
SoundMon20      |X |X |X |X |  |X |  |X |  |  |X |X |X |  |  |  |X |  |  |  |
SoundTracker15  |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |  |X |
SoundTracker31  |X |X |X |X |X |X |X |X |X |  |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 |X |  |  |  |
StarTrekker4AM  |X |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 |  |  |  |  |X |  |  |  |
TakeTracker     |X |X |X |X |X |X |X |X |X |  |X |X |X |X |X |X |X |X |  |X |
TFMX_15         |X |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 |  |  |  |  |  |  |  |  |
TME             |X |X |X |X |X |X |X |X |X |X |X |X |  |X |X |X |X |  |  |  |
TronicTracker   |X |X |X |X |  |X |X |  |  |  |X |X |  |  |  |  |X |  |  |  |
VectorDean      |  |  |X |X |  |X |  |  |  |X |  |  |  |  |  |  |X |  |  |  |
Whittaker       |  |  |X |  |  |X |  |  |X |X |  |  |  |  |  |  |X |  |  |  |
----------------+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

* For further info see
                Tunes
                .
```

1.68 iffcfg

IFF-8SVX Config:

Max CHIP buffer size in Kb: Everything bigger than this will be handled as a big sample; that is, it will be loaded in segments.

Load buffer size in Kb: The amount of used chip mem for a big sample. Mono samples will use 2 times the value (because of doublebuffering) and a stereo sample will use 4 times the value.

Always loop small samples: APlayer will always play a loop defined in a small sample (but not in big samples). But if you want AP to loop a whole small sample if there aren’t any loop settings in the sample, check this. For big samples, use the loop-gadget in the main window.

Auto load on small samples: Defines whether APlayer should jump to the next file in the module list when it has finished a small sample. APlayer always jumps to the next module when the sample is big (unless the loop-gadget is used).

1.69 aiffcfg

IFF-AIFF Config:

Max CHIP buffer size in Kb: Everything bigger than this will be handled as a big sample; that is, it will be loaded in segments.

Load buffer size in Kb: The amount of used chip mem for a big sample. Mono samples will use 2 times the value (because of doublebuffering) and a stereo sample will use 4 times the value.

Always loop small samples: APlayer will always play a loop defined in a small sample (but not in big samples). But if you want AP to loop a whole small sample if there aren't any loop settings in the sample, check this. For big samples, use the loop-gadget in the main window.

Auto load on small samples: Defines whether APlayer should jump to the next file in the module list when it has finished a small sample. APlayer always jumps to the next module when the sample is big (unless the loop-gadget is used).

1.70 riffcfg

RIFF-WAVE Config:

Max CHIP buffer size in Kb: Everything bigger than this will be handled as a big sample; that is, it will be loaded in segments.

Load buffer size in Kb: The amount of used chip mem for a big sample. Mono samples will use 2 times the value (because of doublebuffering) and a stereo sample will use 4 times the value.

Always loop small samples: APlayer will always play a loop defined in a small sample (but not in big samples). But if you want AP to loop a whole small sample if there aren't any loop settings in the sample, check this. For big samples, use the loop-gadget in the main window.

Auto load on small samples: Defines whether APlayer should jump to the next file in the module list when it has finished a small sample. APlayer always jumps to the next module when the sample is big (unless the loop-gadget is used).

1.71 sidcfg

SID Config:

Use Rewind Buffers: If you want to be able to use rewind when you are playing SID modules, then check this.

```
Warning: This uses  a lot of memory  (approximately 0.5 kb/sec), so don't
use it if you can't live without the memory!

Rewind Speed: This slider is used to decide how much the tune  should  be
rewound  every time you push the  rewind button. (For advanced users: The
amount is the number of frames that should be rewound).

Forward Speed: This slider is used to decide how much the tune should  be
forwarded  every time you push  the forward  button. (For advanced users:
The amount is the number of frames that should be forwarded).

I love copying text frames... :-)
```

1.72 smuscfg

```
IFF-SMUS Config:

Instrument Path: The full path  or an assign name  to the place where the
instruments for your SMUS  modules is to be found. The path can be chosen
by pressing the diskgadget to the right of the string gadget.
```

1.73 players

```
  Players:
  --------

  ActionAmics:      This player is used in the game  "DynaBlaster",  so  you
                    can play the music from this game and rip the samples if
                    you like.

----------------------------------------------------------------------------

  DeliCustom:       Yes,  that's  right.  Now  is  APlayer  able   to   play
                    Delitrackers  custom  modules.  Currently  there  is not
                    support  of  all  Delitrackers global functions or tags,
                    only  a  few. The modules I have, work just fine, but if
                    you  have  a module that doesn't work, please send it to
                    me.  You  can not pack these modules, because the player
                    has to load them as object files.

----------------------------------------------------------------------------

  DigitalMugician: Nice player with some cool sound effects. The program is
                    not the most system friendly I have seen.

----------------------------------------------------------------------------

  DSS:              When you buy the DSS sampler, you will get a  music  and
                    sampler  program. The music program will save modules in
                    this  format,  so  with this player you are able to play
                    these type of modules.
```

```
-----------------------------------------------------------------------------

  FastTracker:      I don't  know much  about this  player.  It's a 6 and  8
                    channels  ProTracker player. It  can do the  same as the
                    ProTracker  player, which  means  that it  has the  same
                    commands etc. This player uses a NotePlayer.

-----------------------------------------------------------------------------

  FastTracker20:    This player can play the new format on the PC,  the *.XM
                    modules. The original music program on the PC is made by
                    Triton.  This  player  is coded by Jarno Paananen and it
                    uses a NotePlayer.

-----------------------------------------------------------------------------

  Fred:             There is a lot of some nice fred tunes on  aminet,  some
                    of them by Jogeir. There is a player inside the modules,
                    but  I  have ripped one of them and added some features.
                    It  will  ofcourse  use  the player inside the module if
                    it's not the same as the one I know. Currently I support
                    2 different players, which only differs slightly.

-----------------------------------------------------------------------------

  FTM:              This is a 8 voices player with one of  the  best  mixing
                    routine ever heard on the Amiga.

-----------------------------------------------------------------------------

  Future13:         This  is  probably  the  most  used  type  of modules in
                    intros, because of the short module size.

-----------------------------------------------------------------------------

  Future14:         This is the newer version of Future Composer. It can now
                    handle different wavetables.

-----------------------------------------------------------------------------

  Hippel:           Argghh, this is one of the hard ones.  There is a player
                    inside each module, but they are all different.
                    You  will probably get a Hippel module this player can't
                    recognize.

-----------------------------------------------------------------------------

  Hippel-COSO:      This is Hippel  modules  without  a  player  inside  the
                    module.  It  will  try  to load an extra sample  file if
                    needed,  but this hasn't  been tested,  because we don't
                    have any of these modules.  If you have one, please send
                    it to us.

-----------------------------------------------------------------------------

  IFF-8SVX:         This player can  play the IFF-8SVX  sample format. It is
                    the most used format on the Amiga. It can play any  size
```

```
                   of  samples  and  if the sample is too big, it will load
                   one part  of it  at a time. It can  also  handle  stereo
                   samples  and  IFF-crunched  samples,  in  both  mono and
                   stereo.

---------------------------------------------------------------------------

  IFF-AIFF:        This format is used on  the  Macintosh.  This player can
                   play  both 8 and 16 bit samples in both mono and stereo.
                   16 bit samples will be converted to 8 bit realtime while
                   playing.

---------------------------------------------------------------------------

  IFF-SMUS:        This is a very old music format.  The format is designed
                   to  use with midi. This player will only play upto the 4
                   first priority voices.

---------------------------------------------------------------------------

  InStereo!:       This player is the  successor of Synthesis.  It can make
                   some nice chip sounds.

---------------------------------------------------------------------------

  JamCracker:      Is not the most used format,  but  there  is  some  nice
                   tunes available made by Dr. Awesome.

---------------------------------------------------------------------------

  MajorTom:        I only got two modules in this format and has never seen
                   the music program. Anybody have it?

---------------------------------------------------------------------------

  MED:             This player has one of the biggest  sources I  have ever
                   seen  (about  80Kb).  This player will play 4 voices MED
                   modules and can handle MMD0, MMD1 and MMD2 module types.

---------------------------------------------------------------------------

  MON:             Maniacs of Noise have created this player. The player is
                   inside the modules.

---------------------------------------------------------------------------

  MusicAss:        This format has  very  short  modules  with  the  player
                   inside the modules. I have ripped the player and fixed a
                   volume  bug  so  they  will  now  work on A4000. It will
                   of course only use the player inside the module if it is
                   not the same as the ripped one.

---------------------------------------------------------------------------

  MultiTracker:    This tracker is  a 1-32 channels  ProTracker player. The
                   file format is not the same as in normal mod files. This
                   player uses a NotePlayer.
```

```
-----------------------------------------------------------------------------

  NoisePacker20:   This format is a packed NoiseTracker format  written  by
                   Twins of Phenomena.

-----------------------------------------------------------------------------

  NoisePacker30:   Is a newer version of the NoiseTracker packer.

-----------------------------------------------------------------------------

  NoiseTracker:    Before ProTracker this format was  the most used.  I use
                   the  same  player  as  in  the ProTracker player, but it
                   can't  change  the  CIA  tempo and it doesn't have any E
                   commands   except   for  the  filter  command.  See  the
                   ProTracker description for more information. This player
                   uses a NotePlayer.

-----------------------------------------------------------------------------

  OctaMed:         This player is almost the same as the MED player, except
                   that it play 5-8 voices modules.

-----------------------------------------------------------------------------

  Oktalyzer:       A very old 8 voices format. The mixing routine will only
                   work with  good  results on a 68000. I have some modules
                   that  will  sometimes  lock my computer (A1200). It will
                   in the near future hopefully use a NotePlayer instead of
                   the intern mixing routine.

-----------------------------------------------------------------------------

  ProTracker:      This format is probably  the most  known  format ever on
                   the  Amiga.  I  use  the  ProTracker version 1.1b player
                   written  by  Lars  "Zap" Hamre/Amiga Freelancers. I have
                   fixed  a lot of bugs (see the history :) and I have also
                   optimized  it  so  it  won't use so much CPU time as the
                   original player does. The  main  optimizing is that I've
                   changed all  the patterns  note periods to index numbers
                   in the  period  tables.  This removed  the routine which
                   searched  for the period in the period tables to get the
                   index  in  the finetunes period tables. This player uses
                   a NotePlayer to play the notes.

-----------------------------------------------------------------------------

  PumaTracker:     I don't know anything about this player.  I  only have a
                   few modules of this type.

-----------------------------------------------------------------------------

  QuadraComposer:  Got the player from the author,Bo Lincoln.  Supports CIA
                   and all commands without crashing like in other players.

-----------------------------------------------------------------------------
```

RIFF-WAVE:	This format is used on the PC. It can play both 8 and 16 bit samples in mono or stereo. If the sample is too big, it will be loaded in parts.
RonKlaren:	This player will play EaglePlayer Ron Klaren modules. This is the only type of module which cannot be crunched because the files are loaded as object files. There can be some problems on A4000.
ScreamTracker30:	This module format is probably one of the most used on the PC. This player will play that format. Jarno Paananen has coded the original player. I have modified it a little bit, so it won’t try to play adlib samples There will be silence instead of some crap sound.
SID:	This play has to use the playsid.library to run. It can play all kinds of C64 songs by emulating the CPU and SID in the C64.
SidMon10:	Player is inside the module. I have ripped the player and added a few features. If the player isn’t the same as the player inside the module, the inside player will be used instead.
SidMon20:	This is the new version of SidMon. It hasn’t the player inside the module anymore.
SonicArranger:	This player will play both songs and modules with player inside the module. With modules it will use the song player.
SoundControl:	Nothing to say about this player.
SoundFX13:	This player looks a lot like the SoundTracker player.
SoundFX20:	A new version of this program. It now supports 31 samples!! (whauww ;-)

```
---------------------------------------------------------------------------

  SoundMon20:       Nothing to say about this player.

---------------------------------------------------------------------------

  SoundTracker15:   In this very OLD tracker only 15  samples  were allowed.
                    I  use  the  same  player  as in the ProTracker  player,
                    but it only supports the SoundTracker commands.
                    See  the  ProTracker  description  for more information.
                    This player uses a NotePlayer.

---------------------------------------------------------------------------

  SoundTracker31:   This is also a very OLD tracker but  it has 31  samples.
                    As  always  I  use  the same player as in the ProTracker
                    player.  See that description for more information. This
                    player uses a NotePlayer.

---------------------------------------------------------------------------

  StarTrekker4:     This player will play the normal  4  voices  StarTrekker
                    modules.

---------------------------------------------------------------------------

  StarTrekker4AM:   This player will play 4 voices StarTrekker modules  with
                    AM sounds. It needs the extra .nt file to the module.

---------------------------------------------------------------------------

  Synthesis:        See InStereo!

---------------------------------------------------------------------------

  TakeTracker:      This  player   is  exactly   the  same   player  as  the
                    FastTracker  player,  except that there is another  test
                    routine. This player supports 1 to 32 channels modules.

---------------------------------------------------------------------------

  TFMX_15:          This player can play the old TFMX formats. It’s a format
                    created by Chris Hülsbeck.

---------------------------------------------------------------------------

  TFMX_7V:          This player can play the  7  voices  TFMX  modules.  The
                    player is coded by J. Hippel and C. Hülsbeck.

---------------------------------------------------------------------------

  TFMX_Pro:         Well, nothing more to say, except that this is the newer
                    version of TFMX modules.

---------------------------------------------------------------------------

  TME:              Nothing to say about this player.
```

```
---------------------------------------------------------------------------

  TronicTracker:   Nothing to say about this player.

---------------------------------------------------------------------------

  VectorDean:      This player is used  in  the  game  "Canon Fodder 2".  I
                   don’t know if it was used in number 1 too.

---------------------------------------------------------------------------

  Whittaker:       Player inside the module. The players are different from
                   each other, so you can get some problems.
```

1.74 noteplayers

```
                NotePlayers:
  ------------

  14BitStereo-32Voices:   Can mix upto 32 voices, into the four amiga sound
                          channels. The samples can be 8 bit and placed  in
                          both chip- and fast-mem. The mixing routine is 16
                          bit. If the mixing routine runs out of cpu  time,
                          the sample rate is lowered.

                          The main routine  is  based  on  Jarno  Paananens
                          routines, with some fixes and additions by me.

                          See the
                 32 Voices configuration
                  for more info.

---------------------------------------------------------------------------

  Fastmem-4Voices:        This  play  modules  from  fastmemory. It  handles
                          both signed and unsigned samples.

                          See  the
                 Fastmem 4 Voices Configuration
                  for  more
                          info.

---------------------------------------------------------------------------

  Mono-32Voices:          Can mix upto 32 voices, into the four amiga sound
                          channels. The samples can be 8 bit and placed  in
                          both  chip-  and  fast-mem. The mixing routine is
                          only 8 bit.

                          Please note that the mixer always plays in  mono,
                          even  if  some  channels  are  turned off. If the
                          mixing routine runs out of cpu time,  the  sample
                          rate is lowered.
```

```
                           The main routine  is  based  on  Jarno  Paananens
                           routines, with some fixes and additions by me.

                           See the
                 32 Voices configuration
                  for more info.

-----------------------------------------------------------------------------

  Paula-4Voices:           Passes on the data it gets from the player to the
                           sound chip. The samples can only be placed in the
                           chip memory.

-----------------------------------------------------------------------------

  ReSurround-32Voices:     If  you  are  the  lucky  owner  of  a  surround
                           amplifier, you can connect your Amiga to  it  and
                           use  this player to listen to the modules in REAL
                           surround :).

                           It can mix upto 32 voices, into  the  four  amiga
                           sound  channels.  The  samples  can  be 8 bit and
                           placed  in  both  chip and fast-mem.  The  mixing
                           routine is only 8 bit.
                           Please note that the mixer always plays  in  real
                           surround,  even  if some channels are turned off.
                           If the mixing routine runs out of cpu  time,  the
                           sample rate is lowered.

                           The main routine  is  based  on  Jarno  Paananens
                           routines, with some fixes and additions by me.

                           See the
                 32 Voices configuration
                  for more info.

-----------------------------------------------------------------------------

  Stereo-32Voices:         Can mix upto 32 voices, into the four amiga sound
                           channels. The samples can be 8 bit and placed  in
                           both  chip-  and  fast-mem. The mixing routine is
                           only 8 bit. If the mixing routine runs out of cpu
                           time, the sample rate is lowered.

                           The main routine  is  based  on  Jarno  Paananens
                           routines, with some fixes and additions by me.

                           See the
                 32 Voices configuration
                  for more info.

-----------------------------------------------------------------------------

  Surround-32Voices:       Can mix upto 32 voices, into the four amiga sound
                           channels. The samples can be 8 bit and placed  in
                           both  chip-  and  fast-mem. The mixing routine is
                           only 8 bit.
```

Please note that the mixer always plays in surround, even if some channels are turned off. If the mixing routine runs out of cpu time, the sample rate is lowered.

The main routine is based on Jarno Paananens routines, with some fixes and additions by me.

See the
32 Voices configuration
for more info.

1.75 fastcfg

Fastmem 4 Voices 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 configurations, 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.

1.76 32config

32 Voices Configuration:

First of all you have to choose which mixing routine you want to use. This can either be the 68000/68010 or 68020+. The 68020+ mixer is ofcourse optimized for higher processors.

This configuration is used to set the mixing rate relative to the number of channels used in a module. This means that the higher mixing rate, the better sound. But remember that high mixing rates demands more cpu power. As soon as you press return, after you have typed a new mixing rate, the new mixing rate is used. This is ofcourse only if you have changed the mixing rate for the current number of used channels.

You can also set the volume boost for every number of channels. If you are listening to modules that uses a lot of channels, you can take advantage of volume boost. This is because if a lot of channels has to be mixed, the volume will be lowered, and therefore it needs to be boosted. The boost value will only take effect when a new module is played.

If you don’t know which values to type in, you can get APlayer to choose the values best suited for your system. This can be done by pressing the autoadjust button. It CAN’T be done if APlayer is playing a module, or the audio channels is allocated by another program.

First, you are asked if you want to check all the channels. If you answer NO, only the 31-32 channel mode is tested, and the result is used in all other channels mode, else every channel mode is tested.

Then you can decide the maximum mixing rate to be used. This can be a number between 4000 and 56000. The higher the mixing rate, the better the sound quality.

The last two requesters will let you decide wether you want to adjust the found mixing rate a bit down (0-100 percent), which will leave more cpu power to the rest of the system.

After this the test is performed. To optimize the results, do NOT touch your computer while the test is in progress. You can adjust the volume in the main window, if you don't want to hear the tones while the test is running.

Please note that the numbers found in the test may not be the optimal setting, but should give you a hint on how the setting should be.

Click
here
to see how to use the menus in this window.

1.77 tooltypes

Tooltypes and CLI arguments:

(Lines in italic explains what will happen when APlayer is started with the argument/tooltype and APlayer is running already.

CX_POPUP (POP): Defines if you want the APlayer to open its window at startup, or start in hidden 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.
The modules will be added to the list.

MODULELIST (ML): The name of a modulelist file you want APlayer to use. For instance S:example.APML. The modules in the modulelist will be placed before any modules specified on the Command line.
The modules in the modulelist will be added to the list.

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 APlayer to open. Remember that the name is case SenSitiVe. Default is the default public screen, e.g. Workbench.
APlayer will close all the APlayer windows and reopen them on the specified public screen.

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

INFOOPEN (IO): This will force the info window to open at start.
Same effect.

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

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

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.
This will shuffle all modules in the modulelist, including any just added modules.

PATH (P): This option takes one argument, which is a path. The path will be scanned for modules, just like the
scandir
option. This can be used
in conjunction with the UNIQUE argument to add files from a dir, which is constantly updated.

UNIQUE (U): This will avoid any modules to be represented more than once in the modulelist. This works both with modules, the "Path" argument and lha files.

NOTE: If you specify any modules, a modulelist, or use the PATH argument, your
scandir
will be ignored.

These parameters/tooltypes can ONLY be used when APlayer is already running:

JUMP (J): This will automatically jump to the first module just added, independent of your
config setting
.

QUIT (Q): This will quit the player, WITHOUT executing the arexx macro defined in your
settings
.

1.78 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.

Jarno Paananen for  the source to  his great PS3M, which  made us able to
support S3M, Fasttracker][ :) and many other formats ...

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

Marley/Infect for a lot of different moduletypes.

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.

KiLLraVeN/MYSTiC Thanks  to  you for  the  bugreports. Keep  them coming.
(NOT :^) And for the fabulous Arexx script concerning modulelists.

Nemesis1 for the sheep module.

Kaikumaa Timo for the SoundTracker15 module.

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

1.79 history

```
                History:


              1.00

              1.01

              1.02
```

1.03

1.04

1.10

1.20

1.21

1.30

1.80 h100

```
1.00 (Released 2-Sep-1994)
--------------------------
- First public release.
```

1.81 h101

```
1.01 (Not released)
-------------------
- Ups!!! Fixed a bug in the tune  selector.  You  could  only  select  10
  tunes, even if there were more.
- Fixed a bug in the Protracker player. It made  a  noisy  sound  if  you
  played an empty sample with the volume command.
- A little bug fixed in the double buffering loader. If  the  loader  has
  loaded the next module into the memory and you then delete it from  the
  list  before it will start, the player will play it anyway. Now it will
  load the next module instead.
- If you press the 's' key in the main window,  the  sample  info  window
  will always open, even if the gadget was ghosted. This is fixed now.
```

1.82 h102

```
1.02 (Not released)
-------------------
- The global APG_WaitDMA routine is  changed,  so  it  will  use  EClocks
  instead of raster lines. This mean the global data APG_Hz is obsolete.
- When APlayer tries to open a file, and it couldn't locate it,  it  will
  now show the filename it tries to open instead of a simple "Object  not
  found" requester.
- A little bug in the configuration loader. It didn't show an error if it
  couldn't open the file and it cleared all players instead of  only  the
  extern.
- The module loader will now show an open error requester.
- The "Delete from list" is changed to "Remove from list".
- Started to make ARexx interface.
```

1.83 h103

```
1.03 (Not released)
-------------------
- Now APlayer only allocates the channels when a  module  is  in  memory.
  This means, all players are changed a bit and the two global  functions
  APG_AllocChannels and APG_FreeChannels are also changed a bit.
- Fixed a bug in the IFF-8SVX and RIFF-WAVE player config routine.
- Changed the SID player to use the playsid.library.
```

1.84 h104

```
1.04 (Not released)
-------------------
- Added the FTM, Whittaker and the TFMX 1.5 player.
- Changed the loader routine, so now it will cut any fileextension if  it
  couldn't open the file first time. This improves the TFMX loader.
- A minor bug in the SoundMonitor replayer fixed.
- A new version of the TFMX 7-Voice player is added. Now you can play the
  Turrican III main tune without a crash after 3 minutes.
- Also a new version of the TFMX-Professional player is inserted. It will
  not use VBlank anymore.
- Fixed some small bugs here and there.
```

1.85 h110

```
1.10 (Released 27-Dec-1994)
---------------------------
- Changed all the windows to be auto adjusted.
- The sample info & the module list windows  are  now  intelligent.  That
  means, they will auto resize to the number of items in the list.
- Added zoom gadget in the titlebar in the main window.
- Removed the window positions & sizes from the main preference file into
  and new preference file called APlayer.win. See the docs for more info.
- Fixed a major bug in the free module routine.  Sometimes  it  closes  a
  library that should not be closed, such as the gadtools.library.
- Fixed a bug in the QuadraComposer player. It crashed when  it  plays  a
  sample that doesn't exists.
- Fixed another bug in  the QuadraComposer  player. It crashes the second
  time a module was started.
- Now you can use the  cursor keys and the return key in the Sample Info,
  Module Editor, FSS and Players window.
- Players config files are moved to a subdirectory.
- You  can now  use a lots  of keys to  control the main  window  gadgets
  (hotkeys).
- Added the Hippel, Hippel-COSO and the PumaTracker player.
- Added the IFF-AIFF player, and I hate IEEE numbers :^(.
- Changed the RIFF-WAVE player, so it also support 16-bits samples.
- You  can  now delete modules from your FSS list now!!! It was like hell
  to make a delete function in a binary tree.
- The title in the main window are now font sentitive :)
- The  about  window  are  made  to  a separate window, just like all the
```

```
  others.  That  means  it  will  be  updated  every time a new module is
  loaded.
- HolyNoise player changed to MajorTom player.
```

1.86 h120

```
1.20 (Released 15-Mar-1995)
---------------------------
- Optimized  the keypad playing. Now will APlayer first start to play the
  sampling and then update the sample window.
- You can now use the keypad to select a sample when you play a module.
- Fixed a little bug in all the double click routines.
- Fixed  a  bug  in the JamCracker player. If you played on the keyboard,
  the sample sounded strange.
- Fixed a bug in the loader routine.
- Changed  the  about  window,  so  now  you  can  switch between the two
  "windows".  It  will also  be  updated  every time a module changes CIA
  tempo.
- Now  you  can  encrypt  your modules with xpk. APlayer will ask for the
  password. It will also unpack recursively. That means you can pack more
  than  one time. You can use this, if you want to pack your modules with
  SQSH and then encrypt them.
- Fixed a bug in the players own  config. Didn’t put  the window to front
  if it was already open.
- Added  volume  and  channel control and removed an  enforcer hit in the
  Fred player.
- Added channel  control and song-end to  the Music-Assembler  player and
  fixed  a bug  in the  player (not my fault!), so  now it also  works on
  an A4000.
- The  Hippel player  is totally rewritten. Still some problems with some
  tunes.
- Added a lot of new global functions and a few tags.
- Added the IFF-SMUS player.
- Added  some  new  CLI  and tooltype  arguments + changed some of them a
  little bit. Now you can start APlayer again with new arguments.
- Fixed a little bug in the channel on/off keyboard routine.
- Now you can press return in the about window to close it.
- Added the Action Amics, In Stereo!, SidMon 1.0 and SidMon 2.0 players.
- I forgot  to set  the IntuiTicks  IDCMP flag  in all the  windows where
  there was a listview gadget. This is  fixed now, so there should be a
  repeat on the arrows.
- Added  the  Sonic  Arranger, Sound Control, SoundFX 1.3 and SoundFX 2.0
  player.
- Fixed some small bugs here and there.
- Added the TME, Tronic Tracker and the VectorDean player.
- A  big bug  fixed (when APlayer tries to load the next module in hidden
  mode).
- Some  small  bugs in  the Protracker player (Thanks to KiLLraVeN/MYSTiC
  for telling us).
- Changed  all the  listviews. Now  I have  implemented  my own  listview
  routines, and  that means  that the look is the  same on both KS3.0 and
  KS2.0!!!
- Now the FSS will find out if you have moved some  modules next time you
  hear them.
- Added Up/Down arrows in the module list editor.
- Added new Med/OctaMed players. Now supports MMD0/MMD1/MMD2 files.
- Added multiplay in the sample info window (press "DEL")
- New intelligent author finding system implemented.
```

1.87 h121

```
1.21 (Released 21-Mar-1995)
---------------------------
- Due  to  a BIG bug  in version 1.20, which  made APlayer  unuseable  on
  machines below  OS3.0, we  have released this  version, which have been
  succesfully tested on an A600 (OS2.0).
- The author finding system have been improved on some points.
- The  name of  the actual module  is displayed in  the titlebar when the
  window gets zipped.
- A few bugs have been fixed in the SID-, and the MusicAssembler-players.
- The default  hotkeys in the config  have been modified  a bit, we found
  out  that  "control" couldn't  be  abbreviated  as "ctrl". But  all the
  hotkeys works as always.
```

1.88 h130

```
1.30 (Released 29-May-1995)
---------------------------
- Added menu in the ARexx configuration window.
- Changed the player config window design and added menus.
- Small minor bugs/fixes.
- Intern ProTracker to ProTracker/NoiseTracker/SoundTracker 15/31
- New Player Configuration file format (NOT BACKWARDS COMPATIBLE).
- Made the Paula noteplayer
- Added 2 new ARexx commands: GetNotePlayer & GetChannels.
- Renamed the Arexx command "GetType" to "GetPlayer"
- Added the NotePlayer bit in the "GetInfo" ARexx commando.
- Changed the Sample Info window.
- Added the Stereo-32Voices noteplayer.
- Added the FastTracker, TakeTracker and MultiTracker player.
- Added  the   FastMem-4Voices,   Mono-32Voices   and   Surround-32Voices
  noteplayers.
- Optimized the APG_CalcVolume function.
- Added the RealSurround-32Voices and 14Bit-32Voices noteplayer.
- Added the ScreamTracker 3.0 and FastTracker 2.0 player.
- Now  can  you  change  the  tempo  on all players which uses the intern
  interrupt.
- When you close the main config window, you will not automatic close the
  player and arexx window anymore.
- Added key file system.
- Fixed a little bug in the RIFF-WAVE player.
- The  RexxMsg  field  in  the  main cycle gadget is removed if ARexx not
  turned on.
- Added  to  the  module  list window titlebar the number of items in the
  module list.
- Added the DeliCustom player.
- Fixed  bug  that prevented  resizing of ModuleList-window  after adding
```

```
    modules from CLI.
  - Fixed  bug that  tried to load  LhA-icons dropped  on the icon  or main
    window as normal modules.
```

1.89 contact

```
  Send bugreports, new players including modules and suggestions to

  Coder:
  ------

  Thomas Neumann (Tax)
  Kongensgade 78
  3550 Slangerup
  Denmark

  Send technotapes and modules to

  Designer & Alphatester:
  ------------------------

  Jakob Langgaard (Jail)
  Krebsen 101
  3650 Ølstykke
  Denmark

  E-mail: no more :-( (going to the army for 8 months)

  Designer & Betatester:
  ----------------------
  Asger Høgsted

  bims@diku.dk (internet)
```

1.90 arexxmain

```
                 ARexx with the APlayer:

  This section will describe all  the  commands  available  for  the  ARexx
  interface  in APlayer. If you don't know anything about ARexx, you should
  either skip this part, or try to find some documentation on the subject.

  To help you to get a better overview of the commands, we have split  into
  the following sections:



                 Channels
                 - Manipulate Channels

                 Eject
                 - Controls the Eject Function
```

```
Loop
- Change Loop

Main Cycle
- Manipulate the Main Cycle

Modulelist
- Manipulate the Module list

Pause
- Change the Pause state

Play
- What to Play

Requester
- Make ReqTools Requsters

Sample
- Use the Sample Functions

Speed
- Change/Get the Speed State

Tech Info
- Get a lot of Technical Info

Various
- Miscellaneous

Volume
- Change/Get the Volume State

Wind
- Forward/Rewind

Window
- Window Commands
```

1.91 archannel

```
Channels:

Command:
  ChannelOff channel

Description:
  Turns off the channel "channel".

Input:
  channel - A channel number between 1 and 4.

Result:
  None.
```

```
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    ChannelOn channel

  Description:
    Turns on the channel "channel".

  Input:
    channel - A channel number between 1 and 4.

  Result:
    None.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    state = GetChannel channel

  Description:
    Get the current state of the channel "channel".

  Input:
    channel - A channel number between 1 and 4.

  Result:
    state - This is a boolean where FALSE means off and TRUE means on.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    state = ToggleChannel channel

  Description:
    Toggles the state of the channel "channel".

  Input:
    channel - A channel number between 1 and 4.

  Result:
    state - This is a boolean where FALSE means off and TRUE means on.
```

1.92 areject

```
  Command:
    ClearList

  Description:
    This will eject the current playing module, and free the  rest  of  the
    list. Which is the same as if you've clicked the eject button 2 times.

  Input:
    None
```

```
 Result:
   None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   Eject

 Description:
   This will either eject the currently playing module or eject the  whole
   list.  If a module is being played it will be ejected, and if no module
   is loaded played the list will be ejected.

 Input:
   None

 Result:
   None
```

1.93 arloop

```
 Command:
   state = GetLoop

 Description:
   Get the current state of the loop gadget. The result is a boolean and 1
   means on and 0 means off.

 Input:
   None

 Result:
   state - This is a boolean where FALSE means off and TRUE means on.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   LoopOff

 Description:
   Set the loop to off.

 Input:
   None

 Result:
   None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   LoopOn

 Description:
   Turns the loop on.
 Input:
   None

 Result:
   None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   state = ToggleLoop

 Description:
   Toggles the loop and return the new state.

 Input:
   None

 Result:
   state - This is a boolean where FALSE means off and TRUE means on.
```

1.94 armaincycle

```
 Command:
   name = GetCycle

 Description:
   Get the current position of the main cycle. It  will  return  a  string
   with the cycle name, like Author.

 Input:
   None

 Result:
   name - a string with the name of the cycleposition.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   RexxMsg string

 Description:
   Changes the contents of the REXXMSG in the main cycle.

 Input:
   String - a normal text string.

 Result:
   None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   error = SetCycle name
```

```
Description:
  Set the main cycle to the "name", like SetCycle Author. It will  return
  an error if it couldn't find the name.

Input:
  name - a string with the name of the cycleposition.

Result:
  error - an error code which indicates if it could find the name in  the
          cycle. 0 means ok and 1 means error. Note that the result is in
          RC and not in RESULT.
```

1.95 armodulelist

```
Command:
  error = AddMod file

Description:
  Adds the module with the name "file" to the module list.

Input:
  file - the name of the module you want to add.

Result:
  error - if it can't find the file. 0 means ok and 1 means  error.  Note
          that  the  result  is  in  that  the result is in RC and not in
          RESULT.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  AppendList num file

Description:
  Adds a module list with the name "file" to the module  list.  "num"  is
  the number in the list you want to insert the new list at. -1 means the
  buttom of the list.

Input:
  num - the position where you want to insert.
  file - the modulelist to insert.

Result:
  error = this is an error code where 0 means ok and 1 means error.  Note
          that the result is in RC not in RESULT.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  error = DelMod num

Description:
  Deletes the module "num" in the module list.

Input:
    num - the number of the module to be deleted from  the  list,  starting
          from 1.

  Result:
    error = this is an error code where 0 means ok and 1 means error.  Note
            that the result is in RC not in RESULT.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    error = ExgMods num1 num2

  Description:
    Exchange the two modules "num1" and "num2" in the module list.

  Input:
    num1 and num2 - the number of the modules  to be exchanged in the list,
                    starting with 1.

  Result:
    error = this is an error code where 0 means ok and 1 means error.  Note
            that the result is in RC not in RESULT.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    max = GetMaxNames

  Description:
    Returns the number of modules in the module list.

  Input:
    None

  Result:
    max - the number of modules in the list.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    name = GetModName num

  Description:
    Returns the name of the module with the number "num" with full path.

  Input:
   num - the number of the module from which you want  the  name,  starting
         with 1.

  Result:
    error = this is an error code where 0 means ok and 1 means error.  Note
            that the result is in RC not in RESULT.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    num = GetModNum name
  Description:
    This will return the number of the module named "name". If it  couldn't
    find the name, a zero will be returned.

  Input:
    name - the file name (without path) of the module in the list.

  Result:
    num - the number of the module in the  list.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    LoadList file

  Description:
    Loads a new APML list with the filename "file".

  Input:
    file - the name of the modulelist to be loaded.

  Result:
    None

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    error = SaveList file

  Description:
    Saves the current module list with the filename "file". If  the  module
    list is empty, an error will be returned.

  Input:
    file - the name of the modulelist to be saved.

  Result:
    error = this is an error code where 0 means ok and 1 means error.  Note
            that the result is in RC not in RESULT.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    Shuffle

  Description:
    Shuffles the modulelist.

  Input:
    None

  Result:
    None

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
Command:
  SortList

Description:
  Sorts the modulelist in alphabetical order.

Input:
  None

Result:
  None
```

1.96 arpause

```
Command:
  error = Pause

Description:
  Pauses the current module. If the current  player can’t pause or  there
  aren’t any modules in the memory, an error will be returned.

Input:
  None

Result:
  error = this is an error code where 0 means ok and 1 means error.  Note
          that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  error = TogglePause

Description:
  Toggles the pause state of the current module. If  the  current  player
  can’t pause or there aren’t any modules in the memory, an error will be
  returned.

Input:
  None

Result:
  error = this is an error code where 0 means ok and 1 means error.  Note
          that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  error = Unpause

Description:
  Starts playing again. If the current player can’t pause or there aren’t
  any modules in the memory, an error will be returned.

Input:
  None

Result:
  error = this is an error code where 0 means ok and 1 means error.  Note
          that the result is in RC not in RESULT.
```

1.97 arplay

```
  Command:
    Play

  Description:
    This will do the same as if you pressed the play button.

  Input:
    None

  Result:
    None

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    error = PlayMod num

  Description:
    Load and start the module with the number "num" (1 - x) in  the  module
    list. If the "num" doesn't exist, an error will be returned.

  Input:
    num - the number of the module you want to play, starting with 1.

  Result:
    error = this is an error code where 0 means ok and 1 means error.  Note
            that the result is in RC not in RESULT.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    error = PlayRNDMod

  Description:
    Loads and starts a random module from the list.

  Input:
    None

  Result:
    error = this is an error code where 0 means ok and 1 means error.  Note
            that the result is in RC not in RESULT.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    error = PlayTune num
  Description:
    Start to play the tune "num".  An error is returned if the "num" is out
    of range. Note that this is for subsongs, like in sid modules.

  Input:
    num - the number of the tune you want to play, starting with 1.

  Result:
    error = this is an error code where 0 means ok and 1 means error.  Note
        that the result is in RC not in RESULT.


 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    NextMod

  Description:
    Loads and plays the next module in the list. If only one module in  the
    list,  nothing  will  happend.  If  an error occurs the player will act
    exactly the same way as if you've pressed the next module button in the
    main window.

  Input:
    None

  Result:
    None

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    PrevMod

  Description:
    Loads and plays the Previous module in the list. If only one module  in
    the  list, nothing will happend. If an error occurs the player will act
    exactly the same way as if you've pressed the Previous module button in
    the main window.

  Input:
    None

  Result:
    None
```

1.98 arrequester

```
 Command:
   dir,success = GetDir title

 Description:
   Popups a file requester where the user can pick one directory.  If  the
   user  selects cancel, "success" will be 1, else it will be 0.
 Input:
   title - a string which contains the name of the requester title.

 Result:
   dir     - the path the user has selected.

   success - this is an error code where 0 means ok  and  1  means  error.
             Note  that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   name,success = GetFile title

 Description:
   Popups a file requester where the user can pick one file. If  the  user
   selects cancel, "success" will be 1, else it will be 0.

 Input:
   title - a string which contains the name of the requester title.

 Result:
   file    - the filename with path which the user has selected.

   success - this is an error code where 0 means ok  and  1  means  error.
             Note that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   num,success = GetNumber min max title

 Description:
   Popups a number requester where the user can write  a  number.  If  the
   user selects cancel, "success" will be 1, else it will be 0.

 Input:
   min   - the minimum of the range.
   max   - the maximum of the range.
   title - a string which contains the name of the requester title.

 Result:
   num     - the number the user has written.

   success - this is an error code where 0 means ok  and  1  means  error.
             Note that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   string,success = GetString title

 Description:
   Popups a string requester where the user can write  a  string.  If  the
   user  selects  cancel,  "success"  will  be  1,  else it will be 0. The
   "string" will be the entered string.
```

```
Input:
  title - a string which contains the name of the requester title.

Result:
  string  - the string which the user have typed.
  success - this is an error code where 0 means ok  and  1  means  error.
            Note that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  pressed = Request gadget text

Description:
  This will popup a requester where the user can select between a  couple
  of  gadgets,  like ok and cancel. The "gadget" is a string with all the
  gadgets you want in the requester seperated with a "|",  ex."ok|cancel"
  or "yes|maybe|no".

Input:
  gadget - a string with the gadget names.
  text   - the text which is printed in the requester window.

Result:
  pressed - the number of the gadget which  the  user  has  pressed.  The
            rightmost gadget is number 0, the  rest  of  the  gadgets  is
            numbered from left to right.
```

1.99 arsample

```
Command:
  name,success = GetSampleName num

Description:
  Get the samplename from the sample number "num". If the number  is  out
  of range then success will be 1, and name will contain crap.
  The range of numbers could vary from moduleformat to moduleformat.

Input:
  num - the number of the sample from which you want the name.

Result:
  name    - the samplename
  success - this is an error code where 0 means ok  and  1  means  error.
            Note that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  success = Savesample num filename

Description:
  Saves the sample "number" with the name  "filename".  This  returns  an
  error if the sample number is out of range or the size is 0.
```

```
Input:
  num      - the samplenumber
  filename - the name you want the sample to.

Result:
  success - this is an error code where 0 means ok  and  1  means  error.
            Note that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  StopSample

Description:
  Stops the playing sample.

Input:
  None

Result:
  None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  success = PlaySample {i}num note

Description:
  Play  the  sample  "num"  with  the note "note". The "num" value can be
  between  1  and  the  max  number of samples the player can handle. The
  "note"  value can be between 0 and 35. 0 means C-1 and 35 means B-3. If
  the current player can't play the sample, an error will be returned.

Input:
  num  - the samplenumber
  note - the note number

Result:
  success - this is an error code where 0 means ok  and  1  means  error.
            Note that the result is in RC not in RESULT.
```

1.100 arspeed

```
Command:
  speed = GetSpeed

Description:
  Get the current speed slider position and place the result in  "speed".
  The result is a signed integer.

Input:
  None

Result:
    speed - the current speed

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    SetSpeed speed

  Description:
    Set the speed to "speed". "Speed"  can  range  from  -111  to  112.  If
    "speed" is  out of range  the  speed  will  be set to either maximum or
    minimum.

  Input:
    speed - the speed to be set

  Result:
    None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    speed = SpeedDown

  Description:
    Move the CIA speed one tick down.

  Input:
    None

  Result:
    speed - The new speed

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    speed = SpeedUp

  Description:
    Move the CIA speed one tick up.

  Input:
    None

  Result:
    speed - The new speed
```

1.101 artechnical

```
  Command:
    author = GetAuthor

  Description:
    Returns the author of the current playing module.  If  there  isn't  an
    author it will return unknown.
```

```
 Input:
   None

 Result:
   author - The name of the author

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   name = GetChannels

 Description:
   Returns the number of  used channels.  This is virtual channels, which
   means that it can vary from 1 to 32 channels.

 Input:
   None

 Result:
   type - the number of used channels.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   num = GetCMod

 Description:
   Returns the current playing module number. 0 if no module is playing.

 Input:
   None

 Result:
   num - the number of the current module

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   name = GetFilename num flag

 Description:
   Returns the module filename. "num" is the module number in  the  module
   list. "flag" indicates if you want path. 0 means no, 1 means yes.

 Input:
   num  - the module number
   flag - path or not (see above)

 Result:
   name - the filename

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   info = GetInfo

 Description:
   Will return a longword where each bit represent the  information  about
   the current player. Below is a table of the bits returned.

   Bit                    Bit

   0 = Rewind              10 = Length
   1 = Forward             11 = Position
   2 = Pause               12 = Max Patterns
   3 = Volume              13 = Show Samples
   4 = Fade                14 = Play Samples
   5 = Tempo               15 = Accompany
   6 = Channels            16 = Force Filter Off
   7 = Modulename          17 = FastMem
   8 = Author              18 = Own Config
   9 = SubSongs            19 = Noteplayer

 Input:
   None

 Result:
   info - a number containing the above bits

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   len = GetLength

 Description:
   Get the length of the current  playing  module.  If  the  length  isn't
   available 0 will be returned.

 Input:
   None

 Result:
   len - the song length

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   max = GetMaxPatterns

 Description:
   Returns the number of patterns in the current playing  module.  If  the
   number of patterns isn't available 0 will be returned.

 Input:
   None

 Result:
   max - number of patterns

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   max = GetMaxSamples
 Description:
   Returns the max number of samples the current player can handle. If the
   player can’t handle any samples, 0 will be returned.

 Input:
   None

 Result:
   max - the max number of samples

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   max = GetMaxTunes

 Description:
   Get the max number of tunes in the  current  playing  module.  It  will
   return 0 if there isn’t a module in memory.

 Input:
   None

 Result:
   max - the number of tunes

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   rate = GetMixingRate

 Description:
   This get the actual mixing rate used by the  used  noteplayer. It  will
   return  0  if  there isn’t a module in memory or if no noteplayer, with
   mixing routines, is in use.

 Input:
   None

 Result:
   rate - The mixing rate

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   name = GetName

 Description:
   Returns the name of  the  current  playing  module.  This  is  not  the
   filename  but the name of the module taken from the module. If it isn’t
   available it will return the filename.

 Input:
   None

 Result:
   name - the module name
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   name = GetNotePlayer

 Description:
   Returns the name of the current noteplayer.

 Input:
   None

 Result:
   type - the noteplayer name

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   type = GetPlayer

 Description:
   Returns the name of the current used player library, eg. Protracker or
   Future Composer.

 Input:
   None

 Result:
   type - the player name

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   mode = GetPlayMode

 Description:
   Get the current playing state. The result is  a  boolean  and  1  means
   VBlank and 0 means CIA.

 Input:
   None

 Result:
   mode - the current playing state

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   pos = GetPosition

 Description:
   Get the current position. If the current player can't get the position,
   -1 will be returned.

 Input:
   None

 Result:
   pos - the current position

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   size = GetSize

 Description:
   Get the filesize (unpacked) in bytes of the current playing module.

 Input:
   None

 Result:
   size - the module size

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   tune = GetTune

 Description:
   Get the current playing tune number.

 Input:
   None

 Result:
   tune - the current tune number

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   success = SetAuthor author

 Description:
   Change the author. It will return an error if no module is in memory.

 Input:
   author - the new author name

 Result:
   success - this is an error code where 0 means ok  and  1  means  error.
             Note that the result is in RC not in RESULT.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

 Command:
   success = SetName name

 Description:
   Change  the  module  name.  It  will return an error if no module is in
   memory.

 Input:
   name - the new module name
```

```
Result:
  success - this is an error code where 0 means ok  and  1  means  error.
            Note that the result is in RC not in RESULT.
```

1.102 arvarious

```
Command:
  Filter State

Description:
  Turn on or off the filter. 1 means on and 0 means off.

Input:
  state - the filter state

Result:
  None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  Quit

Description:
  This will quit APlayer immediately.

Input:
  None

Result:
  None
```

1.103 arvolume

```
Command:
  vol = GetVolume

Description:
  Get  the  current  volume and place the result in "vol". If the current
  player can't change the volume, the result will be 64.

Input:
  None

Result:
  vol - the current volume

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  SetVolume vol
```

```
Description:
  Set the volume to "vol". If the number is out of range the volume would
  be set to either max or min.

Input:
  vol - the new volume

Result:
  None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  vol = VolumeDown

Description:
  Move the volume one tick down. It will return the new volume.

Input:
  None

Result:
  vol - the new volume

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  vol = VolumeUp

Description:
  Move the volume one tick up. It will return the new volume.

Input:
  None

Result:
  vol - the new volume
```

1.104 arwind

```
Command:
  Forward

Description:
  Step one pattern forward. The same as pressing the forward button.

Input:
  None

Result:
  None

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Command:
  Rewind

Description:
  Step one pattern backward. The same as pressing the rewind button.

Input:
  None

Result:
  None
```

1.105 arwindow

```
  Command:
    Iconify

  Description:
    Iconify APlayer with an AppIcon.

  Input:
    None

  Result:
    None

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  Command:
    Popup

  Description:
    Popup the main window.

  Input:
    None

  Result:
    None
```

1.106 index

AREXX

Channels

Eject

Loop

Main Cycle

Modulelist

Pause

Play

Requester

Rewind

Sample

Speed

Tech Info

Various

Volume

Window

A

About

AIFF Config

Allocate Channels

APML©

App-Icon

ARexx Config

ARexx interface

C

Cancel Config

Channels

Checkmarks

CLI Parameters

Commodity Hotkeys

Configuration

Contact

D

Default Config

Disclaimer

Double Buffering

E

Early load

Eject

Error Requesters

Expunge Libraries

Extract pattern

F

Fade At Pause/Next

Fade Module At End

Fade Speed

Favourite Song System (Config)

Favourite Song System (Main)

Featurelist

FilePaths and Patterns

Force Filter Off

Forward

FSS

Future

H

History

How to Install

How to use

I

Iconify

IFF Config

Introduction

J

Jump To Loaded Module

Jumpload

K

Keyboard Control

Keypad

L

Lha Check

Listview Control

Load Config

Load Libraries

Loop

M

Main Cycle Gadget

Make ARexx Port

Module formats

Module list (main)

Module List Editor

Module list

N

Next

P

Pause

Play samples

Play

Players Configuration

Players

Powerpacker.library

Previous

Programmers

Public Screens

Q

QuadraComposer

R

Requirements

Rewind

RIFF Config

S

Sample Info Window

Samples, save

Save Config

Save Windowpositions

Scandir

Screen Selector

Scrollbars

Shuffle

SID Config

SMUS Config

Sound Channels

Speed Reset

Speed Setting

Status Bar

T

Thanks to

ToolTypes

Tunes

U

Unpack.library

Use Config

V

VBlank Interrupt

Volume Reset

Volume Setting

X

Xpk-libraries