0 Welcome !

Thank you for deciding to buy **SoundFX** (or for considering it at this moment). **SoundFX** is an extensive and also quite complex program. Therfore it is really important that you at least glance through the documentation.

Contents

0 Introduction

Below I start with some general overview of **SoundFX**, followed by some legal talk and − very important − the information about registration along with contact data about the author [thats me ;−)].

Contents

[SoundFX]

[SoundFX] [Introduction]

0.1 What is SoundFX ?

SoundFX is an editor for digitized audio data (samples). **SoundFX** is built in a modular way and features a comfortable GUI (Graphic User Interface). With **SoundFX** you can add effects (which are really unique on AMIGA) to your samples and edit them extensively. Think of **SoundFX** as a swiss army knife for sounds!
Here is an overview of its features :

- more than 65 operators with lots of parameters and possibilities for modulation such as :
    - sound synthesis
        - AM−Synthesis (amplitude modulation)
        - CS−Synthesis (composite synthesis = additive and subtractive sound synthesis)
    - FM−Synthesis (frequency modulation)
    - 3D−cube parameter modulation (Mix, Equalize)
    - effects such as Hall, Echo, Delay, Chorus/Phaser, Morph, Pitchshift, Timestretch ...
    - operators like Resample, ZeroPass (FadeIn/FadeOut), Middle, Amplify, Mix, DeNoise, ConvertChannels ...
    - 2D/3D−spectral analysis
    - very good filters and boosters with resonance !!!
    - several types of modulation
        - even volume and pitch tracking
    - more then 250 presets included
- internal signal resolution of 80/16 bit
    - 80 bit floating point during calculations
    - 16 bit in in the sample buffer
- good play routines
    - 8 bit standard player
    - 14 bit cascade player (without additional hardware)
    - 14 bit calibrated cascade player (without additional hardware)
    - AHI−player for sound cards
    - plays samples directly from fast−ram or from hard−disk while using max 16 kByte chip−ram during playback
- conversion of different sound sample formats
    - IFF−8SVX/16SV/AIFF/AIFC/MAUD,RAW,RIFF−WAV,VOC,SND−AU,...
    - with compression support
- works now also with samples bigger than available memory
- works in mono, stereo and quadro !!!
- operations are non−destructive, so the source sample will be neither overwritten nor deleted
- extensive number of cut−functions

- freehand−edit
- flexible screen display
  - number of sample buffers is limited by your system resources only
  - each sample has its own window, with changeable position and size
  - smooth variable zooming (can be
  - X− and Y−zoom !!
  - and rulers with configurable units
- HTML online help
  - by pressing the "HELP"−key in any window
  - asynchronous (the help window could stay opened)
- clipboard support with all 256 entries
- datatype support (loader)
- arexx−port
  - with many procedures and functions (actually about 90)
  - with several examples
  - arexx−scripts can be started directly from **SoundFX**
- system conform GUI
- font− and screen−sensitive
- modular concept, means unlimited
  - operators (65 at this time)
  - loader (19 at this time)
  - player (4 at this time)
  - rexx−macros (several scripts included)
  - saver (15 at this time)
- supports AMIGA−specific functions
  - file information in filenotes
  - generation of projekt−icons
  - applikation−icon

In the unregistered version saving of samples and using the arexx−port is not available !

[SoundFX] [Introduction]

[SoundFX] [Introduction]

0.2 Where does SoundFX run ?

The program runs on all AMIGA computers with AmigaOS 3.0 or greater. I have stopped building the version for plain 68000 system (but could immediately do this again, if there is really someone who need it). As some effects rely heavily on CPU power (or FPU for those who have it:) and the GUI can become quite complex, an accelerator card (with a FPU) is recommended. In addition to this, memory usage can increase greatly with use of 16/32 bit processing. Finally, a graphic−card helps to prevent loosing overview.

SoundFX can be used on MorphOS systems (with 68k emulation) and on Amiga−emulators (Amithlon and UAE).

Ideally you system should look like mine − then it would be unlikely that **SoundFX** will not work ;−). This would then be an A2000 with a 060 based board (64 Mb RAM) and SCSI controller, graphic−board (PicassoIV), sound−card (Prelude & Repulse) and OS3.5.

Further I recommend installing the programs listed below for an enhanced GUI and increased productivity :

- MagicMenu
- ReqAttack
- VisulaPrefs (this is definitely worth registering too)

[SoundFX] [Introduction]

© by Stefan Kost 1993−2004 www.sonicpulse.de

0.3 Copyright

SoundFX
© Copyright 1993−2004 Stefan Kost. All Rights Reserved.

No warranties will be given for full functionality of the software. Furthermore I accept no liability for damage because of misuse. If you found any error in the program, then please contact me with a description of it. I will try to fix it soon as possible.

The program package, except of the key−file, is freely distributable. Its even desired to spread it, as long as the fees are not more than 5.− DM or $3.−. If you want to distribute the program as part of a compilation or series, please contact me and ask for permission.

This demo−package may be relased on following disk−series or CDs without previous request.:

- Aminet CD
- Fred Fish CD
- Saar PD−Serie
- Time PD−Serie
- Amiga−Magazin PD/CD
- AmigaPlus CDs

I strongly recomend you not to use a cracked version, because it might crash very often and possibly damage your hardisk !
If you realy think **SoundFX** is too expensive for you, then better tell me.

---

popupmenu.library
© Copyright ?−2000 Henrik Isaksson, All Rights Reserved.

---

openurl.library
© Copyright ?−2000 Troels Walsted Hansen, All Rights Reserved.

---

stormamiga.lib
© Copyright ?−2001 Matthias Henze, All Rights Reserved.
Try out his HighSpeed Math Libraries !

---

titlebar.image
© Copyright ?−2002 Massimo Tantignone, All Rights Reserved.
Have a look at VisualPrefs !

---

identify.library
© Copyright 1996−2001 Richard Körber, All Rights Reserved.
Thanks to Dan Jedlicka for the example code.

ShowTip
© Copyright 2002 Dan Jedlicka, All Rights Reserved.

[SoundFX] [Introduction]

0.4 Registration

Okay folks here it is. The price tag (yuck!). There's no saving of all your great work in the non−registered version, so you girls and guys mmmmmmmight as well get the chance to have it all.
So what's going to be. Stefan put all this work into this nice program and he's working at it all the time. It really is up to him then you see. Nono. No "why don't ya mail yours to me and I'll send you a sixpack of ..." or something like that , 'cause your letter'll get trashed (or deleted) faster than you can say "burned at the stake".
Stefan has every right to put a price tag onto such a big and complex piece of software. Hey it's not Imagine so there's no "for a $1000 it CAN be yours!!". Play around with it. See what you like, what you don't. Write that down and e−mail it to Stefan (or me,I'll relay everything to him). After you've had a swell time, think of what it would be like to have it all on disk. All your hard work not gone when you exit **SoundFX** or reach down to hit that switch. Think about it ... you can save 16−bit stuff. Convert just about anything into anything, perform dozens of mutilating operations on those innocent samples, twist 'em to your hearts desire. Imagine your friends awe at your samples quality. This is the best in effect software and it won't cost ya $200.
If Stefan doesn't put a price tag in here I guess you should write an e−mail to him and ask. Tell him what you don't like about the prog, while you're at it. And say what you like as well. The guy deserves a patt on the back for it.
So , I'll leave you in the capable hands of Stefan now. He'll give you his address and phone number and way you can send the truck of cash to his place.

AiRoN − first English translation of documentation and tracking (Can you say Protracker ? − I thought you could)

(Stefan continues writing ...) Thanks AiRoN − I think everybody knows now, that it's a good idea to register and here comes the price.

Version	€	US$
Standard	20	26
Aminet 12 (−50%)	10	13
Delfina (−50%)	10	13
CD with recent version	5	6

Of course you could pay more :−).
Please send me your data like full name, address. After that I'll send you your personal keyfile. If you want you'll get a CD with recent program version as well.

With this keyfile all functions in the program will be available. The keyfile contains your personal data. That's why you **must not** copy it. Of course do newer versions of **SoundFX** accept this keyfile too (there will be NO upgrade fees, never ever!).

The payment can be

- cash
- a money transfer to my account
- per credit card via RegNet

0.5 Author

Here you find all data you need to contact me or to register. You can find my contact data as well on my homepage. If I need to move, there you will definitely find the recent address.

postal address
Stefan Kost
Simildenstraße 5
04277 Leipzig
Germany

bank account
1822direkt (Frankfurter Sparkasse)
BLZ: 5005 02 01
KTO: 1251049344
for transfers for foreign countries, please use the following data:
IBAN: DE64 5005 0201 1251 0493 44
BIC (SwiftCode): FRASDEFF

further Communication channels
e−mail : webmaster@sonicpulse.de, st_kost@gmx.de
phone : +49 (0)341 3910484
icq : nickname=ensonic,icq−id=33451292

... and check my webpages :
http://www.sonicpulse.de − my software (download new versions of **SoundFX**)
http://www.eksor.de − my music
http://www.imn.htwk−leipzig.de/~kost − old university−homepage

0.6 The most important chapters

Because you will probably not read all this from the beginning to the very end right now, I have prepared a list with the most important chapters below. To use **SoundFX** effectively I strongly recommend, that you have a quick look at these chapters at least. Otherwise it may happen that you probably never learn about some features.

Contents

1 Usage

The next sections deal with the usage of **SoundFX**. I will try to cover every part of the interface in detail. Please send me feedback, if I didn't succeed, so I can go over those sections again.

Contents

[SoundFX]

[SoundFX] [Usage]

1.1 General

Start **SoundFX** by double−clicking the **SoundFX** icon or by invoking it from the shell. This causes a window to be displayed, which informs you about each phase of the starting procedure.

After loading the **SoundFX** screen will appear. There all action will take place. The screen is a public screen, which means that other applications can open windows there too. The public screen is called "SFX_PubScreen". Pressing the "Help" Key activates the online−help for the active window. At the top screen border, you can see the screen bar:

Beside program name and version number, you find information about current memory usage here too.

On the first start, loading the software takes a bit longer as it creates some indexes for the online−help and database−files for the external modules. On subsequent starts the files are only regenerated if changes to the installation have been made.

When starting **SoundFX** from the shell, you can pass filenames of audio files as arguments, which will be loaded then as well. Further you can enter **SoundFX** as the default tool in icons of sound files. When double−clicking such an icon, **SoundFX** and the sound file will be loaded. If **SoundFX** already run, new files will be added to it.

[SoundFX] [Usage]

[SoundFX] [Usage]

1.2 Menus

Depending on which **SoundFX** window is active, you can access one of the pull−down menus described in the next sub chapters.

Grayed menu entries signal that the menu entry is not available at the moment. This happens e.g. if you have not yet loaded samples or have not marked a range.

Contents

1.2.1 Main menu

This menu is available, when no other dialog window is active.

main menu	sub menu	description
Project	New	opens a dialog for generating an empty sample
	Load	load samples with the current loader (see choosing a loader)
	Save	save a sample with the current saver (see choosing a saver)
	Close	removes selected samples after prompting the user
	Execute	start the current operator (see choosing an operator)
	Execute Rexx	start the current rexx−script (see choosing a rexx−script)
	Play All	play the whole sample
	Play Range	play the selected area
	Stop	stop playing
	Record	opens the recording window (requires AHI)
	Batch Processor	opens the batch processor window
	Info	opens the information window
	MRU (5x)	the 5 samples you have loaded last, can be reloaded herewith these entries are stored in the file "data/MRU.cfg".
	Quit	end the program after prompting the user
Edit	...	similar to the edit toolbar
Range	...	for setting, adapting and reseting ranges
Zoom	...	similar to the zoom toolbar
CleanUp	Current	reorder the active sample−window
	All	reorder all sample windows
	All normal	reorder all sample windows and resize them to standard size
	All zoomed	reorder all sample windows and resize them to small size
Utilities	Swap byte order	repair files saved with wrong byte order
	Swap sign	repair files saved with wrong sign
	Interleave channels	repair files saved with wrong channel
	De−interleave channels	repair files saved with wrong channel
Prefs	GUI	preferences for the GUI
	Sample	preferences for the sample window
	Virtual memory	preferences for virtual memory
	Miscellaneous	miscellaneous preferences
	Use	remembers the current settings as long as the computer is switched on
	Save	save the current settings permanently
	Load last used	load the last used settings
	Load last saved	load the last saved settings
	Reset to defaults	set default settings

Help	...	invoke the online−help with the chosen topic, go to the support page at the internet, write an e−mail to the author or display version info of the software.

[SoundFX] [Usage] [Menus]

[SoundFX] [Usage] [Menus]

1.2.2 Module menu

You find this menu in the settings windows of the modules.

main menu	sub menu	description
Project	Load	load settings
	Save	save current settings
	Start	start the current module
	Reset	reactivate last settings
	Default	reset to initial settings
	Help	open help about the current module
	About	open an information window
	Quit	close the module

[SoundFX] [Usage] [Menus]

[SoundFX] [Usage]

1.3 Toolbars

At the upper border of the **SoundFX**−screen are several tool bars. They offer quick access to the functionality of **SoundFX**>. Many of the offered functions can be accessed via the main menu too. If the mouse−pointer is over a toolbar button, you can see in the statusbar at the lower screen border a short help text related to the button.

Contents

[SoundFX] [Usage]

1.3.1 Loaders

1 2 3 4

button	description
1	name of the active loader−module
2	opens the drop−down list
3	opens the settings−window for the selected loader. you can start the loader from there as well.
4	starts the selected loader

You can read about theses modules in detail in section 2.2

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage] [Toolbars]

1.3.2 Savers

1 2 3 4

Button	Description
1	name of the active saver−module
2	opens the drop−down list
3	opens the settings−window for the selected saver. you can start the saver from there as well.
4	starts the selected saver

You can read about theses modules in detail in section 2.5

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage] [Toolbars]

1.3.3 Operators

1 2 3

button	description
1	name of the active operator−module
2	opens the drop−down list
3	starts the selected operator with the settings−window

You can read about theses modules in detail in section 2.1

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage] [Toolbars]

1.3.4 Rexx−Operators

1 2 3

button	description
1	name of the active rexx−module
2	opens the dropdown−list
3	starts the selected rexx−module

You can read about theses modules in detail in section 2.4

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage] [Toolbars]

1.3.5 Players

1 2 3 4 5 6 7

button	description
1	name of the active player−module
2	opens the drop−down list
3	opens the settings−window for the selected player
4	plays the active sample with loop
5	plays the selected range
6	stops the player
7	opens the recording window (requires AHI)

You can read about the player−modules in detail in section 2.3

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage] [Toolbars]

1.3.6 Buffers

1 2 3 4

button	description
1	name of the active sample
2	opens the dropdown list
3	opens the settings−window for the selected sample
4	hides/shows the selected sample

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage] [Toolbars]

1.3.7 Edit

1 2 3 4 5 6 7

SoundFX comes with plenty of editing functions (probably many more than in other software). Please keep in mind that these are destructive operations. That means you directly edit the sample − there will be no new buffer with the result and you can't take edits back. Better save the sample more often to disk. To select a range for processing, just click the starting point and move the mouse to the ending point while keeping the mouse−button pressed. While selecting, the range will be highlighted and the positions as well as the length can be seen in the status−bar. You can fine−tune the selection with the function from the range−toolbar.

These functions are available (every button is followed by a menu) :

button	description
1	cut − cut selection into copy−buffer
2	copy − copy selection into copy−buffer
3	paste − paste contents of copy−buffer
4	erase − erase selection
5	grab − generate a new buffer containing the selection
6	zero − set selection to zero (absolute silence)
7	overwrite − overwrite with contents of copy−buffer

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage] [Toolbars]

1.3.8 Zoom

1 2 3 4 5

These functions allow you to enlarge or shrink the selected range freely, so that you can work more efficient. To select a range to zoom, just click the starting point and move the mouse to the ending point while keeping the mouse−button pressed. While selecting, the range will be highlighted and the positions as well as the length can be seen in the status−bar. You can fine−tune the selection with the function from the range toolbar.

button	description
1	zoom−mode − zoom into which direction
2	enlarge, if no range is selected then enlarge by 2
3	shrink
4	1:1, means one pixel is one sample
5	show all (zoom out totally)

As these functions are used quite often, **SoundFX** offers the following shortcuts :

	x−axis	y−axis
zoom in	"<"	CTRL+"<"
zoom out	">"	CTRL+">"

1.3.9 Range

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

These functions are for finely adjusting loops, marked ranges and the zoomed area :

button	description
1	range mode : Loop : editing the looped part Mark : editing the highlighted area Zoom : editing the enlarged area Trace : inspect sample values and freehand correction some actions automatically switching the mode : loop : switching loop on or off in the options mark : select a range with the mouse zoom : use hotkeys for zooming or use buttons of the zoom−toolbar
2	lock begin or end (will not be moved on subsequent edits)
3	move begin or end
4	move to the left border
5	move to left fast
6	move to left slowly
7	move to the next left zero−crossing
8	move to the next right zero−crossing

9	move to right slowly
10	move to right fast
11	move to the right border
12	move to the upper border
13	move upwards fast
14	move upwards slowly
15	move to the upper peak
16	move to the lower peak
17	move downwards slowly
18	move downwards fast
19	move to the lower border

The facility for seeking zero−crossings is excellently suitable for generating crackle−free loops. Just set the loop points manually first. Then play the sample. On every retrace you will quite likely hear a crack. Now activate "lock" (2) and click on "<0" (7) to adjust the start and on "0>" (8) to adjust the endpoint until the crack is minimal or even gone.

If you have chosen "trace" and activated a sample−window, you will see the value under the mouse−pointer in fields (8) and (9) of the status bar. The sample−value will be show in field (10) and can even be changed there.

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage] [Toolbars]

1.3.10 Window Mode

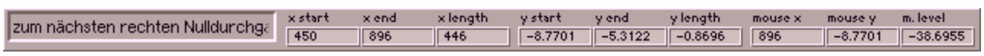
1

button	description
1	switch between multiple sample windows on screen or one big window

[SoundFX] [Usage] [Toolbars]

[SoundFX] [Usage]

1.4 Statusbar



1 2 3 4 5 6 7 8 9 10

button	description
1	quick help − just move the mouse over the toolbars ...
2	start of X−range
3	end of X−range
4	length of X−range

5	start of Y−range
6	end of Y−range
7	length of Y−range
8	X−value under mouse−pointer
9	Y−value under mouse−pointer
10	Y−value in the sample at the mouse−pointer

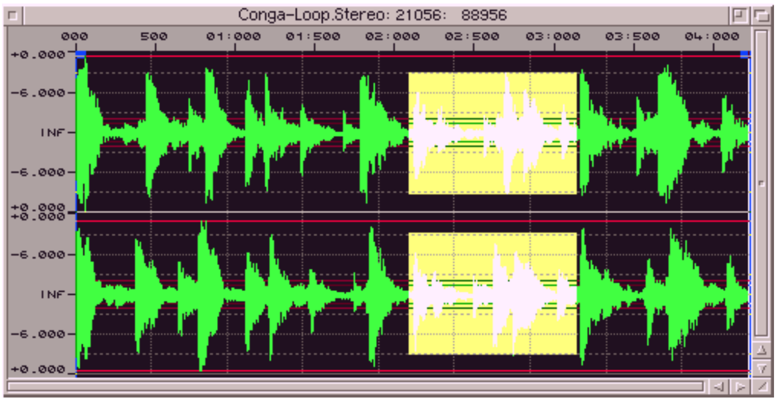
1.5 Windows

Many menu−entries and toolbar−buttons are followed by dialog−windows.

You will probably note, that none of the **SoundFX** windows have cancel−buttons. I have left them out, as you can achieve the same results but closing the window with the close−gadget in the upper left corner of the window−frame or by pressing the ESC−key. Similar to that pressing the ENTER or RETURN key you select the okay−button (marked in bold) of that window.

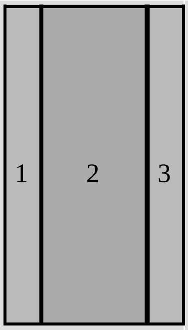
Contents

1.5.1 sample window

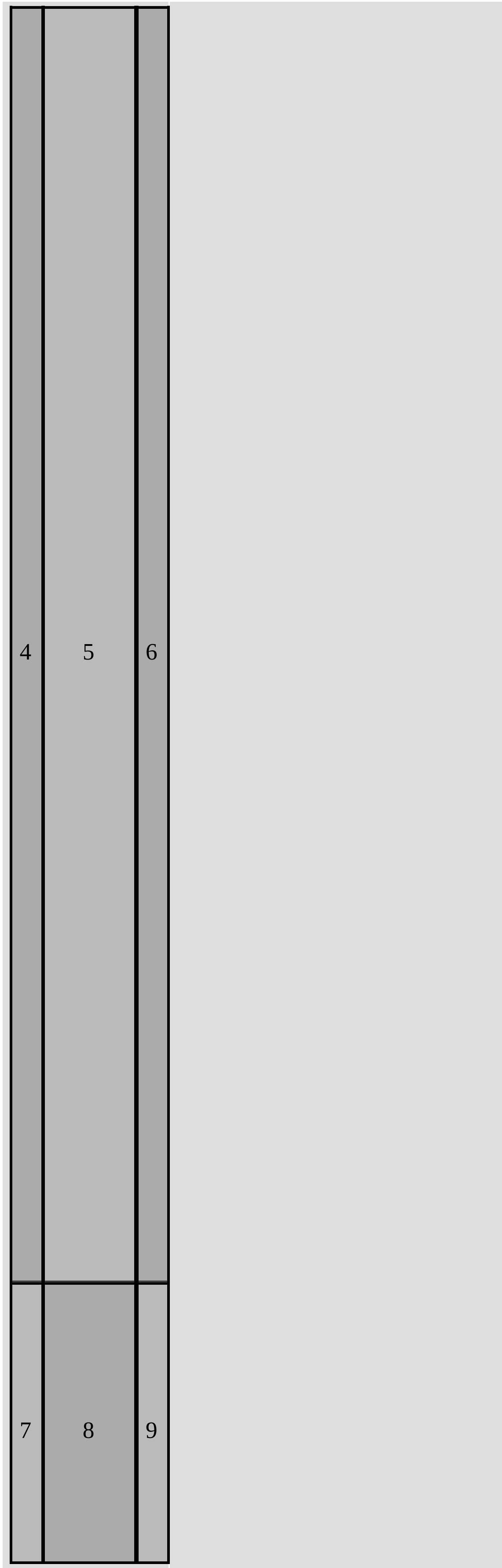


the window :
When a sample has been loaded or generated, it is then displayed within its own window. Size and position can be changed via the windows gadgets. Several lines are drawn to help reading positions and levels of the sample. Additional lines can be drawn to display the maximum, average and real (acoustical) amplitude.
If 'Loop' is activated and start and repeat lengths are set, vertical lines with boxes attached to the top will visualize the looping part. If some range is selected (marked), this is shown by an highlighted filled box.
In the window title bar **SoundFX** displays the samples name, playback rate and length. While playing a sample you see the play−position there.

actions inside the window :
When moving the mouse around the mouse pointer will change its shape to indicate what action can be performed.
The loop lines can be moved by clicking and holding the left mouse−button onto the box and moving the mouse.
Clicking down the left mouse−button inside the sample window but outside of the loop boxes or a previously marked area will start a new marking operation. When clicking inside a mark (not near the borders), it can be moved around, while holding the left mouse−button. When clicking inside a mark at the borders, the range can be modified into that direction. Here is a 'picture' to make it more clearly (anyway the mouse pointer shape should clearly show the available action) :



1 change left and upper border
2 change upper border
3 change right and upper border
4 change left border
5 move around
6 change right border
7 change left and lower border
8 change lower border
9 change right and lower border



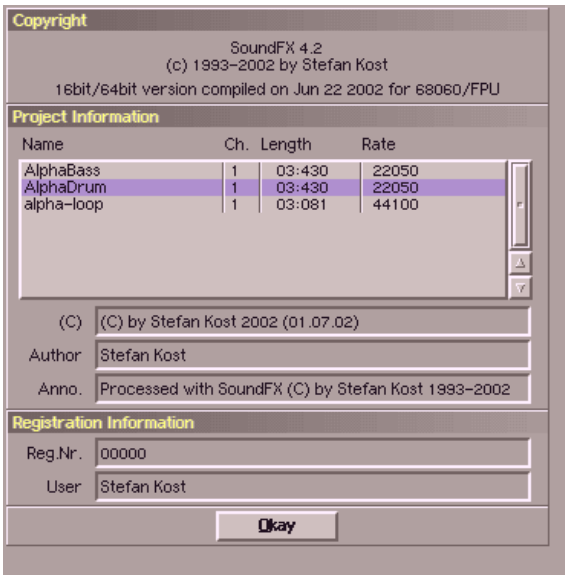
This area (or range) can be magnified, cut or copied. If an area was magnified, moving the slider at the bottom or right of the window, will move the display through the sample data. This area will be continuously updated while sliding. While modifying loops, marking ranges and zooming areas **SoundFX** displays information about start, end and length in the status bar.

If you have zoomed you view more than 1:1 and selected "Trace" in the range−toolbar, it is possible to draw directly into the sample data while left mouse−button is pressed. With this function you can manually remove errors (cracks). The sample display will be refreshed when you release the mouse−button.

[SoundFX] [Usage] [Windows]

[SoundFX] [Usage] [Windows]

1.5.2 information window



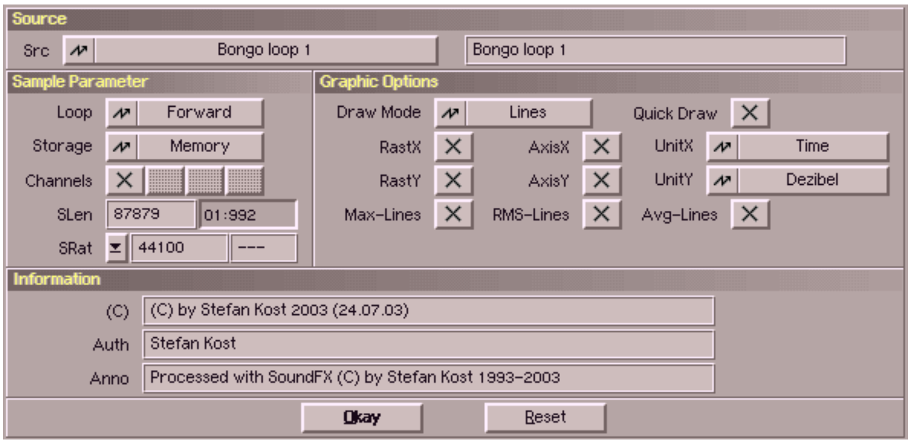
Information displays, as its name suggests, useful information about the program like :

Range	Description
program name	If this not reads "**SoundFX**", you are using the wrong software ;−)
version number	Please always include this, when contacting me with a problem
copyright & author	...
list of samples	a list of loaded samples. After selection one entry, additional information will appear in the fields below.
registration information	Your registration number and name (if there is a name, then it is hopefully yours !!!).

[SoundFX] [Usage] [Windows]

[SoundFX] [Usage] [Windows]

1.5.3 sample options window



This windows allows to modify options for the sample windows, which are described in detail below :

gadget	description
Draw Mode	With this cycle button you can choose how the sample should be drawn. Available are the following modes : • 1. Lines • 2. Dots • 3. DotAbs • 4. Filled • 5. FilledAbs • 6. FilledHQ (very exact, but slow)
Loop	For switching the loop mode.
Storage	Herewith you can specify, if a sample should be kept in memory or swapped to hard−disk. **SoundFX** normally decides this automatically. This can come handy, when you don't need a sample for a while and want to free that memory for other samples.
Channel	Determines which channel should be displayed in the window (makes only sense with stereo/quadro samples). Each button corresponds to one channel. Following operations will be limited to these channels only.
Raster X/Y	With these check boxes you can disable the drawing of the raster. This speeds up the drawing.
Axis X/Y	And with those you can disable the axis. This enlarges the drawing space for the waveform.
Unit X/Y	These gadgets are for choosing the unit to be used for each axis. These units will also be used by the status bar.
Max−Lines	These lines show the maximum volume of a sample.
RMS−Lines	The rms−lines show you the real acoustic volume of a sample. Calculating this and also the next may take a while (for long samples).
Avg−Lines	And these lines show the average volume.
Quick Draw	If this is selected, the drawing of raster and Max−,RMS−, Avg−lines will be left out during scrolling.
SLen	Here you can change the length of the sample. This is necessary should you want to for example do a 'Echo' effect on a short sample that is supposed to be longer than the sample itself. Simply enter desired length and **SoundFX** will add the empty section you requested, giving you the room you need for the effects stuff. Additionally **SoundFX** shows you the length in the current unit.
SRat	To change the playback rate you can choose one of these three options pop−up button: Will put you into the period choice window. The results are then entered into the gadgets at the right. rate gadget: Change the playback rate 'directly' by entering the rate in Hz. Higher rates give you higher pitches. 8000 to 48000 would constitute normal playback rates. The gadget next to ours will display the note (as seen in trackers) after entering the playback rate. If there is no note

		equivalent to the period a '−−−' string is shown.
	note gadget	Here you can enter a desired note, which must comply with this : • 1. char : key="C,D,E,F,G,A,H" • 2. char : white keys="−", black keys="#" • 3. char : octave="0,1,2,3,4,5,6,7" examples : "C#3", "E−0", "H−7" **SoundFX** will display the period (ProTracker) for the chosen note in the corresponding gadget.
	If you change the rate during the active−buffer plays, you will hear the changes immediately.	

Further you can change the strings, which are saved with the sample in some file formats.

A click at "Okay" closes the window and one at "Reset" restores the settings to the choices made in the sample window preferences (please note that the parameters "SLen" and "SRat" are not restored).

[SoundFX] [Usage] [Windows]

1.5.4 period choice window

In this window you could select the sampling rate. You can do it in the following ways :

method	description
mouse	Simply click on the wished note in the keyboard. Rate and referring note will be shown in the fields below.

keyboard	Choose with F1−F5 the octave and select the pitch with the following keys :
	
	`s d g h j`
	`y x c v b n m`

Below the keyboard−image you see the rate, note and frequency. With the cycle−gadget below you can choose between the often used rates.

sampling rate	typical application
8000 Hz	Sound boards (typical for SND−AU samples)
11025 Hz	Sound boards (typical for old samples)
22050 Hz	Sound boards (typical frequency oft most samples)
28867 Hz	maximum playback rate of the Paula−chip in normal mode
32000 Hz	Consumer DATs and samplers
44100 Hz	CD−Player
48000 Hz	DAT−Recorder/Player
57734 Hz	maximum playback rate of the Paula−chip in productivity mode

96000 Hz	high quality audio processing

With the cycle gadget PlayMode, you can choose, if you would like to listen to the sample while choosing the playback rate. If you have chosen PlayMode=PlayAll and click onto the keyboard panel, you will immediately hear the sound in the respective tune. Of this only aplies to selecting the playback−rate of an already existing sound (when choosing the playback rate in operators like Noise, then there has not yet been calculated anything).

After clicking onto Okay the values will be accepted.

[SoundFX] [Usage] [Windows]

[SoundFX] [Usage] [Windows]

1.5.5 window function window

You can select a windowfunction and eventually adjust a parameter in this window. The chosen function will be displayed graphically. The upper graph shows the course of the windowing function in the time−domain and the lower graph shows its effect in the frequency−domain. This way one can see, that some functions filter better in the stop−band, but make the slope less steep. This window usually get opened from an operator (choice of window function).

The choice of a windowing function is always a compromise. Here an example for a FIR−filter :

Window	Description
Rectangle	+ good slope − bad gain
Hamming	− bad slope + good gain

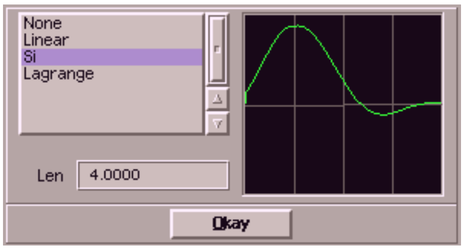
Multiple application of a filter let both characteristics become better.

[SoundFX] [Usage] [Windows]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Usage] [Windows]

1.5.6 interpolation type window



You can select an interpolationtype and eventually adjust a parameter in this window. The effect of the chosen type will be displayed graphically. When digitizing a sound, the hardware takes probes after very short intervals. This results in the digitized wave−form. But some effects need values between these probe−points. Here too **SoundFX** is flexible and offers a rich choice:

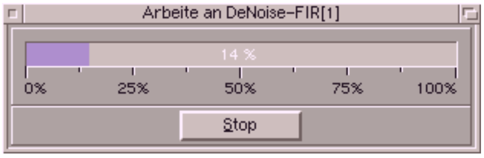
choice	description
None	no interpolation (the nearest value will be taken)
Lin	linear interpolation
Si	curved interpolation over points
Lagrange	curved interpolation over points

For the last two methods it is necessary to specify the size of the interpolation range, which is how many surrounding values should be taken into account to calculate an inbetween value. Do not make this too big (bigger that 10).

[SoundFX] [Usage] [Windows]

[SoundFX] [Usage] [Windows]

1.5.7 status window



In this window the progress of an operation will be shown. Therefore **SoundFX** utilizes a growing status−bar with a percentage−display. Additional information is given in the title of the window.

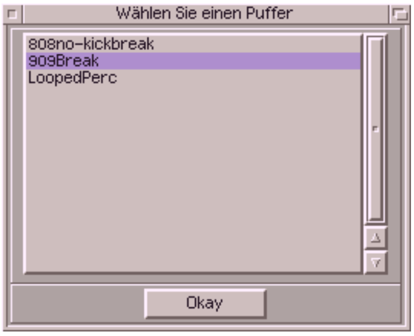
The calculation can be stopped with one click at "Stop", pressing the keys "S","s","ESC" or a click at the "Close"−gadget of the window.

[SoundFX] [Usage] [Windows]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Usage] [Windows]

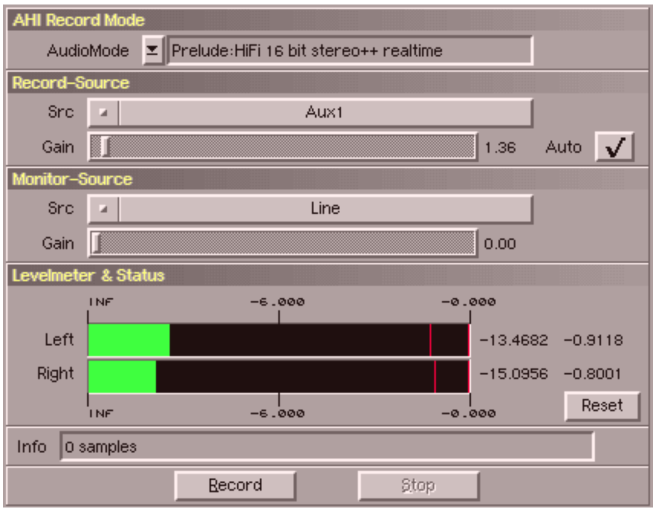
1.5.8 source selection window



This windows is for choosing an entry from a list. It will be opened after clicking on the pop−up symbol. The chosen entry will be displayed in the field beside the pop−up button. You can choose by either double−clicking an entry or by pressing "Okay".

[SoundFX] [Usage] [Windows]

1.5.9 record window



SoundFX naturally offers functionality for recording own sounds from external sources (e.g. microphone). **SoundFX** uses AHI for recording. If you want to record audio directly from a CD then please have a look at the CDDA−Loader. This window offers the following functions :

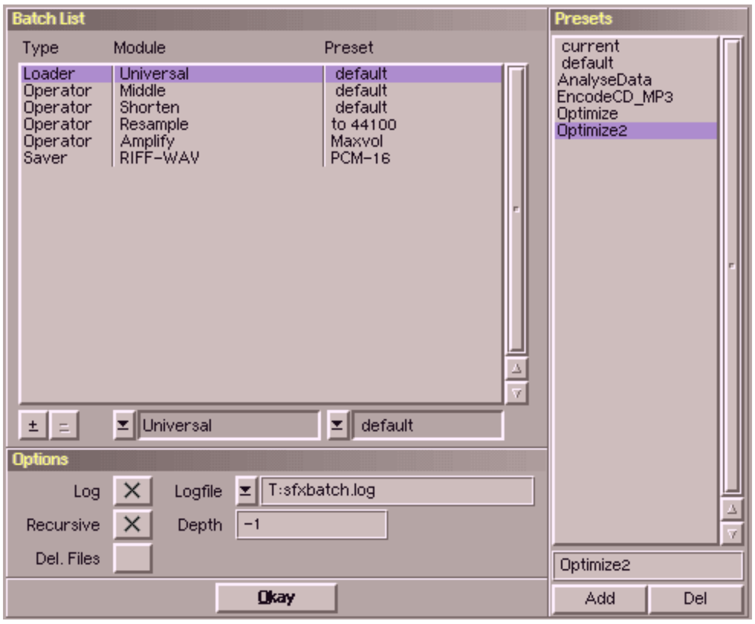
gadget	description
AHI Record Mode	choose an audio−mode for recording.
Record Source	a list of available recording sources.
Record Gain	for adjusting the record gain.
Record Auto	This is a special feature of **SoundFX**. Move the gain to full right and activate 'Auto'. Now **SoundFX** will lower the gain continously, until it does not clip anymore.
Monitor Source	a list of monitor−outputs .
Monitor Gain	for adjusting the monitor volume.
Level Meter	These level−meters show the volume of the input signal. The red bars mark the maximum value. The values to the right of the meters show the current input and maximum.
Status	Shows how much data already has been recorded.
Reset	for resetting the maximum−display.

A click on "Record" obviously starts recording, where a click on "Stop" ends it. When **SoundFX** records audio the level−meters are inactive to save processing power.

Please note, that AHI currently always records in stereo 16 bit . Future version might record in mono as well. For **SoundFX** there is currently no easy workaround available, other than using the convert−channels operator afterwards. Another problem is, that some of you may not be able to use the gain−sliders. The reason for it is that the recording−hardware and/or the AHI−driver does not support this.

[SoundFX] [Usage] [Windows]

1.5.10 batch processor window



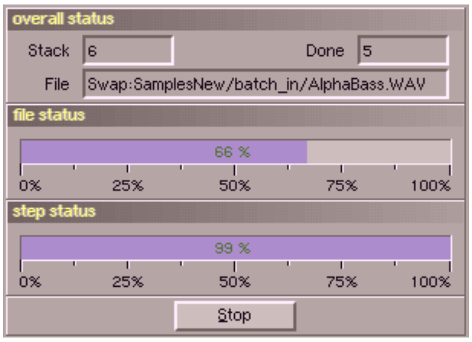
The batch processor allows you to apply a couple of operations (loading, processing, saving) to a whole directory of samples. Therwith you can run a set of processes onto many and/or long files automatically.
Have a look at the presets for examples of operation.

Range	Description
Batch List	This list always consists of one loader and one saver. Inbetween you can add as many operators as you like. Furthermore you can assign a preset to each operation.
Options	Here you can choose, if you want **SoundFX** to log the execution to a file and specify to which file the log should go to. Additionally you can ask **SoundFX** to recursively descend directories. A depth of "−1" means "unlimited depth". This will cause all files to be processed. Finally you can choose that **SoundFX** deletes the source files after processing. This saves space on you harddisk, but be sure to have the files backed up somewhere else.
Presets	Like the preset−selection in the operator−windows, you can save the setting made at the left side and quickly recall them.

[SoundFX] [Usage] [Windows]

[SoundFX] [Usage] [Windows]

1.5.11 batch processor status window



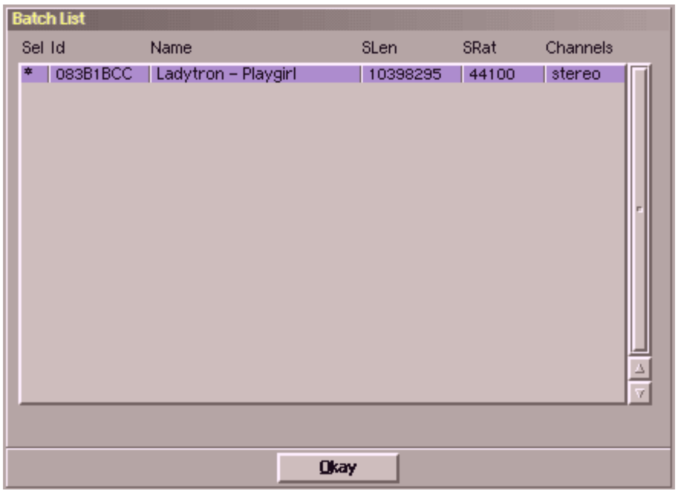
In this window the progress of the batch processing will be shown. This happens in three areas. The upper one gives a main overview. The field "Stack" tells how many files are in the queue. This number may raise during the operation, if further subdirectories are found. The filed "Done" counts the samples which have been processed and the field "File" informs about the current sample. The two status bars below show the progress for the current file and for the current operation.

The calculation can be stopped with one click at "Stop", pressing the keys "S","s","ESC" or a click at the "Close"−gadget of the window.

[SoundFX] [Usage] [Windows]

[SoundFX] [Usage] [Windows]

1.5.12 recovery window



...

[SoundFX] [Usage] [Windows]

[SoundFX] [Usage]

1.6 Settings

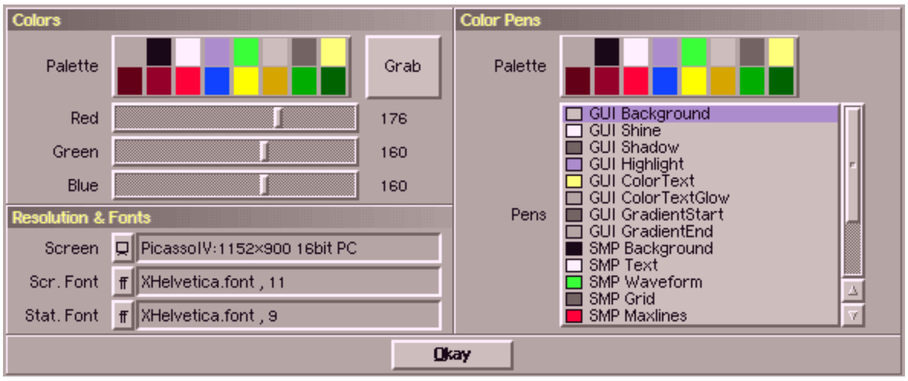
Many properties of **SoundFX** can be customized to your personal preferences in the windows described next.

These settings are stored temporarilly in ENV:sfx.cfg and permanently in ENVARC:sfx.cfg

Contents

[SoundFX] [Usage]

1.6.1 Preferences for the GUI



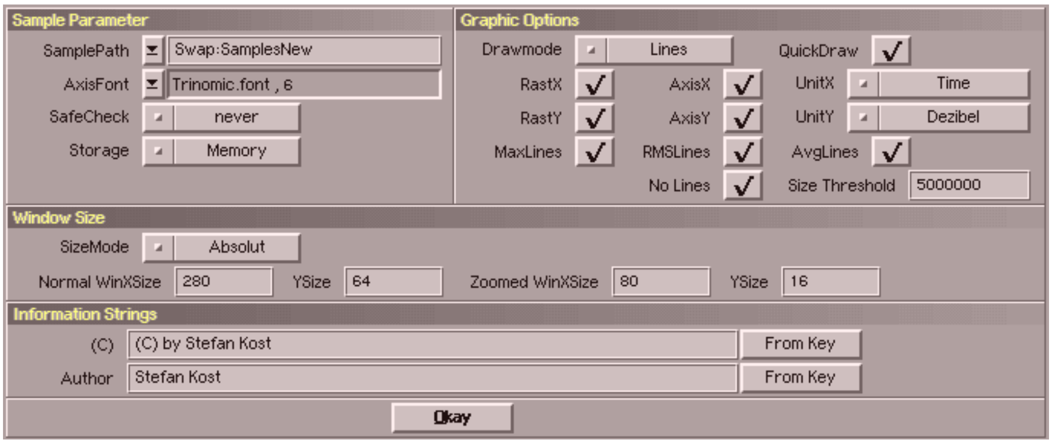
In this window you can change various parameters related to the **SoundFX**−GUI. Below comes a description of the various buttons and functions :

button	description
Palette (left)	Choose a color in the palette which you want to change.
Red,Green,Blue	Change the individual color components for the chosen color.
Screen	In the following window you can choose a screen mode for your **SoundFX** screen (displays only useful modes). Please note, that when choosing highcolor (15/16 bit) or truecolor (24 bit) graphic modes the presentation of the samples can look a bit different (marked ranges and loops).
Scr. Font	Here you can choose a font for the layout. Now non−proportional fonts are available too. But they might sometimes lead to too wide windows and gadgets. The default font (Trinomic.font) is only 6−points high and is necessary if you want to use **SoundFX** on a Hires−NoLace−Screen (640x256). On a resolution of 1024x768 I use XHelvetica with size 11. Do only use larger fonts, if you have chosen a higher screen−resolution
Stat. Font	This font gets used for the status bar fields. I recommend using a fairly small font like e.g. Tinomic in size 6 or XHelvetica in size 9.
Palette (right)	Here you can choose a color which you want to assign to a pen.
Pens	Choose a pen which you want to change.

[SoundFX] [Usage] [Settings]

[SoundFX] [Usage] [Settings]

1.6.2 preferences for the sample windows



In this window you can change various parameters regarding to sample projects. Here's a description of the various buttons and functions :

button	description
Loader/Saver Path	These are the default paths for the file−requests for loading and saving samples.
Axis Font	This font will be used for the rulers inside the sample window.
Safe Check	Here you can choose, how the program should prevents you from discarding unsaved samples : • never : request appears never • if unsaved : request appears only if the sample has not been saved yet • always : request appears ever
Storage	Herewith you can specify, if a sample should be kept in memory or swapped to hard−disk. **SoundFX** normally decides this automatically.
Draw Mode	This cycle−gadget lets you choose a drawing style for the sample waveform. This is you choice : • 1. Lines • 2. Dots • 3. DotAbs • 4. Filled • 5. FilledAbs • 6. FilledHQ (very exact, but slow)
Quick Draw	If this is selected, the drawing of raster and Max−, RMS− and AvgLines will be switched off during scrolling.
Raster X/Y	With these check boxes you could disable the drawing of the raster.
Axis X/Y	And with those you can disable the axis. This enlarges the drawing space for the waveform.
Unit X/Y	These gadgets are for choosing the unit to be used for each axis. This unit will also being used by the status−bar.
Max−Lines	You could disable the calculation of the max. amplification lines. This speeds up the drawing, especially of longer samples.
RMS−Lines	These lines are showing you the real acoustic volume of a sample. Calculating this and also the next may take a while (for long samples.
Avg−Lines	These lines are showing you the real acoustic volume of a sample. Calculating this and also the next may take a while (for long samples).
No Lines	Should **SoundFX** leave out Max−, RMS− and AvgLines for long samples?
Size Threshold	Tell **SoundFX** here what you regard as long samples (number of sample−values)
Window Size	Finally you can choose the default sizes of sample windows here. This can be done by entering absolute values (in pixel) or entering relative values (which are in per thousand of

	the screensize)
Info Strings	Here you can change the comments, which are saved along with the samples.

[SoundFX] [Usage] [Settings]

[SoundFX] [Usage] [Settings]

1.6.3 preferences for virtual memory

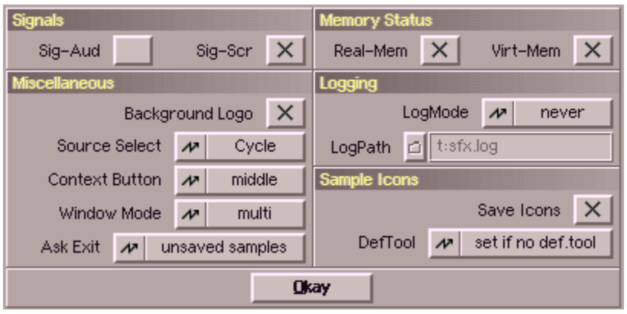
In this window you can change some settings related to virtual memory. Here's a description of the various buttons and functions :

button	description
Enable	Should **SoundFX** use virtual memory at all.
Swap Path	Enter the default path samples are swapped to or choose the path by clicking on the pop−up. **SoundFX** will use as much space there, as it needs.
Blk Size	Size of the i/o−buffer in bytes **SoundFX** will use for drive access. This has nothing to do with your hard−disk's block−size.

[SoundFX] [Usage] [Settings]

[SoundFX] [Usage] [Settings]

1.6.4 miscellaneous preferences



In this window you can change some more settings. Here's a description of the various buttons and functions :

button	description
Sig−Audio	If activated, a signal−sound indicates that calculations are complete
Sig−Screen	If activated, will make to **SoundFX**'s screen pop to the front when a calculation has been finished.
Real−Mem	

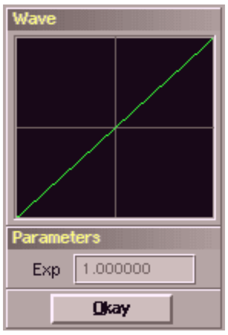
	Should the free memory and the largest available memory block be displayed in the title−bar?
Virt−Mem	Should the free virtual memory (space on your hard−disk in the swap directory) be displayed in the title−bar?
Background Logo	If checked, a **SoundFX** logo will appear in the screen background
Source Select	Which way do you want to select source−samples (e.g. in operator−windows)?
Context Button	Which mouse−button **SoundFX** should use for popup−menus
Window Mode	The window−mode **SoundFX** uses initialy
Ask Exit	How **SoundFX** should behave on exit
Logging	Specify what **SoundFX** should log and choose the path of the logfile.
Save Icons	Should **SoundFX** create icons when saving samples?
DefTool	The default tool is the application that gets started, when one double−clicks a file icon. **SoundFX** can keep the default tool of the icon, enter **SoundFX** if no entry exists or always enter **SoundFX**.

1.7 Modulatorwindow

These windows get activated by operators. Their purpose is to control how a effect−parameter gets modulated. I describe their functionality here, as they appear in nearly all effect−operators.

Contents

1.7.1 Curve

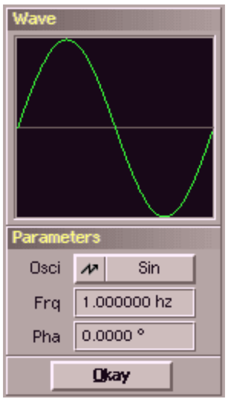


This modulator generates a bended course. The bend can be adjusted with the parameter "exponent" and will be displayed graphically or can be changed with the mouse by dragging it to a desired shape. Below some examples :

variation	description
Linear (exp=1.0)	Runs values from 0.0 at sample start to 1.0 at the end of the sample in the linear way
SpeedUp (exp>1.0)	Similar to the above but the values run at "slow" rate and "faster" to the end
SlowDown (exp	Opposite of "SpeedUp"

[SoundFX] [Usage] [Modulatorwindow]

1.7.2 Cycle



This modulator generates an oscillation. You can choose it's waveform, phase and frequency. The latter could be adjusted in different ways :

variation	description
hz	frequency in hz : 1.5 hz
time	duration of one period in time units or samples : 5 ms
repeats	number of periods (cycles) : 4 rpts

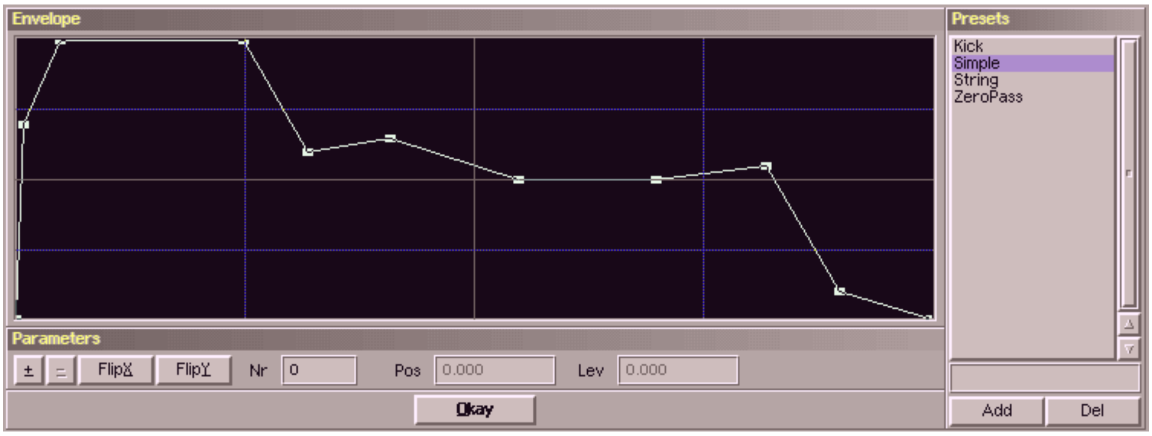
The waveforms Rnd and SRnd produce random peaks, where for SRnd the peaks will be smoothed as well. The parameter frequency determines the number of random values per second (or how long one random value will be hold) and the parameter phase is not used for these both waveforms.

[SoundFX] [Usage] [Modulatorwindow]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Usage] [Modulatorwindow]

1.7.3 Vector

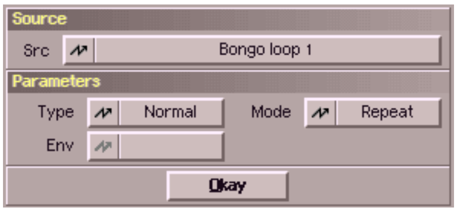


This modulator generates an envelope with a maximum of 20 segments. With "+" and "−" you can add and remove points. With "FlipX" and "FlipY" you can mirror the envelope. In "Nr" you can directly choose a point and position it in the next two fields. Of course, you can use the mouse as well, to move the points.
This modulator supports presets. Therewith you can store generated envelopes and recall them again (they are available in all operators then).

[SoundFX] [Usage] [Modulatorwindow]

[SoundFX] [Usage] [Modulatorwindow]

1.7.4 User defined



This modulator allows using a sample project as modulation source. Below a list of available control types :

variation	description
Normal	If the amplitude of the modulation buffer has reached its negative maximum then this returns the value 0.0 and at the positive maximum 1.0.
Abs	Pretty much the same as 'Normal' with one difference. Sample data of the value 0 (flat line:) gives you values of 0.0 for the modulation curve, maximum negative or positive amplitude of the sample a 1.0.
AmpEnv	This shape gives you the volume envelope of the modulating sample (imagine you stretch a rubber band around the sample)
FrqEnv	This shape returns the pitch envelope of the modulating sample.

Eventually there are different algorithms available for AmpEnv and FrqEnv . These can then be chosen with the cycle−gadget labeled "Env".

The sample buffer you want to use for the modulation curve can be of different length than your to−be−modulated sample. How to handle this is described her :

variation	description

Single	If the sample is shorter, the rest will be filled with silence.
Repeat	If the sample is shorter, it will be repeated for as many times as needed.
Stretch	The sample will be stretched/shrinked to fit exactly.

2 Modules

SoundFX is highly modularized. That means that e.g. all effects are separate modules (plug−ins), which will be loaded only if you are going to use them.

Usually **SoundFX** detects automaticall, that new plug−ins have been installed or removed. If this should fail somehow (e.g. because the clock of you computer is/was wrong), you can force an update by deleting all files ending on ".db" in the subdirectory "data". In a shell you would change into the directory where **SoundFX** is installed and use the following command : "delete data/#?.db".

Nearly every modul has its own settings. These are described along with the module. All these window share the same menuitems.
You can adjust the standard−settings of each modul by saving your settings as "default.cfg".

Contents

2.1 Operators

An operator is a module which processes or generates samples. There are 3 different kinds of operators :

variation	description
effects	process one or many source samples to one or many results.
generators	generate new sounds (synthesizer), do not rely on a source sample.
analyzers	analyze samples (you would never have guessed that ;−)), do not generate any new samples

Most operators are built in a similar fashion. I'd therefore like to explain some things you'll encounter in most operators here and leave it out in the pages about the operators in special.

All parameters you change are held in memory as long as the computer runs, so when you want to use the operator (effect) again (even if you left the program inbetween) you'll get the parameters as you left them. Should the buffer you have used be closed, **SoundFX** changes these settings as those buffers are not existing anymore.

Contents

[SoundFX] [Modules] [Operators]

2.1.1 Source selection

These controls are for choosing a source to operate on. The cycle gadget right to the source allows you to choose the range which should be processed. **SoundFX** automatically suggests the probably most desired mode, e.g. if you have marked a range, then range is preselected. The following variants are possible:

choice	description
All	the whole sample will be processed
Window	only the currently visible part (zoomed) will be processed
Range	only the marked range will be processed

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

2.1.2 Modulator

This area is for adjusting modulatable parameter in **SoundFX**. In the first row you set start and end values. The '<−>' button lets you swap both values.
Now a few words to the parameter themselfs. SIn ce version 3.4 you can use real units in **SoundFX**. E.g. you want to use Amplify to make something sound twice as loud, then you can use the following parameter variations:

example	description
2.0	factor
200 %	absolute, per cent
2000 %%	absolute, per thousand
+ 100 %	relative, per cent
+ 1000 %%	relative, per thousand
+ 6 db	relative, decibel

As you can see − there are lot of possibilities. Below the units currently known to **SoundFX** (contact me if you need more) :

group	description	
amplitude	factor	value
	absolute, per cent	value %
	absolute, per thousand	value %%

	relative, per cent	+/− value %
	relative, per thousand	+/− value %%
	relative, decibel	+/− value db
	absolute, level	value lv
relative frequency	factor	value
	absolute, per cent	value %
	absolute, per thousand	value %%
	relative, per cent	+/− value %
	relative, per thousand	+/− value %%
	relative, semitones	+/− value st
	relative, cents	+/− value ct
	relative, semitones & cents	+/− value:value st:ct
absolute frequency	herz	value hz
	tone	note −/# oktave (e.q. C−3, E#2)
relative time	factor	time
	absolute, per cent	value %
	absolute, per thousand	value %%
	repeats	value rpts
absolute time	hour	value h
	minute	value m
	second	value s
	millisecond	value ms
	second & millisecond	value:value s:ms
	minute & second	value:value m:s
	hour & minute & second	value:value:value h:m:s
	... I think you've got the idea	
	samples	value sv
	movie frames (24 fps)	value mf
	PAL−video frames (25 fps)	value pf
	NTSC−video frames (30 fps)	value nf
proportion	factor	value
	absolute, per cent	value %
	absolute, per thousand	value %%
count	absolute	value
	relative	+/− value
phase/angle	factor	value
	absolute, per cent	value %
	absolute, per thousand	value %%
	degree	value °
	minutes	value '
	seconds	value "
	minutes & seconds	value:value ':"
	... and so on	
	radian	value rad
	english degree	value grd

Not all of these units could be used for all parameter and otherwise sometimes you can use a unit which is unusual for that parameter. The latter case is mentioned in the the apparent description of the operator. The second row : When coding **SoundFX** I wanted to make it as variable (flexible) as possible. The user should be able to access and edit all the parameters in a way either as simple or complex as he/she desires. This led to the development of the 'Blend Shapes'. These are curves (or graphs) that modulate a parameter. A 'Blend Shape' always returns values ranging 0.0 − 1.0. This way it can vary a parameter from its start to it end value. The start value gets used at modulation=0.0 and the end value at modulation=1.0. The following variations are implemented :

variant	description
none	This shape returns in every case a value of 0.0 (if you dont want to modulate something). If you use this enter the value in the first field − the second will be ignored.
curve	bended course
cycle	oscillation
vector	envelope
user	user defined

Examples say more than thousand words. Here are a few for the Amplify−operator :

example	description
1	Youd like to amplify the volume of the sample by 5%. Par.0 : 105 % (100%+5%) Par.1 : doesn't matter Mode : None
2	You want to amplify the sample to 10% at the start and lower to 60% in the end AND the volume change should accelerate to the end. Par.0 : 110 % (100%+10%) Par.1 : 60 Mode : Curve, Exp="2.0"
3	You'd like to produce a tremolo effect (cyclic change of volume − "Helicopter" effect). Par.0 : 120 % Par.1 : 80 % Mode : Cycle, Sin, Frequency, Frq="1 Hz"

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

2.1.3 Interpolator

Effects that need to access samples between two samplevalues, need to use an interpolator for it. After a click onto the popup−symbol appears the interpolation type window where you can choose one.
The textbox right to the popup−symbol shows a shortened version of the active settings.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

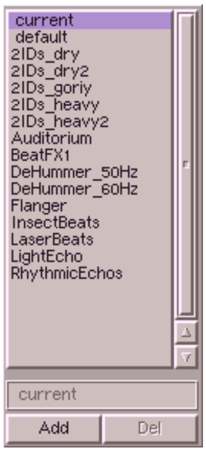
2.1.4 Window funtion selection

FX which are using digital filters or utilizing the fast Fourier tansformation (FFT), need a window−function. After a click onto the popup−symbol appears the window functions window where you can choose one.
The textbox right to the popup−symbol shows a shortened version of the active settings.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

2.1.5 Preset selection



At the right border of nearly all operators you can see a group of buttons helping you to manage you presets comfortably. A preset is a set of parameters, which you can save for later reuse under a expressive name.
An already existing preset can be activated by performing a single click onto that list item. This causes the preset to be loaded immediately. A double click starts the calculation. The preset name can be changed by entering a new name into the input field below the list.
The 'Add' button saves the current entered value under a new name.
The 'Del' button removes the current selected preset.
If you save a preset under the name 'default.cfg', then these values will be taken as initial settings.

If you have mode own presets, which are useful for others too, the please mail them to me.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

2.1.6 List of operators

The following operators are currently available:

Contents

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Amplify

Changes the volume of a sample

Parameter

Amplification (P1)	This value controls the amount of amplification. The volume can be raised and/or lowered.
MaxVol	After a clicking this button, the current source will be scanned and the maximum amplification without clipping will be calculated. the result will be entered into Par0 and modulation will be set to "None".
Wrap	Choose how to handle clipping. The modi below are available : • NoClip : don't test for overdriven values; will produce distorted sounds when raising the volume beyond the maximum • Clip : overdriven values are clipped • Wrap1 : overdriven values are pushed into the opposite side until they don't clip anymore. • Wrap2 : overdriven values are overturned (folded) until they don't clip anymore. Just give it a try. Take a long sinewave and slowly overdrive it.

Notes

Percussion sounds (bassdrums,snare,drums,...) can be lifted a bit (ca. 120 %). This'll produce the typical overdrive effect, by clipping sample data (vertically).
The amount of amplification without hitting the ceiling hard, can be estimated taking a look at the min−and maxlines in the samplewindow.
This operator can also be used for amplitude and ring−modulation, creating further possiblities for sound synthesis. For example, make one sine wave with normal period and another with double. Let the one sine be the source and the other sine be the modulation waveform with User/ Normal set. Set Par0 to 0.0 and Par1 to 1.0. Generate the new sample and take a close look (with zoom maybe?:) at the result. What you have done is called ring−modulation. When choosing the modulation range to be from −1.0 to 1.0 then you will get amplitude modulation.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

AmplifySplit

Allows independet adjustment of volume for upper and lower parts of a sample. Replaces Clap and Clear operators from older versions of **SoundFX**.

Parameter

Upper Amplification (P1)	This value controls the amount of amplification for the upper sample−half.
Lower Amplification (P2)	This value controls the amount of amplification for the lower sample−half.
MaxVolUpper	After a clicking this button, the current source will be scanned and the maximum amplification without clipping the upper values will be calculated.
MaxVolLower	After a clicking this button, the current source will be scanned and the maximum amplification without clipping the lower values will be calculated.

Notes

see Amplify operator

Analyse−Data

Generates histograms of amplitude and amplitude−deltas, as well as a number of statistics of a sample

Parameter

none	

Notes

Once the calculations are complete, a new window is opened, containing the graphs and numbers. With the "channel" cycle−button you can choose for which channel you would like to see the graphs. Close the window by clicking in it's close−gadget.
The data shown helps you in the mastering process to e.g. align the volume of different tracks. If the operator has been invoked from ARexx or from the batchprocessor, the results will be stored in the file "Analyse−Data.log" located in the current saver−path (or the destination path of the batchprocessor).

Analyse−Spect2D

Produces a 2−dimensional frequency−spectrum plot of a sample. This tells you which frequencies are part of the sound over the time. Additionally this helps to spot anomalies and defects, such as klicks and cracks.

Parameter

Palette	• gray : the display uses a grey scale palette • color : the display utilises a high contrast colour palette
Window (W1)	what windowfunction to use
Lines	how many timesslices should SFX render.
MaxLin.	how many timeslices will fit on this screen.
Bands	Just how many bands should SFX use. Less Bands means less math, but you lose out on accuracy.
Gamma	Nonlinear amplification. Values from 100 % towards 0 % means enhancing quiet details. Values above 100 means hiding them. The default value of 75 % is a good choice to make quiet signals visible too.
Mode	• high 2 : four results are merged into one • high 1 : two results are merged into one • normal : every value in the input will be used to form one result • low 1 : every second value in the input will be used, interpolating inbetween data. • low 2 : every fourth value in the input will be used, interpolating inbetween data.

Notes

When calculations are complete a new window is opened on which the graph is drawn. When the window is active and source−sample is playing, the playposition will be drawn into spectrogram too.
Furthermore you can use the key "C" select on of the following modes : no cross hair, single cross hair, harmonic cross hair. The last causes several horizontal lines to move around when moving the mouse. Each doubles the frequency of the one located below. This alows you to find signal harmonics.
For the calculations the Fast−Fourier−Transformation is used.
If you want to store the generated graphs as images I recommend using a image grabber like SGrab, which can be found on Aminet.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Analyse−Spect3D

Produces a 3−dimensional frequency−spectrum plot of a sample

Parameter

Dir	• front : put samplestart at the front−side • back : put samplestart at the back−side
Window (W1)	what windowfunction to use
Lines	how many timesslices should SFX render.
MaxLin.	how many timeslices will fit on this screen.
Bands	Just how many bands should SFX use. Less Bands means less math, but you lose out on accuracy.
Gamma	Nonlinear amplification. Values from 100 % towards 0 % means enhancing quiet details. Values above 100 means hiding them. The default value of 75 % is a good choice to make quiet signals visible too.

Notes

When calculations are complete a new window is opened on which the graph is drawn. For the calculations the Fast−Fourier−Transformation is used.
If you want to store the generated graphs as images I recommend using a image grabber like SGrab, which can be found on Aminet.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Analyse−Stereo

Produces a graph which shows the spatial distribution of a sample

Parameter

none	

Notes

This is know as well as a phase−plot.
When calculations are complete a new window is opened on which the graph is drawn. A signal where both channels are exactly the same, will appear as a line from the middle to the top (center). When you listen to it with headphones, you will hear the signal inside your head. The Phase of such a signal is absolut synchonous. A complete anti−phase signal (one channel is the inverted copy of the other), will appear as a line towards the bottom (wide). If listening to this on headphones, the sound appears to come from outside. Such a signal is mono−incompatible, which means, if one listens to this on a mono kitch radio he/she will hear absolutely nothing. When analysing real−stereo files, the graph further shows with peaks towards left or right how much "stereo" the signal is. Ideally the graph is a peaked ball around the center with a needle towards the top.
If you want to store the generated graphs as images I recommend using a image grabber like SGrab, which can be found on Aminet.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

ChannelJoin

Joins two separate sample−channels

Parameter

none	

Notes

The sourcesamples must have the same length and number of channels. Of course only mono and stereo samples are supported.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

ChannelSplit

Splits one sample channelwise into two separate samples

Parameter

none	

Notes

Of course only stereo and quadro samples are supported.

[SoundFX] [Modules] [Operators]

Function

Mixes the sample with several slightly detuned and delayed variations of itself

Parameter

Effect (P1)	how much the operator effects the outcome
Voice1...4 (P2...P5)	modulated delaytime.
Interpolation (I1)	how to calculate (smooth) inbetween data
Ampf	final amplification

Notes

Drumloops can give interesting results as they continuously get treblier :) and darker.
Futhermore it is effective to apply the fx to long sustained pad−sounds. These getting more depth by that treatment.

ConvertChannels

Converts between different channel formats

Parameter

Matrix (Mat x y)	All input values a multiplied by those factors and outputed as a sum. Meaningful values for the factors are between −1.0 and 1.0.

Notes

This operator is capable of about every thinkable channel transformation. The sample is feed into the source side and comes out of the destination side. The result will have as many channels, as there are filled destination rows.
The included presets nicely demonstrating the way it works.

Convolve

Applies the impulseresponse in src2 to src1. If you e.g. have the sampled impulseresponse of a church−hall then you can apply this reverberation characteristics to any sample in src1.

Parameter

Effect (P1)	how much the operator effects the outcome
Ampf	final amplification

Notes

You probably have no sample impulsresponses − right, go ahead and try a snaredrum sample (something with a noisy fading trail). The resulting signal gets very loud (depends on the src2−sample) − choose a smaller value of Ampd to compensate.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Crackle

add crackle to a sample

Parameter

Crackle Density	How many crackles should be added

Notes

none

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

CrossTalk

Removes or adds crosstalk of channels

Parameter

Width (P1)	−100 % yields a monosignal and 100 % an extreme expansion
Depth (P2)	same as width, (only avalable when processing quadrosamples)
Ampf	final amplification

Notes

Monosamples can't be processed, as they have no room−information. This cannot be fixed by converting them to stereo samples.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

DeCrackle

Dampens strong leveljumps (crackles)

Parameter

Dif.	Leveljump−threshold. If the detected leveljump lays this much above the average leveljumps in current area, it will dampened.
Amp.	Amplitude−threshold. If the current amplitudes lays this much above the average amplitudes in current area, it will dampened.
Adjust	How strong should the crack be dampened. 100 % means fully cancelation.
Size	The maximum length of an anormal signal to be considered as a crackle. Crackles are normaly relative short. This Parameter is use to separate crackles from percussive sounds.
Test	Starts the operator without modifying the sample and shows the results of the crackle−analysis.
Stat.	The amount of crackles detected (absolut and relative to the length) for each channel of the sample.

Notes

This operator detects cracks in samples and makes them quieter. Such cracks are contained in samples recorded from a longplayer or can be caused by r/w−errors on a disk.
Before using this operator, it is recommended to apply the Middle operator, followed by the ZeroPass operator and finally the Amplify operator with the MaxVol function, to prepare the sample.
If the result obtained by this operator sounds damp and misses attacks, then raise the Dif. and Amp. values, so that fewer signals are interpreted as crackles. If obvious crackles are not supressed zoom into one and look at their length. Then adjust the size parameter accordingly. You can use the Test function to tune the results.

DeNoise−FFT

Removes noise from a sample (multifrequency Noisegate)

Parameter

Attack

Notes

If threshold is set too high, too much of the sample will be suppressed. The result might sound damp in this case
The attack−value should be relative small. If it is too small, the result might sound chopped.
A good way to control the effect is to use the Analyse−Spect2D operator with a low gamma−value (e.g. 0.2). You should clearly see the noise in quiet sections. After applying the DeNoise operator, check again with the Analyser. You should be able to see if the noise−levels have dropped.
It is often very difficult to find the right settings. Processing samples with this operator leads in most cases to an alienated sound, which sounds sometimes very interesting.
This operator uses the Fast−Fourier−Transformation for its calculations.
Before using this operator, it is recommended to apply the Middle operator, to prepare the sample.

DeNoise−FIR

Removes noise from a sample (multifrequency Noisegate)

Parameter

Attack

Notes

If threshold is set too high, too much of the sample will be suppressed. The result might sound damp in this case
The attack−value should be relative small. If it is too small, the result might sound chopped.
A good way to control the effect is to use the Analyse−Spect2D operator with a low gamma−value (e.g. 0.2). You should clearly see the noise in quiet sections. After applying the DeNoise operator, check again with the Analyser. You should be able to see if the noise−levels have dropped.
SoundFX divides the sample into several bands and denoises these. Afterwards the signal will rebuild out of these. The diversion is done by using FIR−Filter.
Before using this operator, it is recommended to apply the Middle operator, to prepare the sample.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Delay

Generates Delays, Echos, Flanger and much more

Parameter

Effect (P1)	how much the operator effects the outcome
Feedback (P2)	how much of the result is feeback into the operator. This may be negative producing an inverted feedback.
Delay (P3)	modulatable delaytime.
Ampf	final amplification
Dry	determines how the propotion of the dry signal from the effect−parameter is calculated
Interpolation (I1)	how to calculate (smooth) inbetween data

Notes

Short delay values (about 10 ms) are known to put a metallic character to the sample.
When a sampled sound ends too abruptley, you can let it fade out with a long delay effect. For this control the feedback by e.g. vector−envelope, which raises the feedback towards the end. In **SoundFX**'s Delay you can even modulate the delaytime and you can enter the delaytime as notes. I know it sounds strange, but it makes sense. If you choose a high feedbackpropotion (> 90 %) and set effektpropotion to 100 %, the sample will resonate on the frequency which corresponds to the delaytime. If you enter a 'C−3', **SoundFX** will calculate the right delaytime so that it resonates on that note.
And there is another useful application of this operator. If you have a sample containing hum and you know it's frequency, then choose Dry='Dry=−Eff', Eff=−100 %, Fb=97 % and Delay=. This will wipeout the frequency and all it's high harmonics. Unfortunately it may take some cycles before the humming fades away. Therefore try to have a bit humming in the begin, which you can just cut later.

[SoundFX] [Modules] [Operators]

DelayPlus

Generates Delays, Echos, Flanger plus some really wiered fx and much more

Parameter

Effect (P1)	how much the operator effects the outcome
Feedback (P2)	how much of the result is feeback into the operator. This may be negative producing an inverted feedback.
Delay (P3)	modulatable delaytime.
Cut−Off (P4)	The filter cut−off is the frequency where the filter becomes active.
Resonance (P5)	Resonance attenuates the sound around the cut−off frequency. A value of 1.0 means no attenuation and higher values will lead to stronger attentuations. If you turn up this too far, that the filter will begin to oszillate (quwiek).
Ampf	final amplification
Type	what kind of Filter do you want it to be
Dry	determines how the propotion of the dry signal from the effect−parameter is calculated
Interpolation (I1)	how to calculate (smooth) inbetween data

Notes

see Delay and Filter−StateVariable

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Detune

Detunes a sample (modulated resampling)

Parameter

Factor (P1)	Pitchfactor. A value of 2.0 means, your result is one octave higher (twice as high higher). The sample will be shortened by the same factor as well.
Interpolation (I1)	how to calculate (smooth) inbetween data

Notes

In this operator pitch and length are coupled. If you want to change only the pitch, have a look at the PitchShift operator and if you want to change the length, try the TimeStretch operator.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Distortion

Creates distortion and fuzz effects.

Parameter

Effect (P1)	how much the operator effects the outcome
Distortion Shape (P2)	this shape determines the kind and the amount of distortion
Map	the shape can be mapped in various ways : • full range : as it is [−max to max] • mirrored : copied and rotated around the origin [−max to 0]=−[0 to max], which yields same shapes for positive and negative sample−values
Wrap	Choose how to handle clipping. The modi below are available : • NoClip: don't care • Clip : overdriven values are clipped • Wrap1 : overdriven values are pushed into the opposite side until they don't clip anymore. • Wrap2 : overdriven values are overturned (folded) until they don't clip anymore.

Notes

The shape acts as a kind of lookup table. If the shape would be a straight line (from bottom left to top right), nothing would change in the sound. But the more different the shape looks like, the more distorted the sound will be.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Duplicate

Doubles a sample multiple times

Parameter

Rep.	Repetitions. How many copies of the sound do you want to have.

Notes

If you got only one period of a waveform (such as most chipsounds) or only cycle of a drum−loop, you can make it longer, by duplicating it several times. This could be neccessary if you want to generate an effect with this sample.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Dynamic

Amplifies or deamplifies the volume of a sample depending on its amplitude. Provides complex changes of the dynamics of the sample.

Parameter

Effect (P1)	how much the operator effects the outcome
Ratio loud (P2)	amplitude−change of loud signals
Ratio quiet (P3)	amplitude−change of quiet signals
Threshold (P4)	determines the break point between the quiet and loud ratio − whenever the signals amplitude exceeds the threshold the loud ratio will be applied otherwise the quiet ratio will be used.
Knee	the are two variant, one is edgy and the other fades smoothly
Characteristics	These graphs show the effect of the settings. Read it a a translation table − the volume of the source sample denotes the x postion, then the curve can be used to find the respective y postion which denotes the output volume.

Notes

Here are a few examples:

- Compressor: squeezes the sample together:)
  Ratio loud <100 %, Ratio quiet >100 %
- Expander: expands the sample
  Ratio loud >100, Ratio quiet <100 %
- Limiter: amplifies the quiet parts of the sample
  Ratio loud =100, Ratio quiet >100 %
- Delimiter: amplifies the loud parts of the sample
  Ratio loud >100, Ratio quiet =100 %

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Echo

Adds echos to the sample

Parameter

Effect (P1)	how much the operator effects the outcome
Delay (P3)	delaytime for the echos
Amplitude (P2)	the volume of the echos
Number	the number of echos
Ampf	final amplification
Interpolation (I1)	how to calculate (smooth) inbetween data

Notes

As SFX mixes the echos to the sample and not only copy them, it's possible that the sample overdrives. Choose an amplification−factor smaller than 100 % to avoid the overdrive.
With the Echo−Operator you could also simulate hall−rooms. Choose short delay−values for this.
And remember; higher number of echos yields longer calculation−times.

[SoundFX] [Modules] [Operators]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Modules] [Operators]

Equalize−3Band

Raises or lowers high, mid and low frequencies. Works like the tone control of you hifi.

Parameter

Lower Cut−Off (P1)	frequency which divides the lower from the middle band, relative frequency ranging from 0 Hz to half of sampling−rate
Higher Cut−Off (P2)	frequency which divides the middle from the upper band, relative frequency ranging from 0 Hz to half of sampling−rate
Lower gain (P3)	amplification for the lower band
Middle gain (P4)	amplification for the middle band
Higher gain (P5)	amplification for the higher band
Ampf	final amplification

Notes

On your hifi you normaly can not change the filter cut−offs. If in doubt, just leave them as they are.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Equalize−FFT−3D

Morphs between 8 equalizer curves in a cube into a result−curve, which then modifies the amplitude of the frequency components of a sample.

Parameter

Frequency−Curves (Eqf1..8)	Your source equalizer curves. When you click on the PopUp−Symbol a file requester appears to let you choose an equalizer preset. These can be made with the Equalize−FFT operator. You can even select multiple presets at once. This will load multiple curves.
X−Axis (P1)	location of the point on the X−axis
Y−Axis (P2)	location of the point on the Y−axis
Z−Axis (P3)	location of the point on the Z−axis
Path	This area shows the path of the curve inside the cube. During the calculation a point will wander along the curve from one end to the other. The distance of the point to the corners defines how much of the equalizer shape assigned to that corner effects the result equalizer shape. With the "View" gadget, you can choose from where to look on the cube and with "Prec." you choose how detailed the curve will be drawn.
Window (W1)	what windowfunction to use
Bands	Just how many bands should SFX use. Less Bands means less math, but you lose out on accuracy.
Steps	SFX does a transformation every sample−values. Lower values mean better quallity but longer calculation. Steps can no be bigger than the half of the number of bands.

Notes

The results of the operator are very unpredictable. This means you are invited to experiment (e.g. try a large noise sample and one of the included presets). The FX is generally well suited to produce e.g. Sci−Fi sounds.
This operator uses the Fast−Fourier−Transformation for its calculations.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Equalize−FFT

Modify the amplitude of the frequency components of a sample

Parameter

Frequency−Curve	Here you can draw the shape of the frequency spectrum.
Arrow−Gadgets	for moving the curve
F−Gadget	Flip, mirrors the curve
Band	number of the band you're currently working on
Val	value of current band
Frq	shows the frequency range for the current band.
Range	Simple tool to do a straight line between two bands. For those who can't draw these lines (like me) just click on the first band then range and then the second band.
Mode	Gives you the choice of moving just the current band or all when using the arrow buttons..
Window (W1)	what windowfunction to use
Bands	Just how many bands should SFX use. Less Bands means less math, but you lose out on accuracy.
Steps	SFX does a transformation every sample−values. Lower values mean better quallity but longer calculation. Steps can no be bigger than the half of the number of bands.

Notes

This operator uses the Fast−Fourier−Transformation for its calculations.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Filter−CRSHiPass

Works on low frequencies, means supresses or boosts them while leting high frequencies pass through unaltered.

Parameter

Effect (P1)	how much the operator effects the outcome. Negative values produce the opposite effect − they boost frequencys.
Cut−Off (P2)	Area for averaging−calculations. The wider that range, the higher the cut−off frequency gets.
Resonance (P3)	Strength of resonance (also Peak or Q−Factor). As a strong resonance thins out the signal, there is an amplification−factor that runs parallel with the resonance, thus gets modulated too. A resonance of 0.0

	should have an Amp=100 %. Higher resonances should get higher amplify values. You'll have to experiment to find the perfect values (try resonance+100 %).

Notes

These filters are based on a very simple model and are therefore not very precise, but quite fast to calculate. And be careful. If you just hear a loud metallic noise, then you've turned resonance up too far.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Filter−CRSLowPass

Works on high frequencies, means supresses or boosts them while leting low frequencies pass through unaltered.

Parameter

Effect (P1)	how much the operator effects the outcome. Negative values produce the opposite effect − they boost frequencys.
Cut−Off (P2)	Area for averaging−calculations. The wider that range, the higher the cut−off frequency gets.
Resonance (P3)	Strength of resonance (also Peak or Q−Factor). As a strong resonance thins out the signal, there is an amplification−factor that runs parallel with the resonance, thus gets modulated too. A resonance of 0.0 should have an Amp=100 %. Higher resonances should get higher amplify values. You'll have to experiment to find the perfect values (try resonance+100 %).

Notes

These filters are based on a very simple model and are therefore not very precise, but quite fast to calculate. And be careful. If you just hear a loud metallic noise, then you've turned resonance up too far.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Filter−FIRBandPass

Works on frequencies except a specific frequency−band, means supresses or boosts them and let the band pass through unaltered.

Parameter

Effect (P1)	how much the operator effects the outcome. Negative values produce the opposite effect − they boost frequencys.
Low Cut−Off (P2)	lower bound of the band, relative frequencies ranging from 0 Hz to half of sampling−rate
High Cut−Off (P3)	upper bound of the band
Nr. (Length)	How many coefficients should be used. The more they are, the better results will be get.(max 1024, but 64 is usually enough)
Window (W1)	what windowfunction to use

Notes

Please don't wonder because of these long calculation−times. If you are using e.g. 64 coefficients, SFX needs to do 128 multiplications and 128 additions for each samplevalue. Since SFX lets you modulate filterspecifications (and not using fixed ones like other programs do), it has to redesign the filter each samplevalue. Therefore again a bunch of calculation steps are neccessary.
For FIR−filters a mathematical coprocessor really helps !

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Filter−FIRBandStop

Works on a specific frequency−band, means supresses or boosts them and let the band pass through unaltered.

Parameter

Effect (P1)	how much the operator effects the outcome. Negative values produce the opposite effect − they boost frequencys.
Low Cut−Off (P2)	lower bound of the band, relative frequencies ranging from 0 Hz to half of sampling−rate
High Cut−Off (P3)	upper bound of the band
Nr. (Length)	How many coefficients should be used. The more they are, the better results will be get.(max 1024, but 64 is usually enough)
Window (W1)	what windowfunction to use

Notes

Please don't wonder because of these long calculation−times. If you are using e.g. 64 coefficients, SFX needs to do 128 multiplications and 128 additions for each samplevalue. Since SFX lets you modulate filterspecifications (and not using fixed ones like other programs do), it has to redesign the filter each samplevalue. Therefore again a bunch of calculation steps are neccessary.
For FIR−filters a mathematical coprocessor really helps !

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Filter−FIRHiPass

Works on low frequencies, means supresses or boosts them and let high frequencies pass through unaltered.

Parameter

Effect (P1)	how much the operator effects the outcome. Negative values produce the opposite effect − they boost frequencys.
Cut−Off (P2)	all frequencies below are getting processed, relative frequency ranging from 0 Hz to half of sampling−rate
Nr. (Length)	How many coefficients should be used. The more they are, the better results will be get.(max 1024, but 64 is usually enough)
Window (W1)	what windowfunction to use

Notes

Please don't wonder because of these long calculation−times. If you are using e.g. 64 coefficients, SFX needs to do 128 multiplications and 128 additions for each samplevalue. Since SFX lets you modulate filterspecifications (and not using fixed ones like other programs do), it has to redesign the filter each samplevalue. Therefore again a bunch of calculation steps are neccessary.
For FIR−filters a mathematical coprocessor really helps !

Filter−FIRLowPass

Works on high frequencies, means supresses or boosts them and let low frequencies pass through unaltered.

Parameter

Effect (P1)	how much the operator effects the outcome. Negative values produce the opposite effect − they boost frequencys.
Cut−Off (P2)	all frequencies above are getting processed, relative frequency ranging from 0 Hz to half of sampling−rate
Nr. (Length)	How many coefficients should be used. The more they are, the better results will be get.(max 1024, but 64 is usually enough)
Window (W1)	what windowfunction to use

Notes

Please don't wonder because of these long calculation−times. If you are using e.g. 64 coefficients, SFX needs to do 128 multiplications and 128 additions for each samplevalue. Since SFX lets you modulate filterspecifications (and not using fixed ones like other programs do), it has to redesign the filter each samplevalue. Therefore again a bunch of calculation steps are neccessary.
For FIR−filters a mathematical coprocessor really helps !

Filter−FIRMatrix

Filters or boosts the signal via a convolution−matrix.

Parameter

Effect (P1)	how much the operator effects the outcome. Negative values produce the opposite effect − they boost frequencys.
Cut−Off (P2)	Area for averaging−calculations. The wider that range, the higher the cut−off frequency gets (you've got to look at this relatively,as the Matrix permits many different characteristics to be set).
Resonance (P3)	Strength of resonance (also Peak or Q−Factor). As a strong resonance thins out the signal, there is an amplification−factor that runs parallel with the resonance, thus gets modulated too. A resonance of 0.0 should have an Amp=100 %. Higher resonances should get higher amplify values. You'll have to experiment to find the perfect values (try resonance+100 %).
Matrix	List of factors for use with the multiplications in the cross section. Values shold not exceed 15.0.

Notes

A Matrix−filter, such as this, is a FIR−Filter where you can enter the coefficients yourself, e.g. if you have designed them with an other program.
With the matrix you can simulate various Filtercharacteristica. If you want to get e.g. a highpass−filter, just set the first value to e.g. 5 and the next ones till cut−off to −1 (e.g. if cut−off=7, then the next six values).

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Filter−FIRMutate

Dampens/boosts the signal. The filter coefficients are taken from src 2. Therefore src 2 controls all the parameters like filter type (lowpass, highpass, ...), the cut−off frequencies, the filter slope and so on.

Parameter

Effect (P1)	how much the operator effects the outcome
Filter−Offset (P2)	Modulates the point in the scr 2 sample, where the operator starts retrieving the filter coefficients.
Filter−Stretch (P3)	Modifies the mapping from samplevalues to coefficients.
Window (W1)	what windowfunction to use
Nr. (Length)	How many coefficients should be used. The more they are, the better results will be get.(max 1024, but 64 is usually enough)
Interpolation (I1)	how to calculate (smooth) inbetween data
Ampf	final amplification

Notes

A filter such as this, is more an experimental thing. There is nearly no way to know the result before. Good results are achived by e.g. changing the filter−offset very little (e.g. linear from 0.0 to 0.1) or using a relativ short sample for scr 2. Furthermore it sounds interessting to blend the filter−stretch from e.g. 0.125 to 8.0.
Curved−interpolation is useful, when using very short sample for src 2 or small filter−offset changes.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Filter−StateVariable

Filters/boosts frequencies according to the filtertype. Can resonanate at the cut−off−frequency.

Parameter

Effect (P1)	how much the operator effects the outcome. Negative values produce the opposite effect − they boost frequencys.
Cut−Off (P2)	frequency where processing starts, relative frequency ranging from 0 Hz to half of sampling−rate
Resonance	From 1.0 to infinity. Too high values will make your sample scream (basicly lots of overdrive on the cutoff−frequency)
Ampf	final amplification
Type	what kind of filter you want to apply

Notes

This filter is not a accurate as a FIR−filter, but is much faster and can resonate.

Fold

Folds the sample data.

Parameter

Effect (P1)	how much the operator effects the outcome
Amp	This parameter tells the operator to hold a constant volume.If deactivated, the amplitude drops for effect−values of 50 % .

Notes

Please be cautious with the Effect parameter as this operator can seriously change (damage ;−)) your sample. (but why bother, you can't bust up your source so play away :)

Gamma

Gamma−correction for sampledata

Parameter

Gamma (P1)	Factor for non−linear amplification/dampening. A value of 1.0 has no effect. A larger value dampens the data (makes quiet signals even quieter). A smaller value amplifys (makes quiete values louder).

Notes

You may need this operator in the folowing case: You've got a sample which uses the full amplituderange, but is still too quiet because of it's Dynamic. To make it louder you must amplify it without to change the volume of the maxima and minima − only amplify the values in the middle. This is exact that, what this operator does. (It is basically the same as a gamma−operator for image−processing.)

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Hall

Reverberates the signal. Simulates three reflectionphases − early reflections, mainhall, diffuse hall.

Parameter

Effect (P1)	how much the operator effects the outcome
Feedback, Early Reflections (P2)	how much of the result is feeback into the operator. This may be negative producing an inverted feedback.
Volume, Early Reflections (P3)	how loud the early reflections appear in the result
Delay, Early Reflections (ErDelS,ErDelE,ErNr)	nr of delays and the time−range they cover
Feedback, Main Reflections (P4)	how much of the result is feeback into the operator. This may be negative producing an inverted feedback.
Delay, Main Reflections (MrDelS,MrDelE,MrNr)	nr of delays and the time−range they cover
Diff	length of diffuse hall.
Ampf	final amplification

Notes

I know that this is far from perfect yet. It's basically the same algorithm as before, but with lots of parameters exposed.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Invert

Swaps upper and lower half of the sample

Parameter

Effect (P1)	how much the operator effects the outcome

Notes

Should the effect parameter be set to 1000 or visa versa, then somewhere an area of the sample could be zeroed (oppposite match).
Mixing an inverted sample on top of the original with a slight delay can produce effects that sound like resonance stuff.
creating a stereo sample from a mono source, when one channel is inverted produces wide−stereo sounds.

Logic

Does a logical operation to the sampledata whith the choosen function.

Parameter

Effect (P1)	how much the operator effects the outcome
Logic Operant (P2)	value what is to be used for the operation
Type	What function should be used

Notes

For people who like to experiment. Effect parameter should be kept low.
One can also use this operator to "encode" or encrypt samples. For this you need a sample TO encrypt and a sample THAT encrypts. The later is the key. Set the LogicOperand parameter to 32767 lv −32768 lv and activate the blendshape "User/Normal". Choose your key sample for the modulator. The Effect parameter is set to 100 %. Choose the function "Xor". After the operation, if you'll listen to the new data, there won't be much in terms of listenable sound. Repeat the same operation and you'll get your sample back the way it was.

[SoundFX] [Modules] [Operators]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Modules] [Operators]

Middle

Searches for the middle of sample data and centers the sample on the x−axis.

Parameter

none	

Notes

Whenever you digitize sound, it can happen that the sample data lies a bit off the x−axis. The middle of the sample just ain't where it should be. On the x−axis. This means that your sample contains an overall offset in its data and should you apply effects to it, can drift away from the middle and at some point become overdriven onesidedly. (One half runs up (upper) or down (lower)). This operator prevents this from ever happening again. (Though some of you will overdrive samples a lot at some time, this ain't the cause anymore. It's your fault:)
Apart from overdriving this is important for restauration (DeCrackle, NoiseGate, ...) so that these operation can correctly analyse the signals.

[SoundFX] [Modules] [Operators]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Modules] [Operators]

Mix−3D

Mixes 8 samples via a path in a cube

Parameter

Sources	The source−samples which then go into the mix.
X−Axis (P1)	location of the point on the X−axis
Y−Axis (P2)	location of the point on the Y−axis
Z−Axis (P3)	location of the point on the Z−axis
Path	This area shows the path of the curve inside the cube. During the calculation a point will wander along the curve from one end to the other. The distance of the point to the corners defines how much of the source assigned to that corner will gets mixed into the result. With the "View" gadget, you can choose from where to look on the cube and with "Prec." you choose how detailed the curve will be drawn.

Notes

Just mix various versions of one sample together and do this twice with different curves. Then join the results to one stereo sample.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Mix

Mixes two samples.

Parameter

Mixratio Source 1 (P1)	how much of source 1 goes into the result; controls the propotion of source 2 as well, which is 100 % minus this value.
Delay Source 2	delays the source 2

Notes

Smooth change in the mixratio can be used to create blending from one sample to another.
Due to the fact that the mixing is done in 80−bit resolution too, there is no need to have a 'clipping' mixmode. Try it and you'll see.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Morph−FFT

Changes the frequencyspectrum of source 1 in that of source 2.

Parameter

Morph (P1)	Controls the transition from source 1 to source 2.
Bands	Just how many bands should SFX use. Less Bands means less math, but you lose out on accuracy.
Steps	SFX does a transformation every sample−values. Lower values mean better quallity but longer calculation. Steps can no be bigger than the half of the number of bands.
Window (W1)	what windowfunction to use

Notes

Try to slowly morph e.g. two long sinewaves with different pitch.
This operator uses the Fast−Fourier−Transformation for its calculations.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

MultiDelay

Generates up to 8 Delays at once.

Parameter

Delay (Del1...8)	delaytime.
Volume (Amp1...8)	how loud should this delay be
Fb Local (FbL1...8)	how much of the result is feeback into the delay. This may be negative producing an inverted feedback.
Fb Global (FbG1...8)	how much of the result is feeback into the operator. This may be negative producing an inverted feedback.
Ampf	final amplification
Dry	how loud should the source be mixed in
Num	how many delays should be used

Notes

As a novelty since V 3.4 you can enter the delay−time as notes too. Just load the preset "Resonate−CEG". With those settings you let the sample resonate on the c−major chord. This gets more clearer, if you run it twice, but its strongly recommended to process the source sample with Middle before.

[SoundFX] [Modules] [Operators]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Modules] [Operators]

Noise

Generates coloured noise

Parameter

Minimum Change (P1)	minimum level change from one sample to the next one
Maximum Change (P2)	maximum level change from one sample to the next one
SLen	length of noise
SRat	sampling rate of the sample. Can be entered as rate, note or with the period−window.

Notes

none

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

NoiseGate

Fades parts which are quieter than the threshold out.

Parameter

Threshold (P1)	Amplitude which serves as a threshold for starting to fade out
Attack	the sound will not just get muted, it be be faded out and back in.
Shape	type of the fade

Notes

Can be used with solo recordings (e.g. speech, guitar, ...) that contain noisy pauses.
For percussive material I recommend using shorter attack values (e.g. 0.5 ms), otherwise the attack can be a bit longer (e.g. 1.0 ms).
Bevor man diesen Operator nutzt empfiehlt es sich, erst den Middle Operator anzuwenden, um das Sample vorzubereiten.
Remember this operator a lot more effective with real 16−bit samples than with 8−bit samples.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Panorama−2Ch

Distribute a mono−signal between left and right channel.

Parameter

Left−Right Position (P1)	Propotion for left and right. 0 % (or 0.0) means left and 100 % (or 1.0) right.

Notes

none

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Panorama−4Ch

Distribute a mono−signal between 4 channels.

Parameter

Left−Right Position (P1)	Propotion for left and right. 0 % (or 0.0) means left and 100 % (or 1.0) right.
Front−Back Position (P1)	Propotion for front and back. 0 % (or 0.0) means front and 100 % (or 1.0) back.

Notes

The result could be transformed back into a stero−sample by using the SurroundEncoder. The resulting sample enfolds its depth by playing it via a surround−decoder.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Function

Distribute a mono−signal between left and right channel.

Parameter

Left−Right Position (P1)	Propotion for left and right. 0 % (or 0.0) means left and 100 % (or 1.0) right.

Notes

none

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Function

Distribute a mono−signal between 4 channels.

Parameter

Left−Right Position (P1)	Propotion for left and right. 0 % (or 0.0) means left and 100 % (or 1.0) right.
Front−Back Position (P1)	Propotion for front and back. 0 % (or 0.0) means front and 100 % (or 1.0) back.

Notes

The result could be transformed back into a stero−sample by using the SurroundEncoder. The resulting sample enfolds its depth by playing it via a surround−decoder.

[SoundFX] [Modules] [Operators]

PitchShift

Changes the pitch of a sample without making it shorter or longer.

Parameter

Effect (P1)	how much the operator effects the outcome
PitchShift Factor (P2)	factor for change in pitch
Window	windowrange; use values in the range of 5 to 100 ms for good results
Smooth	how much percentage of the windowrange should used for crossfade; usually between 25 % and 50 %
Interpolation (I1)	how to calculate (smooth) inbetween data

Notes

Before I give some more detailed tips, I will generally describe how this all works. If you want to pitch up a sound, you can achive this by playing the sound faster and thus compressing the wave (on the time axis). Unfortunately this makes the sound shorter as well. To compensate this, **SoundFX** will repeat small chunks of sound to stretch the sample. While doing this **SoundFX** has to take care that those chunks fit relative seamingless together to avoid crackles. The winsize parameter determines how far SFX searches maximaly for a good transition. The size depends on the material to pitch−. I recomend smaller values (30−50 ms) for percussive samples (this avoids that attacks are repeated audible) and longer values (100−200 ms) for synth/pad/string sounds (to avoid loops).
If the modulator is a sine wave and a small pitchfactor is used (+/− 10 ct), you'll get a vibrato effect.
If you'd like to manipulate synthetic waveforms that have a constant period you should enter the period in Winsize. This'll produce clean pitchshifts.
Factors shouldn't exceed 4.0 with sampled sounds as such high factors result in bad pitchshifts (this is due to the way the pitchshifter works). Synthetic waveforms can be pitchshifted however the far you like.
If the result contains crackles try to slightly change the window−size and/or raise the smooth value.

QuantizeHoriz

"Holds" the sample values for a given time.

Parameter

Effect (P1)	how much the operator effects the outcome
Quantisation Range (P2)	how long a sample value is supposed to be held

Notes

This effect gives a sample a "Nintendo" sound. It's also well known as Sample&Hold.

QuantizeVert

Brings down the bit resolution of the sample

Parameter

Effect (P1)	how much the operator effects the outcome
Quantisation Range (P2)	to how many bits is the sample to be scaled

Notes

none

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Resample

Changes the sampling−rate and sample length while preserving the original sound.

Parameter

SLen old	old sample length
SLen new	new sample length. Factor and new rate are calculated and entered
SRat old	old sampling−rate
SRat new	new sampling−rate. Factor and new length are calculated and entered
Factor	factor of change in length and rate. A factor of 1.0 changes nothing.
Lock	Determines which parameter should be locked. If you want e.g. resample several samples with different rates all to the same rate, you would choose "SRat" then.
Interpolation (I1)	how to calculate (smooth) inbetween data
Aliasing Filter	If enabled, the sound gets filtered before resampling. This is important when lowering the sampling−rate.

Notes

When you sample a sound and use it in a music program it is sometimes out of tune. This operator can correct this. For this you'll have to enter the playback rate as for example the rate you hear the note "C" on with this sample. Now you enter "Resample" and set the resampling rate to "C" −> 16780 and generate the new sample. The new sample will play a "C" at the correct rate now.
With "Resample" it is also possible to change the length of the sample for when you want to modulate something with this sample and need to get to a correct length to do so. Use interpolation with this so that the waveform won't gets too "edgy".

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Reverse

Turns the sample backwards

Parameter

Effect (P1)	how much the operator effects the outcome

Notes

When Effect is set to 50% an X−Fade is performed. This means that the reverse sample is mixed into the original sample. This is a neat way to hide loops. Who've just done in string samples as the beginning will sound just like the end.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

SampleJoin

Appends one sample to the end of the other

Parameter

none	

Notes

The sourcesamples should have the same number of channels.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

SampleSplit

Splits one sample at certain positions.

Parameter

Pos	where should it be sliced
GrabMark	get the Splitpos from current range
Splits	how many slices

Notes

If you want to slice e.g. a drumloop, then use e.g. Pos=25 % and Splits=3. You'll get 4 samples then. The Pos parameter denotes the size of one slice. The last slice will contain all the remaining bits though.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Shorten

Optimizes the samplelength.

Parameter

Threshold	The operator cuts the sample from begin and end until the amplitude peaks over the theshold. This level can be adjusted separately for start and end of the sample.

Notes

With 8−bit samples it will be less successful as with 16−bit samples, because the last ones have a larger amplitude range.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Slide

Does a vertical slide to the sample data.

Parameter

Distance (P1)	Value by which the data will slide. Negative values slide down , positive up.
Wrap	Choose how to handle clipping. The modi below are available : • NoClip: don't care • Clip : obverdriven values are clipped • Wrap1 : overdriven values are pushed into the opposite side until they don't clip anymore. • Wrap2 : overdriven values are overturned (folded) until they don't clip anymore.

Notes

none

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Smear

The readout position of the sample data is modulated and the resulting values get mixed into the original data.

Parameter

Effect (P1)	how much the operator effects the outcome
Smear Range (P2)	how far to position
Interpolation (I1)	how to calculate (smooth) inbetween data

Notes

The range shouldn't be too big as that rarely produces nice effects.
Normaly you should modulate the Smear Range−parameter.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Subtract

Subtracts the sample data of 2.sample from 1.sample

Parameter

Delay Source 2	delays the source 2

Notes

If both samples are identical and the delay time is 0 the result is an empty sample.
You can use this effect to determine the change a previous actio has made. Apply an effect then subtract the original from the effect sample. The result is the pure effect signal. An interesting applicatiohn for this is to see what gets lost when using compression 8such as mp3) when saving sounds. Just reload the sample after saving and subtract the compressed from the original.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

SurroundEncoder

Encodes the audio data of a quadro sample into a stereo sample, which when replayed via a Surround Decoder which regain all its depth.

Parameter

Surround/Mode	Invert is faster, but causes cacelation of signals on some room positions. Phaseshift do not has those problems, but is slower. Windowfunction and number of coefficients is only needed for Mode=Phaseshift.
Surround/Window (W1)	what windowfunction to use
Surround/Nr. (PhaseNr)	How many coefficients should be used. The more they are, the better results will be get.(max 1024, but 64 is usually enough)
Rearfilter	Normally the sound which goes to the rear channel gets filtered. Here you can decide if you want this to happen.
Rearfilter/Window (W2)	what windowfunction to use
Rearfilter/Nr. (RearNr)	How many coefficients should be used. The more they are, the better results will be get.(max 1024, but 64 is usually enough)

Notes

Use e.g. the Panorama−4Ch operator to generate quadrosamples.

[SoundFX] [Modules] [Operators]

Swap

Swaps sample data repeatedly within a certain range

Parameter

Effect (P1)	how much the operator effects the outcome
Swap Range (P2)	Range, inside of which samplevalues are to be swapped

Notes

The range shouldn't be too big as the sample might sound "retarded" or like severely stupid mamals coughed into the sample.
The sample will sound sharper, with a saw characteristic, because many trebbles have been added.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Synthesize−Add

Waveform generation through additive and sound− synthesis, including frequency−and amplitude modulation.

Parameter

Wave (Oszillator)	What waveform will be used for the oscillator is determined here : • Sin : Sine • Tri : Triangle • Saw : Sawtooth • Sqr : Square
Wave/Pha. (Phase)	Phaseshift (0−360 Degrees)
Curve Editing/Range	Tool to let you create a smooth linear line between two sliders. Click the first then range and then the second.
Curve Editing/Mode	Here you can choose how to move or flip the sliders with vertical arrows : • Cur : current slider • All : all sliders • Pos : all positive sliders • Neg : all negative sliders
Curve Editing/Nr	Number of the high tone.
Curve Editing/Val	Amplitude for the high tone
Miscellaneous/SLen	Length of the sound
Miscellaneous/OnePer	Calculates the length of one period using the current rate and puts the result into SLen.
Miscellaneous/SRat	Playbackrate of the sample. Can be entered as rate, note or choosen from the period−window
Miscellaneous/Volume (Scale)	Volume of the waveform
Miscellaneous/MaxVol	Calculates the volume for optimum dynamics.
Miscellaneous/Frq (Pitch)	

	Basispitch of the sound to be generated. This can happen directly or through the period−window. It's advisable (read:really good) to choose a "C" as a note so you can use the result in any music program easily.
Harmonics (SVal)	This area has 64 sliders for all the obertone−parts. If the slider's in the the middle (value=0), then this high tone won't be incorporated into the resulting waveform.
Harmonics/horiz. arrows	Horizontal moving of the lists in steps of one or five.
Harmonics/vert. arrows	Vertical moving of the list or the current slider in steps on one or five.
Harmonics/F−Gadget Flip.	Vertically mirrors the list or the current slider.
Frequency (P1)	Factors for frequency−modulation
Amplitude (P2)	Factors for amplitude−modulation

Notes

Every sound consists of one basic tone and many "high" tones. Frequencies of these "high" or upper tones are a multiples of that of the basic tone. With the help of this operator you can build very complex waveforms by entering the different high tones. It might be a good idea to load the example files and take a look or listen in to the resulting waveform. Every high tone has it's own volume setting. The "val" will show it to you in numbers. This value should decline with a rising number of high tones (chance for overdrive:). Positive values are added and negative subtracted. You can produce intersting results for example by taking a basic sample made at "C−2" and another at [C−2] + (([C#2]−[C−2]) / 4) Some examples :

C−0	65.4063913	67.35102453
C−1	130.8127827	132.7574159
C−2	261.6255653	265.5148317
C−3	523.2511306	531.0296635

You now mix these two samples with Mix with a 50/50 setting. This gives you a sample that sounds a bit etheral, alive and fatter.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

Synthesize−FM

Waveform generation by fm−synthesis like on a Yamaha CX−7.

Parameter

Miscellaneous/SLen	Length of the sound
Miscellaneous/SRat	Playbackrate of the sample. Can be entered as rate, note or choosen from the period−window
Miscellaneous/Volume (Scale)	Volume of the waveform
Miscellaneous/Frq (Pitch)	Basispitch of the sound to be generated. This can happen directly or through the period−window. It's advisable (read:really good) to choose a "C" as a note so you can use the result in any music program easily.
Miscellaneous/Operator	Choose for which operator (wave generator) you want to edit wave, ampitude and frequency.
Wave (Oszillator)	What waveform will be used for the oscillator is determined here : • Sin : Sine • Tri : Triangle • Saw : Sawtooth • Sqr : Square

Wave/Pha. (Phase)	Phaseshift (0−360 Degrees)
Frequency	This defines the operators frequency relative to the basis pitch.
Amplitude	This defines the operators amplitude.
Modulation−Matrix	A checked box means that the amplitude of the src−operator modulates the frequency of the dest operator. As you can easilly see, there are lots of variations possible.

Notes

As a speciality of this operator you can import presets saved by FMSynth (fileversion 1.3) as well. I would like to say thank you to the author Christian Stiens for the source at this place.
FM−synthesis is a complex matter. Just have a look at the included presets and modify them. If you generated some good one, just send them to me, so I can include them in further versions.

TimeStretch

Changes the length of a sample without making its pitch higher or lower.

Parameter

TimeStretch Factor (P1)	factor for change of the length
Window	windowrange; use values in the range of 5 to 100 ms for good results
Smooth	how much percentage of the windowrange should used for crossfade; usually between 25 % and 50 %

Notes

Before I give some more detailed tips, I will generally describe how this all works. If you want to pitch up a sound, you can achive this by playing the sound faster and thus compressing the wave (on the time axis). Unfortunately this makes the sound shorter as well. To compensate this, **SoundFX** will repeat small chunks of sound to stretch the sample. While doing this **SoundFX** has to take care that those chunks fit relative seamingless together to avoid crackles. The winsize parameter determines how far SFX searches maximaly for a good transition. The size depends on the material to pitch−. I recomend smaller values (30−50 ms) for percussive samples (this avoids that attacks are repeated audible) and longer values (100−200 ms) for synth/pad/string sounds (to avoid loops).
If the modulator is a sine wave and a small pitchfactor is used (+/− 10 ct), you'll get a vibrato effect.
If you'd like to manipulate synthetic waveforms that have a constant period you should enter the period in Winsize. This will produce clean pitchshifts.
Factors shouldn't exceed 4.0 with sampled sounds as such high factors result in bad pitchshifts (this is due to the way the pitchshifter works). Synthetic waveforms can be pitchshifted however the far you like.
If the result contains crackles try to slightly change the window−size and/or raise the smooth value.

Vocode−FFT

Forces the sources2 (modulator) to "sing" with the sound of source1 (carrier).

Parameter

Effect (P1)	how much the operator effects the outcome
Bands	Just how many bands should SFX use. Less Bands means less math, but you lose out on accuracy.
Steps	SFX does a transformation every sample−values. Lower values mean better quallity but longer calculation. Steps can no be bigger than the half of the number of bands.
Window (W1)	what windowfunction to use
Ampf	final amplification
EAmpf	amplification for the envelopefollower
EFCoef	factor for the inertia of the envelope follower. Meaningful values are ranging from 0.8 to 1.0.
Src2Inv	Should I flip the envelope for the modulator (src2) (loud becomes quiet and reversed).

Notes

Sources should be of high quality. They should be rich with high tones, as the result might otherwise sound too "thin".
In some cases the result seems to be empty. Use Amplify with MaxVol to bring the sample full volume or recalculate the sample with higher Ampf− and EAmpf−values.
Speech samples as Source2 and synth−sounds as Source1 produce good results.
This operator uses the Fast−Fourier−Transformation for its calculations.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules] [Operators]

ZeroPass

Fades the volume at beginning from 0 in and at end to 0 off

Parameter

FadeIn/Range (SRange)	Range for fading the sound in
FadeIn/Shape (SModShape)	Type of fade
FadeOut/Range (ERange)	Range for fading the sound out
FadeOut/Shape (EModShape)	Type of fade

Notes

If a sample doesn't starts or ends with a value of zero, we hear that as cracks during play. This operator force the begin and the end to zero and fades to normal volume. The shape "slowdown" creates a fade that sounds linear to the ear.

[SoundFX] [Modules] [Operators]

[SoundFX] [Modules]

2.2 Loader

A loader is a module, which loads samples in a certain . **SoundFX** offers loader for the most common formats.

If you have a sample which can not be loaded, then there are two main reasons for this :
1. I have coded the loader badly

2. I don't know about the or don't support it yet
In the first case, please mail the offending sample to me. In the second case do so as well, but try to send as much information about the with it. So if you can dig out some documentation about the in the unexplored depths of the world wide web, chances are rising phenomenally that this can be loaded in one of the next **SoundFX** version. If the formats supports various variants (compression, different bit−depths, etc.) don't hesitate to send me a rich set of test files.

Nearly all savers have a few things in common, which I will describe below. After loading all loaders generate a file−comment with information like , channels and length, if the disk is not write−protected. If there is already a filecomment, if will not be overwritten.

Contents

2.2.1 List of loaders

The following loaders are currently available:

Contents

CDDA−Direct_L

Copies files digitaly (1:1) from CDs. This has the advantage of very high quality, because it avoids converting the data (digital−>analog and again analog−>digital).
Instead of a file requester, a track listing will appear, where you can choose the desired track and set start/end/length.
Read the chapter about recording/sampling as well.

Parameter

Device	Name of the device−driver which controls the cd−drive.
Unit	Numer of device
Method	The method which should be used to read from the drive.
Memory	Which memory should be used for internal read buffer. • Any : doesn't matter • Fast : choose only if you have some • 24bit : go for this if you experience crashes

Notes

This won't work with all drives. At first not every cd−rom or cd−writer is capable of DAE (Digital Audio Extraction) and what is worse, there is no standart way of doing it. To check, if your drive can do it and if yes how, see the list below.

Plextor	CD−ROM PX−32TS	SCSI	yes	Plextor/Sony/IBM
Plextor	CD−ROM PX−40TS	SCSI	yes	Plextor/Sony/IBM
Ricoh		SCSI	yes	Plextor/Sony/IBM, Toshiba
Teac	CD−523S	SCSI	yes	Plextor/Sony/IBM
Teac	CD−R55S	SCSI	yes	Plextor/Sony/IBM
Teac	CD−R58S	SCSI	yes	Plextor/Sony/IBM
Toshiba		SCSI	yes	Toshiba
Traxdata	CDR4120	SCSI	yes	Plextor/Sony/IBM

Please send me reports to complete this list.

[SoundFX] [Modules] [Loader]

[SoundFX] [Modules] [Loader]

Clipboard_L

Loads files from the clipboard. You can exchange data with other programs via the clipboard.
Instead of a file requester, a clipboarde requester will appear, where you can choose one of 256 clips.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16)

Parameter

none	

Notes

none

[SoundFX] [Modules] [Loader]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Modules] [Loader]

DataTypes_L

Loads Sample−Files via AMIGA OS DataTypes. This loader would load every sample, if you have a datatype for its installed. You can try this, when the Universal−Loader fails.
The main disadvantag of the system shipped with OS3.x, is that it only supports 8bit mono samples. Fortunately **SoundFX** can use the extensions introduced by the sounddt41 (which can be found on aminet).

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16)

Parameter

none	

Notes

none

FutureSound_L

Loads FutureSound files. The FutureSound is a very old featuring little. Basically it is a RAW sample with a small chunk of data in front of it, in which length and sampling rate are stored.

Channels	no (mono)
Compression	no (PCM−8)

Parameter

none	

Notes

none

IFF−16SV_L

Loads IFF−16SV Samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−16,FDPCM−16:6,EDPCM−16:5)

Again I got the info on this from Richard Körbners freeware program **SoundBox**. It is basically the 8SVX ,except that it carries the "16SV" mark and stores 16bit sample data in the "BODY" chunk.

Parameter

none	

Notes

none

IFF−8SVX_L

Loads IFF−8SVX Samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,FDPCM−8:4,FDPCM−16:6,EDPCM−8:4,EDPCM−16:5)

This is the most wide spread sound−file on the Amiga. It is build like any other IFF file making it a very flexible whilst retaining compatibility. The IFF−8SV is one of the few that saves loops.
SoundFX also supports quadrosamples, 16−bit and combined samples. I have got the description of the combined samples from the freeware program **SoundBox** by Richard Körber. This saves the full 16−bit data of a sample. Is this sample loaded into a standard program (supporting only plain IFF−8SVX files) then it loads as a standard 8−bit sample. If a program however knows this it loads it as a 16−bit sample.

Parameter

none	

Notes

When **SoundFX** saves a sample in the 16−bit it creates a "BITS" chunk of the following structure :

```
struct chunk_bits {
 char id[4]; // "BITS"
 ULONG len; // 4L
 ULONG bits; // 8/16 bit so far supported
};
```

In addition the "CHAN" chunk has been extended. With a data value of 30, it is a quadrosample.

IFF−AIFC_L

Loads IFF−AIFC Samples.

Channels	yes (mono/stereo)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,A−LAW,µ−LAW)

You will find this fileformat mainly on Apple−Macintosh computers. The AIFC is an extension of the AIFF . It now supports multichannel samples, several bit resolutions and compression.

Parameter

none	

Notes

none

IFF−AIFF_L

Loads IFF−AIFF Samples.

Channels	yes (mono/stereo)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32)

You will find this fileformat mainly on Apple−Macintosh computers. The AIFF supports multichannel samples and several bit resolutions.

Parameter

none	

Notes

none

IFF−MAUD_L

Loads IFF−MAUD Samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,FDPCM−8:4,A−LAW,µ−LAW)

This is an IFF−type , which was introduced by MacroSystems (the producer of the Toccata and Maestro−boards). Is supports multichannel samples, several bit resolutions and compression.

Parameter

none	

Notes

none

MPEG_L

Loads MPEG Samples.with the mpega.library

Channels	yes (mono/stereo)
Compression	yes

You will find lots of sample in this on the internet. Due to its high compression ratio, it's of excellent use for compressing whole songs.

Parameter

Engine	Allows to choose an mpega.library compatible decoder library. There are versions of the mpega.library availabkle, which offer better quality (FPU and MAS) but run slower.
Layer Id align="left">These settings affect Layer I and Layer II files. You can choose the quallity for decoding for mono and stereo files separately. The lower it is, the faster it loads. If you want to save some memory you may force the loader to convert stereo files to mono	
Layer III	Same as above, but for Layer III files.

Notes

none

[SoundFX] [Modules] [Loader]

[SoundFX] [Modules] [Loader]

Maestro_L

Loads Maestro Samples.

Channels	yes (mono/stereo)
Compression	yes (PCM−8,PCM−16)

This failrly simple is saved by the Samplitude Software. The Loader is in an experimental state, because I lack of information about this .

Parameter

none	

Notes

none

[SoundFX] [Modules] [Loader]

[SoundFX] [Modules] [Loader]

RAW_L

Loads RAW Samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,A−LAW,μ−LAW)

A RAW sample realy isn't a . It's 'raw' sound data. This is an advantage in one way as it's easy to handle. The downside is that no other information but the sample itself is saved (no loop points,bit resolution...). **SoundFX** at least trys to scans the sample for sign−type, bit−resolution and endian−type (16 bit).
As a new feature since version 3.70, you can program the RAW−loader by yourself. If you work often with e.g. data from audio−cd's, then name those files ".cdda". To program the loader, you set all parameters in the left half of the RAW−loader :

Type	=PCM16
Endian	=no
Sign	=signed
Channel	=mono/stereo interleaved
SRate	=44100
Offs	=0

Save this as "cdda.cfg". No click on Add (on the right half) to create a new type (the CheckFileTypes has to be selected on for this). Enter ".cdda" into the field which contains "extension/header". Now click on that popup−symbol and select the "cdda.cfg". Everytime a files end on ".cdda" the settings from "cdda.cfg" are used now. If you want to check the file contents and not the ending, use a "#" instead of a "." as the first char (e.g. "#ALAW").

Parameter

Type	type of compression • PCM8 : not compressed 8bit • PCM16 : not compressed 16bit • PCM24 : not compressed 24bit • PCM32 : not compressed 32bit • μ−Law : μ−Law (14:8) compressed 14bit • μ−Law Inv : μ−Law (14:8) compressed 14bit, with inverted bits (ISDN−Master) • A−Law : A−Law (14:8) compressed 14bit • A−Law Inv : A−Law (14:8) compressed 14bit, with inverted bits (ISDN−Master)
Endian	should SFX convert endians. Intel−proecessor based systems store 16 bit data inverted, this oprions fixes that.
Sign	load the Sample as a signed or unsigned sample. • signed : Amiga, Sgi • unsigned : Mac, Atari, PC
Channel	with how many channels is the sample stored and in which way.
SRate	which samplingrate should be used
Offs	how many bytes should be skiped in the begin (to skip a header of known length).
Check File Type	should SFX check the file extension and investigate the data statistically to find out the and adjust the loading parameter accordingly.

Notes

The offs parameter is **not** for seeking into the sample, although it can be used that way. For 16bit sample you need to take care, that you only skip an even number of bytes then.

[SoundFX] [Modules] [Loader]

[SoundFX] [Modules] [Loader]

RIFF−WAV_L

Loads RIFF−WAV Samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,A−LAW,µ−LAW)

This was introduced by Windows on the PC and borrows heavily from the IFF standard. The WAV represents one of the most used formats on the PC.

Parameter

none	

Notes

none

SDS−File_L

Loads Sample Dump Standard files.

Channels	no (mono)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32)

This allows you to exchange samples with you sampler (profi−sampler, not those parallel port ones). Additionally you need a SysEx dumper. Send the sample from the sampler via MIDI/SCSI and save the received data to a file (prefered ending .SDS). These files can be loaded into **SoundFX** then.

Parameter

none	

Notes

none

SND−AU_L

Loads SND−AU samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,A−LAW,µ−LAW,IEEE−32,IEEE−64)

These samples come mainly from the SUN, NEXT or DEC computers or in common : most UNIX−based machines are using this . The is pretty simple − a small header followed by the sound data. In most cases these are µ−Law packed.

Parameter

none	

Notes

none

[SoundFX] [Modules] [Loader]

[SoundFX] [Modules] [Loader]

Studio16_L

Loads Studio16 samples.

Channels	yes (mono/stereo/[quadro])
Compression	no (PCM−16)

Those samples are used with the Studio16 Software, which is bundled with soundcards of the company Surrize. Many thanks to Kenneth "Kenny" Nilsen for his work and help.

Parameter

none	

Notes

This does not support multi−channel−samples (stereo or quadro). **SoundFX** offers a workaround for it. Just save the separate channels in studio16 as name_l.ext and name_r.ext for stereo (where name is the filename and ext is the extension) and name_l.ext, name_r.ext, name_f.ext and name_b.ext for quadro. Then load one of them into **SoundFX**. This loader will then look for the other channels and load them as well.

[SoundFX] [Modules] [Loader]

[SoundFX] [Modules] [Loader]

TX16W_L

Loads samples from the Yamaha TX16W.

Channels	no (mono)
Compression	no (PCM−12)

These samples are always 12−bit, are limited in length to 262144 samples (attack− and sustainpart) and supporting only three different rates (16 kHz, 33 kHz, 50 kHz).

Parameter

none	

Notes

none

[SoundFX] [Modules] [Loader]

[SoundFX] [Modules] [Loader]

Function

The universal−loader tries to identify the sampleformat and loads the sample with the refering loader. It does it in the following way:

- 1.) At first it tries to identify the sample on the basis of its extension.
- 2.) If this is not successful, it tries to find specific strings in the file.
- 3.) If this fails too, it is probably a RAW−Sample and it will be loaded as such.

If a sample is not loaded correctly and you know the , try invoking the respective loader directly.

Parameter

none	

Notes

none

[SoundFX] [Modules] [Loader]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Modules] [Loader]

VOC_L

Loads SoundBlaster−VOC samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,ADPCM−8:4,ADPCM−8:3,ADPCM−8:2,A−LAW,µ−LAW)

The VOC was introduced by "Creative Labs", creators of the Soundblaster−cards on the PC. It was created for easy playback from disks and hard disks or CDs giving it a host of advantages. However due to inconsequent planning it became nessecary to 'add' features which slow down handling of this . Most programs aren't able to read but one (the 1.1 version) of the VOC . **SoundFX** can read and write all known versions of this .

Parameter

none	

Notes

none

[SoundFX] [Modules] [Loader]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Modules]

2.3 Player

A player is a module which outputs samples over a certain audio−devices.

Contents

2.3.1 List of players

The following players are currently available:

Contents

Function

Plays the active sample with the AHI audio system by Martin Blom. This can be downloaded from the following sources :
Aminet:
ahidev.lha
ahiusr.lha
ahiman.lha

WWW:
http://www.lysator.liu.se/~lcs/ahi.html

Parameter

Audiomode	Here you can choose the auidomode (which audiohardware, how many channels,...)and what mixing frequency (sampling rate for playback) should be used.

Notes

none

Function

Plays the active sample over cascaded soundchannels in 14−bit, whithout extra hardware. The maximum playbackrate on PAL/NTSC screens is about 29Khz and on Productivity screens about 58kHz.

Parameter

HFilter	Hardware filter on/off (Power LED)
RateClip	maximum playbackrate, if the samplerate is higher, SFX resamples whileplaying, so the pitch is right.

Notes

none

[SoundFX] [Modules] [Player]

[SoundFX] [Modules] [Player]

Function

Plays the active sample over cascaded soundchannels in 14−bit, whithout extra hardware. The difference to the normal 14bit−player is, that this one uses the Cybersound callibration. This may further raise the playback−quality. The Cybersound callibration program is e.g. included in :
Aminet:disk/cdrom/14CDPlayer.lha
Aminet:mus/play/play16.lha
The maximum playbackrate on PAL/NTSC screens is about 29Khz and on Productivity screens about 58kHz.

Parameter

HFilter	Hardware filter on/off (Power LED)
RateClip	maximum playbackrate, if the samplerate is higher, SFX resamples whileplaying, so the pitch is right.

Notes

none

[SoundFX] [Modules] [Player]

[SoundFX] [Modules] [Player]

Function

Plays the active sample in 8−bit. The maximum playbackrate on PAL/NTSC screens is about 29Khz and on Productivity screens about 58kHz.

Parameter

HFilter	Hardware filter on/off (Power LED)
RateClip	maximum playbackrate, if the samplerate is higher, SFX resamples whileplaying, so the pitch is right.

Notes

none

[SoundFX] [Modules] [Player]

[SoundFX] [Modules]

2.4 Rexx−Operators

A Rexx−operator is a module which can remote−control **SoundFX** via the ARexx−port. It can be used for things like building own effects (which is quite slow) and automating repeative tasks.

Contents

[SoundFX] [Modules]

[SoundFX] [Modules] [Rexx−Operators]

2.4.1 List of rexx−operators

The following rexx−operators are currently available:

Contents

[SoundFX] [Modules] [Rexx−Operators]

[SoundFX] [Modules]

2.5 Saver

A saver is a module which stores sample−data is a certain . **SoundFX** offers you nearly all common variants to use.

Nearly all savers have a few things in common, which I will describe below. If you have choosen the option "save icons" in the prefs the savers will create a standard−icon for the sample. Further all savers generate a file−comment with information like , channels and length.

Contents

[SoundFX] [Modules]

[SoundFX] [Modules] [Saver]

2.5.1 Source selection

These controls are for choosing a source to operate on. The cycle gadget right to the source allows you to choose the range which should be saved. **SoundFX** automatically suggests the probably most desired mode, e.g. if you have marked a range, then range is preselected. The following variants are possible:

choice	description
All	the whole sample will be processed
Window	only the currently visible part (zoomed) will be processed
Range	only the marked range will be processed

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

Clipboard_S

Saves files to the clipboard. You can exchange data with other programms via the clipboard.
Instead of a file requester, a clipboarde requester will appear, where you can choose one of 256 clips.

Parameter

Type	which (IFF−8SVX,IFF−16SV)
Format	type of compression • PCM8 : ungepackt 8bit • PCM16 : ungepackt 16bit

Notes

none

FutureSound_S

Saves FutureSound files. The FutureSound is a very old featuring little. Basically it is a RAW sample with a small chunk of data in front of it, in which length and sampling rate are stored.

Channels	no (mono)
Compression	no (PCM−8)

Parameter

none	

Notes

none

IFF−16SV_S

Saves IFF−16SV Samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−16,FDPCM−16:6,EDPCM−16:5)

Again I got the info on this from Richard Körbners freeware program **SoundBox**. It is basically the 8SVX ,except that it carries the "16SV" mark and stores 16bit sample data in the "BODY" chunk.

Parameter

Type	type of compression • PCM16 : not compressed 16bit • FDPCM16_6 : FibonacciDelta (8:3) compressed 16bit • EDPCM16_5 : ExponentialDelta (16:5) compressed 16bit

Notes

none

IFF−8SVX_S

Saves IFF−8SVX samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,FDPCM−8:4,FDPCM−16:6,EDPCM−8:4,EDPCM−16:5)

This is the most wide spread sound−file on the Amiga. It is build like any other IFF file making it a very flexible whilst retaining compatibility. The IFF−8SV is one of the few that saves loops.
SoundFX also supports quadrosamples, 16−bit and combined samples. I have got the description of the combined samples from the freeware program **SoundBox** by Richard Körber. This saves the full 16−bit data of a sample. Is this sample loaded into a standard program (supporting only plain IFF−8SVX files) then it loads as a standard 8−bit sample. If a program however knows this it loads it as a 16−bit sample.

Parameter

Type	type of compression • PCM8 : not compressed 8bit • PCM16 : not compressed 16bit • PCM24 : not compressed 24bit • PCM32 : not compressed 32bit • PCM16c : not compressed 16bit combined • FDPCM8_4 : FibonacciDelta (2:1) compressed 8bit • FDPCM16_6 : FibonacciDelta (8:3) compressed 16bit • EDPCM8_4 : ExponentialDelta (2:1) compressed 8bit • EDPCM16_5 : ExponentialDelta (16:5) compressed 16bit

Notes

When **SoundFX** saves a sample in the 16−bit it creates a "BITS" chunk of the following structure :

```
struct chunk_bits {
 char id[4]; // "BITS"
 ULONG len; // 4L
 ULONG bits; // 8/16 bit so far supported
};
```

In addition the "CHAN" chunk has been extended. With a data value of 30, it is a quadrosample.

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

IFF−AIFC_S

Saves IFF−AIFC Samples.

Channels	yes (mono/stereo)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,A−LAW,µ−LAW)

You will find this fileformat mainly on Apple−Macintosh computers. The AIFC is an extension of the AIFF . It now supports multichannel samples, several bit resolutions and compression.

Parameter

Type	type of compression • PCM8 : not compressed 8bit

	• PCM16 : not compressed 16bit • PCM24 : not compressed 24bit • PCM32 : not compressed 32bit • µ−Law : µ−Law (14:8) compressed 14bit • A−Law : A−Law (14:8) compressed 14bit

Notes

none

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

IFF−AIFF_S

Saves IFF−AIFF Samples.

Channels	yes (mono/stereo)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32)

You will find this fileformat mainly on Apple−Macintosh computers. The AIFF supports multichannel samples and several bit resolutions.

Parameter

Type	type of compression • PCM8 : not compressed 8bit • PCM16 : not compressed 16bit

Notes

none

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

IFF−MAUD_S

Saves IFF−MAUD samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,FDPCM−8:4,A−LAW,µ−LAW)

This is an IFF−type , which was introduced by MacroSystems (the producer of the Toccata and Maestro−boards). Is supports multichannel samples, several bit resolutions and compression.

Parameter

Type	type of compression • PCM8 : not compressed 8bit • PCM16 : not compressed 16bit

	• PCM24 : not compressed 24bit • PCM32 : not compressed 32bit • FDPCM8_4 : FibonacciDelta (2:1) compressed 8bit • μ−Law : μ−Law (14:8) compressed 14bit • A−Law : A−Law (14:8) compressed 14bit

Notes

none

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

MPEG_S

Saves highly compressed MPEG samples.

Channels	yes (mono/stereo)
Compression	yes

Due to the high compression ratio, this will take a while. In fact it is highly recomended to have an 68060 for this. This module uses external encoders (separate programs). Therfore I've tried to make it highly configurable.

Parameter

Encoder	Choose the executable of the encoder you want to use. It has been testet with the supplied 8Hz, as well as with Pegase, Lame and Ncoder.
Parameter	This is the parameter template which is passed on the command line to the encoder (the encoder will be run as a background process and be feed with data from **SoundFX**). These placeholders are currently suported : • %b : the bitrate • %c : the parameter string for mono/stereo files (see below) • %i : the input filename • %o : the output filename • %r0 : the samplingrate in Hz • %r1 : the samplingrate in kHz (at the moment just 32, 44.1, 48)
MonoStr	The parameter string for mono−files which is used above with "%c".
StereoStr	The parameter string for stereo−files which is used above with "%c".
Wave	This determines in which the sample data is passed to the encoder. • CDDA • RIFF−WAV
BitRate	Strength of compression. Says how many bits per second are allowed. The lower the bitrate is, the lower is the quality.
Pipe	If you have problems with the pipe: device, you may try an alternative one like apipe: or awnpipe:.

Notes

There are presets for most common encoders supplied. Still it is neccesary that you adapt the path of the respective encoder executable or copy the binaries to the sfx/_saver folder under the respective name.
If you want to adapt it for further encoders, I recomend studing the supplied ones.

[SoundFX] [Modules] [Saver]

RAW_S

Saves only the "raw"
sample−data.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,A−LAW,µ−LAW)

A RAW sample realy isn't a . It's 'raw' sound data. This is an advantage in one way as it's easy to handle. The downside is that no other information but the sample itself is saved (no loop points,bit resolution...).

Parameter

Type	type of compression • PCM8 : not compressed 8bit • PCM16 : not compressed 16bit • PCM24 : not compressed 24bit • PCM32 : not compressed 32bit • PCM16c : not compressed 16bit combined • µ−Law : µ−Law (14:8) compressed 14bit • µ−Law Inv : µ−Law (14:8) compressed 14bit, with inverted bits (ISDN−Master) • A−Law : A−Law (14:8) compressed 14bit • A−Law Inv : A−Law (14:8) compressed 14bit, with inverted bits (ISDN−Master)
Endian	should SFX convert endians. Intel−proecessor based systems store 16 bit data inverted, this oprions fixes that.
Sign	store the Sample as a signed or unsigned sample. • signed : Amiga, Sgi • unsigned : Mac, Atari, PC
Channel	with how many channels should the sample get stored and in which way.

Notes

none

RIFF−WAV_S

Saves RIFF−WAV samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,A−LAW,µ−LAW)

This was introduced by Windows on the PC and borrows heavily from the IFF standard. The WAV represents one of the most used formats on the PC.

Parameter

Type	type of compression • PCM8 : not compressed 8bit • PCM16 : not compressed 16bit • PCM24 : not compressed 24bit • PCM32 : not compressed 32bit • µ−Law : µ−Law (14:8) compressed 14bit • A−Law : A−Law (14:8) compressed 14bit • IEEE−32 : floating point 32bit • IEEE−64 : floating point 64bit

Notes

none

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

SDS−File_S

Saves Sample Dump Standard files.

Channels	no (mono)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32)

This allows you to exchange samples with you sampler (profi−sampler, not those parallel port ones). Additionally you need a SysEx dumper. Save the sample to a file and send this to the sampler via MIDI/SCSI.

Parameter

Channel	Midi channel over which the sample should be transfered.
Sample	Sample bank number into which the sample should store the data.

Notes

none

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

SND−AU_S

Saves SND−AU
samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,PCM−24,PCM−32,A−LAW,µ−LAW,IEEE−32,IEEE−64)

These samples come mainly from the SUN, NEXT or DEC computers or in common : most UNIX−based machines are using this . The is pretty simple − a small header followed by the sound data. In most cases these are µ−Law packed.

Parameter

Type	type of compression • PCM8 : not compressed 8bit • PCM16 : not compressed 16bit • PCM24 : not compressed 24bit • PCM32 : not compressed 32bit • µ−Law : µ−Law (14:8) compressed 14bit • A−Law : A−Law (14:8) compressed 14bit • IEEE−32 : floating point 32bit • IEEE−64 : floating point 64bit
Hdr	Fileheader • SND : SUN's • DEC : DEC−workstation's • I_SND,I_DEC : PC with UNIX (LINUX)

Notes

none

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

Studio16_S

Saves Studio16 samples.

Channels	yes (mono/stereo/quadro)
Compression	no (PCM−16)

Those samples are used with the Studio16 Software, which is bundled with soundcards of the company Surrize.
Many thanks to Kenneth "Kenny" Nilsen for his work and help.

Parameter

none	

Notes

This does not support multi−channel−samples (stereo or quadro). **SoundFX** offers a workaround for it. Stereo−samples will be saved as name_l.ext and name_r.ext (where name is the filename and ext is the extension) and quadro−samples as name_l.ext, name_r.ext, name_f.ext and name_b.ext.

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

TX16W_S

Saves samples for the Yamaha TX16W.

Channels	no (mono)
Compression	no (PCM−12)

These samples are always 12−bit, are limited in length to 262144 samples (attack− and sustainpart) and supporting only three different rates (16 kHz, 33 kHz, 50 kHz).

Parameter

none	

Notes

none

[SoundFX] [Modules] [Saver]

[SoundFX] [Modules] [Saver]

VOC_S

Saves SoundBlaster−VOC samples.

Channels	yes (mono/stereo/quadro)
Compression	yes (PCM−8,PCM−16,ADPCM−8:4,ADPCM−8:3,ADPCM−8:2,A−LAW,µ−LAW)

The VOC was introduced by "Creative Labs", creators of the Soundblaster−cards on the PC. It was created for easy playback from disks and hard disks or CDs giving it a host of advantages. However due to inconsequent planning it became nessecary to 'add' features which slow down handling of this . Most programs aren't able to read but one (the 1.1 version) of the VOC . **SoundFX** can read and write all known versions of this .

Parameter

Type	type of compression • PCM8 : not compressed 8bit • PCM16 : not compressed 16bit • ADPCM8_4 : AdaptiveDelta (2:1) compressed 8bit • ADPCM8_3 : AdaptiveDelta (3:1) compressed 8bit • ADPCM8_2 : AdaptiveDelta (4:1) compressed 8bit • µ−Law : µ−Law (14:8) compressed 14bit
Header	file version : • 1.20 : Use blocktype 9 for soundheader. • 1.15 : Use blocktype 8 and 1 for soundheader. • 1.10 : Use blocktype 1 only for soundheader In my experience most programs can't read the new VOC formats. I therefore added the option to save the old . Using version 1.10 might be the safest. Please take these limitations into account: • 1.15 : only 8−bit samples • 1.10 : only mono and 8−bit samples

Notes

none

[SoundFX] [Modules] [Saver]

2.5.2 List of saver

The following savers are currently available:

Contents

3 The ARexx Interface

The ARexx−port of **SoundFX** is called "REXX_SFX". Through this you can "remote−control" **SoundFX** through ARexx−Scripts. This way you can use **SoundFX** for processing samples for other programs FROM that other program (e.g. a Music program). You can even write own effects.
Important: from version 3.70 all commands are prefixed by "SFX_" to avoid command−collisions.

Have a look at the scripts that come with your **SoundFX** installation. They are installed in the directory named "_rexx".
When writing own scripts, I recommend that you start them with code like this, to set the proper arexx−port:

```
IF (LEFT(ADDRESS(), 8) ~= "SFX_REXX") THEN DO   /* not started by SoundFX ? */
        PARSE ARG opts
        sfxport=WORD(opts,1)
        IF SHOW("Ports",sfxport) THEN DO
                ADDRESS VALUE sfxport
        END
        ELSE DO
                IF ~SHOW("Ports","SFX_REXX") THEN EXIT 10
        ADDRESS 'SFX_REXX'
        END
END
```

Contents

3.1 Functions

Currently **SoundFX** offers more than 100 ARexx functions. If you need more, just let me know. Below you can find an overview of all functions (please note that all function−names start with "SFX_"):

Contents

Activate
Brings **SoundFX** screen to front

CleanUp Mode[0=Cur,1=All,2=AllNormal,3=AllZoomed]
Reorder window(s) on **SoundFX** screen

DisableChannel BufferId ChannelNo
Deactivate a given channel

EditCopy BufferId
Copies the selected region

EditCopyB BufferId
Copies the selected region (sample−begin to region−end)

EditCopyE BufferId
Copies the selected region (region−begin to sample−end)

EditCut BufferId
Cuts the selected region

EditCutB BufferId

Cuts the selected region (sample−begin to region−end)

EditCutE BufferId

Cuts the selected region (region−begin to sample−end)

EditErase BufferId

Erases the selected region

EditEraseB BufferId

Erases the selected region (sample−begin to region−end)

EditEraseE BufferId

Erases the selected region (region−begin to sample−end)

EditGrab BufferId

Copies the selected region into a new buffer

EditGrabB BufferId

Copies the selected region into a new buffer (sample−begin to region−end)

EditGrabE BufferId

Copies the selected region into a new buffer (region−begin to sample−end)

EditOverwrite BufferId

Overwrite samples with contents of the copy−buffer starting from the begin of the selected region

EditOverwriteB BufferId

Overwrite samples with contents of the copy−buffer starting from the begin of the sample

EditPaste BufferId

Inserts the contents of the copy−buffer at the region marker

EditPasteB BufferId

Inserts the contents of the copy−buffer at the region begin

EditPasteE BufferId

Inserts the contents of the copy−buffer at the region end

EditZero BufferId

Silences the selected region

EditZeroB BufferId

Silences the selected region (sample−begin to region−end)

EditZeroE BufferId

Silences the selected region (region−begin to sample−end)

EnableChannel BufferId ChannelNo

Activate a given channel

Exit

Leave **SoundFX** without asking

BufferId= **GetActiveBuffer**

Return the Id of the currently active sample

BuferName= **GetBufferName** BufferId

Returns a string containing the name of the sample−buffer

NumChannels= **GetChannels** BufferId

Returns the number of channels for the given buffer

Length= **GetLength** BufferId

Returns the length of the specified sample−buffer

List= **GetList** ListName[Buffers,Loaders,Operators,Players,Savers]

Returns a new−line delimited list of available modules in the respective category

Value= **GetLoaderParam** LoaderName ParamName

Returns the value of the given parameter of the given loader

LoopEnd= **GetLoopEnd** BufferId

Get the end position of the loop

LoopLength= **GetLoopLength** BufferId

Get the length of the loop

LoopMode[0=Off,1=Forward]= **GetLoopMode** BufferId

Get the loop mode for the specified buffer

LoopStart= **GetLoopStart** BufferId

Get the start position of the loop

MarkXEnd= **GetMarkXEnd** BufferId

Get the x−end position of the mark

MarkXLength= **GetMarkXLength** BufferId

Get the x−length of the mark

MarkXStart= **GetMarkXStart** BufferId

Get the x−start position of the mark

MarkYEnd= **GetMarkYEnd** BufferId

Get the y−end position of the mark

MarkYLength= **GetMarkYLength** BufferId

Get the y−length of the mark

MarkYStart= **GetMarkYStart** BufferId

Get the y−start position of the mark

Value= **GetOperatorParam** OperatorName ParamName

Returns the value of the given parameter of the given operator

ProgDir= **GetProgDir**

Returns the pathname of **SoundFX** installation

Mode= **GetQuietMode**

Returns wheter **SoundFX** is in quiet mode

SampleRate= **GetRate** BufferId

Returns the sampling rate of the specified sample−buffer

GetSample DstAddress

Stores the samples of the currently active buffer into the givven memory location as PCM−8 mono data

Value= **GetSampleValue** BufferId ChannelId Position

Retrieves one 16−bit sample value

Value= **GetSaverParam** SaverName ParamName

Returns the value of the given parameter of the given saver

StorageType[1=Mem,2=Dev]= **GetStorageType** BufferId

Returns the type of storage of the specified sample−buffer

UserInfo= **GetUserInfo**

Returns a text string with information of registered user

VersionInfo= **GetVersion** ComponentName[SoundFX,...]

Returns the version of the specified component in the form X.Y

ZoomXEnd= **GetZoomXEnd** BufferId

Get the x−end position of the zoom

ZoomXLength= **GetZoomXLength** BufferId
Get the x−length of the zoom

ZoomXStart= **GetZoomXStart** BufferId
Get the x−start position of the zoom

ZoomYEnd= **GetZoomYEnd** BufferId
Get the y−end position of the zoom

ZoomYLength= **GetZoomYLength** BufferId
Get the y−length of the zoom

ZoomYStart= **GetZoomYStart** BufferId
Get the y−start position of the zoom

HideBuffer BufferId
Hides a visible sample

ProWinId= **InitProWin** MaxLength Title
Creates a new progress window

ChannelActive= **IsChannelActive** BufferId ChannelId
Returns a alue > 0 if the given channel is active

BufferId= **LoadSample** FileName
Loads the specified file with the currently selected loader

Message MessageText
Displays the supplied text as a message box on **SoundFX** screen

BufferId= **NewBuffer** Length SamplingRate Channels
Prepares a new empty buffer

BufferId= **ProcessSample**
Apply the currently selected operator to the active sample

PutSample SrcAddress Length Name
Loads PCM−8 mono samples from the given memory location into **SoundFX** and names the new sample−buffer

PutSampleValue BufferId ChannelId Position Value
Stores one 16−bit sample value

RedrawBuffer BufferId
Refreshes the sample waveform graphics

Aborted= **RefrProWin** ProWinId NewPosition
Sets the new progress status and check if the user has aborted

RemoveBuffer BufferId
Closes the specified sample−buffer

RemoveProWin ProWinId
Closes the progress window

RenameBuffer BufferId NewName
Gives the specified sample buffer a new name

SaveSample FileName
Saves the currently selected sample under the specified filename with the currently selected saver

BufferId= **SearchBuffer** Name
Looks up a sample buffer by its name

SelLoader LoaderName
Activates the loader with the supplied name

SelOperator LoaderName
Activates the operator with the supplied name

SelPlayer LoaderName
Activates the player with the supplied name

SelSaver LoaderName
Activates the saver with the supplied name

SetActiveBuffer BufferId
Makes the supplied sample−buffer the active one

SetLength BufferId NewLength
Changes the length of the specified buffer

SetLoaderParam LoaderName ParamName Value
Sets the value of the given parameter of the given loader

SetLoaderPreset LoaderName PresetName
Selects a preset for the given loader

SetLoopEnd BufferId NewLoopEnd
Set the end position of the loop

SetLoopLength BufferId NewLoopLength
Set the length of the loop

SetLoopMode BufferId LoopMode[0=Off,1=Forward]
Set the respective loop mode for the specified buffer

SetLoopStart BufferId NewLoopStart
Set the start position of the loop

SetMarkXEnd BufferId NewMarkXEnd
Set the x−end position of the mark

SetMarkXLength BufferId NewMarkXLength
Set the x−length of the mark

SetMarkXStart BufferId NewMarkXStart
Set the x−start position of the mark

SetMarkYEnd BufferId NewMarkYEnd
Set the y−end position of the mark

SetMarkYLength BufferId NewMarkYLength
Set the y−length of the mark

SetMarkYStart BufferId NewMarkYStart
Set the y−start position of the mark

SetOperatorParam OperatorName ParamName Value
Sets the value of the given parameter of the given operator

SetOperatorPreset OperatorName PresetName
Selects a preset for the given operator

OldMode= **SetQuietMode** NewMode[0,1]
(De)activates the quite mode for arexx processing

SetRate BufferId NewSampleRate
Changes the sampling rate of the specified buffer

SetSaverParam SaverName ParamName Value
Sets the value of the given parameter of the given saver

SetSaverPreset SaverName PresetName
Selects a preset for the given saver

SetZoomXEnd BufferId NewZoomXEnd
Set the x−end position of the zoom

SetZoomXLength BufferId NewZoomXLength

Set the x−length of the zoom

SetZoomXStart BufferId NewZoomXStart

Set the x−start position of the zoom

SetZoomYEnd BufferId NewZoomYEnd

Set the y−end position of the zoom

SetZoomYLength BufferId NewZoomYLength

Set the y−length of the zoom

SetZoomYStart BufferId NewZoomYStart

Set the y−start position of the zoom

ShowBuffer BufferId

Makes a hidden sample visible again

ToBack

Sends **SoundFX** screen to back

ToFront

Brings **SoundFX** screen to front

[SoundFX] [The ARexx Interface]

[SoundFX] [The ARexx Interface]

3.2 Naming of operator parameters

Due to that most parameter are similar, I describe them centrally here.

Contents

[SoundFX] [The ARexx Interface]

[SoundFX] [The ARexx Interface] [Naming of operator parameters]

3.2.1 Modulator

These parameters of a modulator can be changed with ARexx. You can find the required <prefix> (e.g. P1) inside the documentation of the operators.

parameter	description
<prefix>S	starting value (modulation returns 0.0)
<prefix>E	ending value (modulation returns 1.0)
<prefix>ModShape	kind of modulation ("None","Curve","Cycle","Vector","User")

Depending on the kind of modulation ,more parameters are accessible.

parameter	description
<prefix>CurveExp	bend (0.0...1.0...infinite)
<prefix>CycleOszi	"Saw","Sin","Sqr","Tri"

<prefix>CycleMode	"Hz","Time","Repeats"
<prefix>CycleFrq	frequency
<prefix>CyclePhase	starting phase (angle)
<prefix>VectorAnz	number of points
<prefix>VectorPos	ix 0..(number−1), pos 0.0...1.0/td>
<prefix>VectorLev	ix 0..(number−1), lev 0.0...1.0
<prefix>UserType	"Normal","Abs","AmpEnv","FrqEnv"
<prefix>UserMode	"Single","Repeat","Stretch"
<prefix>UserModBuf	id of modulation buffer

[SoundFX] [The ARexx Interface] [Naming of operator parameters]

[SoundFX] [The ARexx Interface] [Naming of operator parameters]

3.2.2 Interpolator

These parameters of a interpolators can be changed with ARexx. You can find the required <prefix> (e.g. I1) inside the documentation of the operators.

parameter	description
<prefix>IntType	"None","Lin","Si","Lagrange"
<prefix>IntRange	size of the data range used for interpolation

[SoundFX] [The ARexx Interface] [Naming of operator parameters]

[SoundFX] [The ARexx Interface] [Naming of operator parameters]

3.2.3 Window function

These parameters of a window function can be changed with ARexx. You can find the required <prefix> (e.g. W1) inside the do the operators.

parameter	description
<prefix>WinType	"Rectangle","Bartlett","Fejer","Welch","Hanning","Hamming","Blackman","Kaiser","HalfSine","HalfSineQ
<prefix>WinPar	parameter for the window function

[SoundFX] [The ARexx Interface] [Naming of operator parameters]

4 Error messages and queries

In this chapter you find detailed information about error−messages and queries of **SoundFX**.

Contents

[SoundFX]

[SoundFX] [Error messages and queries]

4.1 Error messages

SoundFX will promptly inform you should it be unable to execute an operation of any kind. This comes in form of a requester on its screen displaying the relevant error message.

Contents

[SoundFX] [Error messages and queries]

[SoundFX] [Error messages and queries] [Error messages]

4.1.1 This is an unregistered version of SoundFX ! ...

You have not yet payed the shareware−fee for **SoundFX**. Thiwhen starting __SFXmbers you when starting **SoundFX** to register soon.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.2 You have used an unregistered version of SoundFX !

You have not yet payed the shareware−fee for **SoundFX**. This message remembers you when exiting **SoundFX** to register soon.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.3 I have already told you that you can not save your samples in the demo−version !

If you try to save something in the demo−version of **SoundFX**, you will get this message. You need to register if you want to save your samples.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.4 I have already told you that you can not use the arexx−port in the demo−version !

If you try to use the arexx−port in the demo−version of **SoundFX**, you will get this message. You need to register if you want to activate the arexx−port.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.5 The installation seems to be incomplete ! ...

Please always install sfx−bin_???, sfx−doc_??? and sfx−data archives. If you omit parts of the software, then this can lead to an unstable installation! Please always use the installer and do do copy the files manually.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.6 This function is not implemented yet!

If some funtions are not yet ready, this message will be shown. It should be gone in the next version.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.7 This operation does not supports device−samples yet

This function does not yet supports samples swapped out to hard disk (virtual memory)!

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.8 Can not open file !

SoundFX can not open the file. If you are going to save a file please verify that the media is writable. It can also be, that the accessbits of the file are not set properly.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.9 Can not read data !

SoundFX can not read data from a file. Probably there are errors in the fileformat (e.g. the file is too short).

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.10 Can not write data !

SoundFX can not write into a file. Eventually problems with the writeprotection or the storage media is full.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.11 Can not access file !

SoundFX can not access the specified file. The cause of it can e.g. be that the file does not exist.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.12 Can not <...> <...> !

SoundFX can not obtain a resource, as probably there is no free memory available or the resource is already in use. In the first case please end other running applications or close large projects, to free the needed memory. Sometimes it is already sufficient to enter the following command in the shell: "avail flush".

[SoundFX] [Error messages and queries] [Error messages]

4.1.13 Can not open library !

SoundFX can not open the specified library with the required version. Check if the library is available and recent enough. You can find out about the later, by using the command "version FULL" in the shell.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.14 Can not close screen ! Please close visitor−windows first !

On the **SoundFX** screen are still foreign windows open. Please close them first, as else **SoundFX** can't close its screen.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.15 Can not make screen public !

There already seems to be a screen with the name **SoundFX** open. If you are not able to close it, you need to reboot your computer the use **SoundFX**.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.16 Can not close that sample yet, its still in use !'

It seems that there is still an operation running that uses this samples. Either wait until the operation that uses the sample is done or cancel the operation.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.17 Clip is empty !

The clipboard is empty. Please copy or cut a region first.

[SoundFX] [Error messages and queries] [Error messages]

[SoundFX] [Error messages and queries] [Error messages]

4.1.18 No AHI System or invalid AudioMode !

The AHI player requires an installed AHI system.
If it is installed, than you have probably not yet choosen an audio−mode to use. Just klick the '?'−button next to the player selection.

4.1.19 Execution of <...> failed !

An action could not be performed for some reason. Please use a tool like "Snoopdos" or "Dostrace" to find out more.

4.1.20 This is not a <...> File !

You probably try to load a file with the wrong loader. If you are unsure, I recommend to use the Universal−loader then.

4.1.21 Can not read this <...> File !

SoundFX does not understand this sub−type. You can get in touch with me and send the file in an email. And if you help me with to find information about this sub−type, chances are good that **SoundFX** will be able to read this file soon.

4.1.22 Sample has no sampling rate

This sample has probably not been saved correctly. Now it can sound too high or too low. Please correct the settings in the sample options.

4.1.23 Can not save the whole wave !

Some file formats are very restricted and can not hold longer samples.

4.1.24 This sample has not been saved correctly ! ...

While loading this sample **SoundFX** has detected error in the file−. **SoundFX** will try to recover as much as possible. If this is sucessful, I recommend to resave the file.

4.1.25 Source must be a <...> sample !

The source sample needs to have the requested number of channels.

4.2 Queries

As soon as you want to start something 'heavy' (e.g. something that might possibly destroy your work or seem 'strange':) **SoundFX** asks you whether you really would like to continue. A requester pops up asking you a relevant question.

Contents

4.2.1 File already exists ! What should I do ?

There is already a file of the name you just have choosen to save a file as. If you select "Okay" the file will be overwritten. "New Name" brings you back to the file−requester and "Cancel" allows you not to do the saving.

4.2.2 Do you really want to quit ?

Safety request, if you are sure to end your session with **SoundFX**. All not saved samples would be lost then.

4.2.3 SoundFX is already running ! Should I start it again ?

You have started **SoundFX** another time. If you choose okay, then it will be this way, otherwise it will be exited immediately. Please remeber that only the first **SoundFX** has an AREXX port. This is because you have to supply the port name in your scripts and there can only be one port of that name.

4.2.4 Do you really want to remove all (hidden/shown) samples ?

Confirm that you really want to close all loaded/hidden/shown samples!

4.2.5 Do you really want to close this sample?

Confirm that you really want to close this sample!

4.2.6 Do you really want to delete this entry?

Confirm that you really want to remove this entry!

5 Workshop

In the next chapter I introduce you to **SoundFX** with the help of examples. You can find most of the final samples inside the drawer "Workshop" in the program drawer.

At first a few general remarks :

- !!!! TRY AND PLAY AROUND !!!! − you could not destroy anything
- intensive use of **SoundFX** is the best way to understand how it works
- do not use only default−settings of the operators
- use the modulation−features − some effects are only interesting, if you modulate something e.g. Detune, Smear
- if you have some questions/problems − write to me !!! − only so I can understand where descriptions are not sufficient, where weaknesses are.

Contents

[SoundFX]

[SoundFX] [Workshop]

5.1 Generating percussion sounds

Next a few examples of how to generate percusive sounds. Typical for those is a hard (short) attack and a short length. For the attack sound noise is often used. To the end a lowpass filter aids to dampen high frequencies.

Contents

[SoundFX] [Workshop]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Workshop]

5.2 Generating synthesizer sounds

Next a few examples of how to generate synthesizer sounds. These are well suited to play melodies and chords.

Contents

[SoundFX] [Workshop]

[SoundFX] [Workshop]

5.3 Generating effect sounds

Next a few examples of how to generate effect sounds. With those one should take care not to overuse them, but totaly without them most songs won't do. A different application is the use when adding sounds to videos.

Contents

[SoundFX] [Workshop]

[SoundFX] [Workshop]

5.4 various effects

Next a few examples of how to generate various complex effect sounds.

Contents

[SoundFX] [Workshop]

6 Appendix

Here you can find e.g. a few lists, tables and summaries for reference.

Contents

[SoundFX]

[SoundFX] [Appendix]

6.1 Future versions

I dont want to tell too much here. Anyway I can assure you that I have lots of ideas on my list and that there definitively will be further versions of **SoundFX**.

Please send stimulations, criticism, ideas, wishes, informations (effects, file formats) − but don't forget I am only ONE HUMAN and not a machine ;−) and my spare time is very limited too.

[SoundFX] [Appendix]

[SoundFX] [Appendix]

6.2 Thanks

Thank you to all people helping me to come this far. Without all the mails **SoundFX** would not be what it is now.
I have decide not to append a list of names here, as for sure I would forget somebody.

[SoundFX] [Appendix]

[SoundFX] [Appendix]

6.3 Glossary

In this section I will explain a few terms, which occur often when working with programs like **SoundFX**. Anyway, I cannot and do not want to replace books about digital signal processing herewith.
If you like to see more terms here, then please suggest them to me.

Contents

6.3.0 Aliasing

When recording a sound, one has to choose a samplingrate high enough to support the highest freuqency in the sound. Otherwise there is aliasing introduced. This means, frequencies which are too high (above the half of the samplingrate) are mirrored around it. So a frequency which is a little too high reappears a little below the boundary.

6.3.0 Bitrate

The bitrate determines how many bits per second are needed for a sound. This unit shows what data throughput will be required to play a file from harddisk or from the internet. Compression technology can significantly reduce the bitrate of an audiofile. The table below gives an overview of common formats and their bitrates :

	bitrate
PCM, 8bit,22050Hz,mono	172.265.. kbit/s
PCM,16bit,44100Hz,mono	689.0625 kbit/s
PCM,16bit,44100Hz,stereo	1378.125 kbit/s
MP3,16bit,44100Hz,stereo	z.B. 128.0 kbit/s
RealAudio,16bit,22050Hz,mono	z.B. 32.0 kbit/s

6.3.0 Bitresolution

The bitresolution determines, with which pressicion the analogue audiodata has been capured. The higher the bitresolution is, the smaller the errors made by digitizing are (quantisation error) and the more authentic the sample would sound. Common bitresolutions are 8−, 12−, 16−bit und 24−bit. Below a small table with resolution, resulting range and usual application :

bits	range			application
8	−128	...	127	home, multimedia
12	−2048	...	2047	home, multimedia
14	−8192	...	8191	semiprofesionell area
16	−32768	...	32767	semiprofesioneller area, homestudio
24	−8388608	...	8388608	professional studio

One can clearly see, that already adding one bit, dramatically increases the range and therewith quality.
The amiga audio hardware normaly only support playback with 8 bits. By using a trick though, you can get 12−bit or even 14−bit.
You can easilly notice the difference, by doing as following :

- load a 16−bit sample (for an 8−bit sample ofcourse both players do sound the same), use a sample with a long decay (e.g. a basedrum, which becomes quite deep to the end).
- pülay the sample with high volume with both players (eventually use headphones).

Noticed the difference in the end?

© by Stefan Kost 1993−2004 www.sonicpulse.de

6.3.0 Channel

One sound can consist of several single sounds, which are played back on separate speakers to generate spatial audio. Below are listed a few types :

name	description
Mono	only one channel and therefore no spacial information.
Stereo	two separate channels (right and left)
Quattro	four separate channels • front left, front right, back left, back right • left, right, front, back
Pseudo Quattro	consists out of 3 or 4 channels • 3 : front left, front right, back middle • 4 : front left, front right, back left, back right This can be realised by a special connection scheme of 3 or 4 speakers with a stereo−source.
Surround	consists out of 4 or 5 channels • 4 : front left, front center, front right, back center • 5 : front left, front center, front right, back left, back right

	This version can be realised by a special connection scheme of 4 speakers with a stereo−source. Much better results you woould gain with a real decoder though.

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Dynamic

The dynamic meassures the span between the biggest and smallest amplitude (volume) of the signal. Usually it is been given in dezibel (db).
Music with a high dynamic requires a recording device which can capture this (means devices with high bitresolution).

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Envelope

An envelope is a segmented curve with a minimal level of e.g. 0.0 and a maximum level of 1.0. Such a curve is used to modulate a parameter of an effect. Below an example : If you would e.g. modulate the volume of a sample by this curve, then it would become lounder in the beginning, reaches then its maximum and would then fade to silence slowly towards the end.

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Filter

Filter are operators which select certain frequencies from a sound and the suppress them. The opposite of filters are boosters. Those would amplify frequencies. In **SoundFX** both are combined into one operator; with a positive effect propotion it filters and with a negative iz boosts.
The names of filter−modules in **SoundFX** consist of two parts, the filtering method and the frequencies the select. Below an overview of the methods :

name	description
CRS	**Cr**oss**S**ection − median filter (simple FIR−filter) These are the most simple, but unfortunately the least controlable filter.
FIR	**F**inite **I**mpulse **R**esponse
IIR	**I**nfinite **I**mpulse **R**esponse
BISQ	**BiSq**uad − combination of FIR and IIR

The ggraphics below are showing the processed frequencies :

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Fourier Transformation

The Fourier−transformation is a method, which divides a sample into its time dependent frequency components. On the base of those data diverse manipulations are thinkable, such as equalizer, vocoder and morpher.
SoundFX uses the FFT (Fast Fourier Transformation).

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Harmonics

Every sound can be composed out of overlapping sinuswaves. These waves are called harmonics. The spectra of a sound is determined by its harmonics.

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Loop

Loops are used to repeat parts of a sample. This can be uused to sustain a sound during that phase longer.
The start− and end−point of a loop should lay on a zero crossing (or atleast on simmiliar levels) to avoid cracks. On the range−toolbar you find function to adjust loops−markers.

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Modulation

Modulation is a process where a parameter gets controlled by signal. This can happen e.g. cyclic by a sine−wave or as well by an envelope. In synthesizers one ooften finds so called LFOs (low−frequency−oszillators). These are used as a modulation source, that means they generate an slowly osillating signal, which changes a different pparameter (e.g. the pitch). An envelope is used to e.g. control the volume.

[SoundFX] [Appendix] [Glossary] []

__HEADER

[SoundFX] [Appendix] [Glossary] []

6.3.0 Overdrive

If you amplify a sound too much, the peaks are going beyond the maximum of the digital range. By that the sound becomes distorted, as this generates harsh harmonics.

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Quantisierung

To manipulate a signal with the computer you need it in digital form. Therfore the signal gets meassured in short intervalls. The values are then rounded and stored. During this process the signal becomes quantized twice (time, amplitude). The rate which we use for taking probes is called samplingrate and the precission corresponds to the bitresolution of the sample. One can apply the following rule of the thumb to both values, the higher the better the result, but the higher the memory−consumtion as well.
If the quantisation of time (samplingrate) is too low, not all frequencies belonging to the signal can be recorded properly. Unfortunately this even mirrors those artefact into other frequencies (aliasing).
During the convertion another error happens − the difference of the real value and the rounded version. This error appears as quantisation−noise. The higher the bitresolution, the less noise there is. If you load a 16bit−sample into **SoundFX** and play it back with 8bit and 14/16bit, you will hear the diffenrence.

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Sample

A sample is digitaly recorded audiodata. The name comes from the fact that we take probes, which are called "samples" as well. They are recorded with a device called sampler (which are exist in different version, from cheap to very expensive) and the process is called sampling or digitizing or in the technical sense as quantisation.

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Samplingrate

The samplingrate tells how many digital values are played back per second. The unit of the samplingrate is Hz (oszillations per second). The half of the samplingrate (nyquist−frequency), determines the highest freuenquency that is contained in a sample. There is a simple explanation for it: to detect a frequency, you need at least one period of the wave and that is at least two values.
As humans hear only up to about 20 kHz, samplingrates of much more than 40 kHz are not usually neccesary. Below common samplingrates are listed :

samplingrate	application
8000 Hz	soundcards (typical for SND−AU samples)
11025 Hz	soundcards (typical for old samples)
22050 Hz	soundcards (typical frequency with many samples)
28867 Hz	max. playbackrate of the Paula−chip in normal mode
32000 Hz	consumer DATs and sampler
44100 Hz	CD−player, soundcards
48000 Hz	DAT−recorder/player
57734 Hz	max. playback of the Paula−chip in productivity mode
96000 Hz	profesional studio equipment

The Amiga audiohardware support a samplingrate of up to about 28 kHz under normal screen−modes and up to about 56kHz under screen−modes with doubled DMA−rate, e.g. "Productivity" (activate such modes only, if you are sure that you monitor can handle it or if you use a graphic−card and there is nothing connected to the normal monitor−output).

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Volume

The volume of a sooound can be given in several ways :

kind	description
maximum volume / peak volume	largest peak in amplitude
average volume	average of all absolute amplitudes
acoustical volume	energy of the sound

SoundFX shows you all these levels inside the sample−window, if you have activated this in the sample options (or generell in the setting for the samples).

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix] [Glossary] []

6.3.0 Waveform

The waveform is the visual display of a sound (graphical display of the samplevalues over time). Below a few basic waveforms :

[SoundFX] [Appendix] [Glossary] []

[SoundFX] [Appendix]

6.4 FAQ

In this chapter I present you a series of **f**requently **a**sked **q**uestions along with answers. If you encounter problems with **SoundFX** please go through this list first, to see if there already is a solution to your problem. If you are not successful, contact me for support.

Contents

[SoundFX] [Appendix]

6.4.1 Features

Q: Will **SoundFX** support virtual memory?
A: Yes, it is available now with V3.70

Q: Will **SoundFX** support the DSP on the Delfina Soundboard?
A: Probably not, because this means lot of work to me and I don't have the time for it.

Q: Will there be a **SoundFX** with PPC support?
A: I'll try to do this, but can't promise anything yet. A prerequisite is that I can buy a modern PPC based AMIGA system.

Q: Will **SoundFX** support MPEG Files? Will **SoundFX** support RealAudio files?
A: MPEG can be loaded and saved. With RealAudio I have my doubts.

Q: Will **SoundFX** support support recording in the near future ?
A: It does it since version 4.00.

Q: Will there be **SoundFX** for Windows/Linux/MorphOS/... ?
A: Such things are as simple as it appears to some people. If there is something in work I will give notice.

6.4.2 Problems

Q: When I play back files from HD, then the sound is interrupted by cracks.
A: At first use a separate partition for swapped files (choose in prefs/vmem). Further use a big block size for this partition (changeable in HDToolBox etc). I recommend 8192..16384 Bytes. WARNING : Changing the block−size will destroy all data on this partition.

6.4.3 Errors

Q: When I try loading a sample of 10Mb, I sometimes get an "Out of memory", even though I still have 13 Mb free.
A: You need those 10 Mb as one block. Enter "avail" inside a shell−window. It will show the larges block available.

Q: I have a 10 Mb sample loaded and still 4 Mb free now I'm trying to cut something (e.g. 512 kb) and I get an "Out of memory".
A: When doing a Cut (or Erase) **SoundFX** has to copy the sample data you want to keep to a new buffer and free the source one.

Q: When starting **SoundFX** under OS3.5 I get the following error "Can't open amigaguide.library >=V34 !".
A: Please check the installation of OS3.5. It seems that it sometimes installs the Data−Types to "libs:datatypes" and not to "sys:classes/datatypes".

6.4.4 Installation

Q: When I load **SoundFX** all Operators, Loader and Savers are empty !
A: Make sure that you have installed a sfx−bin, a sfx−doc and the sfx−data archive. If installation is incomplete **SoundFX** will not run properly.

Q: When I install **SoundFX** installation succeeds quickly, but afterwards the installation directory is empty.
A: Unarchive the lzx−files with '−x' *not* with '−e'. Only '−x' will recreate the full directory structure.

Q: I have got problems with the installation.
A: I generally recommend unpacking all three archive to the same destination (e.g. RAM:) and install afterwards. When you will be asked if you want to overwrite some files during unpacking, your answer does not matter. These files are just the same. This way you can install in one go.

Q: When I install a new version, **SoundFX** starts as a demo version. Do I have to pay for the upgrade?
A: No! All new versions are free for registered users. Paying is strictly voluntarily. To make **SoundFX** easily finding the key−file, it is the best to put it into 'devs:keyfiles/' under the name 'sfx.key'.

6.4.5 Usage

Q: How can I make a sample window forget a range I marked, without cutting or copying or anything
A: You can select it from range−menu or with short−cut Amiga−H.

Q: I am very much intrigued as to why we have control over the vertical component of ranges? Can we grab only peaks of samples?
A: This is currently only for zooming, i.e. you can roughly mark a range, let **SoundFX** extend this that it optimally encloses the peak values and then enlarge the area.

Q: Having a keyboard shortcut for starting an operator would be nice (not Amiga−r, but something to start the calculation in the operator).
A: There is one. Press "Enter/Return".

Q: How can I select the whole sample?
A: Again by using the range−menu or with the shortcut Amiga−A.

Q: About the Del key? I am used to use it for Cut operations, just like SoundForge, CoolEdit, and also word−processors have it.
A: I think the best would be to make all short−cut user definable. Now **SoundFX** uses Amiga−x for "Cut" like all good Amiga−programs do.

Q: If you have a 600 Mb file that you want to process in **SoundFX**, but not room for a second (or third, or fourth) 600 Mb file, how do you handle it? The method of making a new sample every time you apply an effect works well for short samples, but it is a problem for whole recordings.
A: At first − short samples − that is what **SoundFX** is for mainly. As it was often asked for, **SoundFX** learned how to cope with long recordings. If there is no room left in memory, **SoundFX** tries to swap out onto hd and if even there is no space, the operation fails.
If you have alternative ideas, they are very welcome. Just to mention it, I already thought about reusing the space of the source samples for the result. This would work with most effects, but not with all and is sometimes tricky to handle too.

Q: Opening some program windows causes bad refreshing of sample−windows when they are resized. If you open eg.

a loader prefs window, and then attempt to resize a sample−window, its contents is not properly refreshed. The refresh is completed only after you have closed the loader prefs window.
Operator windows DO NOT cause this!
A: The operators are started as separated tasks, most other windows are not. As I don't run them asynchronously, all events you cause for other windows are queued until you close the blocking window.
I don't know if it is worth the work, to make them all asynchronous.

Q: Will the new batch facility allow me to convert them all to WAV in one go, recursively going through the drawers?
A: Yes, as this is exactly what it is for.

6.4.6 Loaders

Q: It would be nice if there were a selection in the loader and saver list−views specifically labeled CDDA load &save. And does **SoundFX** support Motorola byte order CDDA files as well? That would allow conversion of CDDA files from 1 to the other.
A: All can be done. **SoundFX**'s RAW loader has a nice feature − a configurable 'auto−dectection' for raw−files. That means you can associate a file extension or a pattern (some id) in the file with a set of options.
To load cdda−file automatically in the right way, you would create a cdda−preset and edit it's setting to your desire. e.g.:
  Format=16 bit signed
  Endian=Intel
  Channels=stereo interleaved
and associate this with the ending '.cdda'. Then you enable the auto−detection and save this settings as 'default.cfg'. Every time you load a cdda−file via Universal−loader or RAW−loader then, the gets detected properly and converted as said above

Q: It would be nice if you could permanently set the drive unit and device preferences that the CDDA−Direct loader uses.
A: Just choose your device and unit and save this as 'default.cfg'.

6.4.7 Operators : Amplitude

Q: My aim is to do an envelope detector in a few steps (maybe Arexx Script). Does anyone know how to rectify a sample (i.e. mirror the negative parts, absolute value) − I mean not just the display, but the actual sample data. Then I want to LPF the sample to end up with the envelope. What cutoff frequency would be OK?
How is this done when applying the AmpEnv modulation
A: The first is easy. Use AmplifySplit. There you can amplify upper and lower parts individually. Thus you can amplify the upper by "1.0" and the lower by "−1.0" and therfore just inverting the lower part. Then apply a lowpass with a cutoff at about "150 Hz".
Another good idea is to mix the signal with a delayed copy of itself (choose a delay of e.g. "25 ms" in the Mix operator).
But the AmplifySplit and LPF combination works just perfectly. The best results are achieved with the Filter−StateVariable (cutoff between "50" and "200 Hz" and resonance=1). With these values you see really nice and smooth envelope curves which can be used to modulate other effects.

Q: The envelope curve I end up getting lies on the upper part of the window (i.e it only takes positive values), but I would like it to take values from −(max) to +(max).
A: You can use the Slide operator to it down by "50%" and then Amplify to scale it to "200 %". If you use it then in

SoundFX for modulation, just use the modulation mode "abs".
Even easier is to use **SoundFX**'s ability to create those envelopes on−the−fly. You know that **SoundFX** is able to do that ? Choose "blend−shape=User", then activate the settings and choose the source sample (where to grab that envelope from) and "modulation type=AmpEnv" (AmplitudeEnvelope).

Q: In the operator Dynamic, what is the threshold for deciding what is a loud or quiet value? Do we have access to setting this threshold?
A: In former **SoundFX** version this effect was called "CompressorExpander". Such an effect need a threshold to operate on. I renamed the operator in **SoundFX** to Dynamic as it works different. The results are similar though.
You give the Dynamic operator a factor for the loudest value (full amplitude) and a factor for the quietest (zero). In between the operator interpolates linear.

Q: If I give a negative value for quiet, will it sit on zero or be flipped on negative side?
A: SoundFX will never reject a parameter because it looks unfamiliar. That is why you can produce so many different results with just one operator.
When you enter a negative value for the quiet, then it will inverse quiet sounds and the interpolation will range from that negative value to the (probably) positive value for loud.

Q: While mixing a CD last weekend I found that some tracks are much more silent than others. Is **SoundFX** able to do a "maximize" function on a track? I mean to make silent tracks louder and maybe loud tracks more silent and if yes, how?
A: It is easy to make them all loud. You can use the Amplify−Operator for that. Just press MaxVol there and it calculates the optimal amplification. When you have a batch of file to maximize, then use the batch processor :

1. Loader : Universal
2. Operator : Amplify, Preset : MaxVol
3. Saver : e.g. IFF−AIFC

Then you just hit start and select source and destination directories.
If all tracks are amplified to full extent, it is a bit more difficult. It would mean to use Analyse−Data for each track and to write down the "RMS−Volume". Then you could make the louder tracks quieter (with amplify) to reduce their energy (RMS−Volume) or use the dynamic operator to compress the silent tracks (e.g. loud values=1.0, quiet values 1.5).

6.4.8 Operators : Delay

Q: In the MultiDelay operator, what exactly is the difference between local and global feedback.
A: Let me try an illustration :

```
           +--------------------------+
           |                          |
           |             * fbglob +--+
           v          +----------+  |  |
source >---+---+---+--| Delay 1 |--+------+-----> result
               |   ^  +----------+  |  |   |
               |   +----------------+  |   |
               |      * fblocal        |   |
               |                       |   |
               |          * fbglob +--+   |
               |      +----------+  |  |   |
               +---+--| Delay 2 |--+------+
               |   ^  +----------+  |  |   |
               |   +----------------+  |   |
               |      * fblocal        |   |
               :                       :   :
               |          * fbglob +--+   |
               |      +----------+  |      |
               +---+--| Delay 8 |--+------+
```

```
^  +---------+  |
+---------------+
   * fblocal
```

So local feedback is a factor which determines how much of one delays output is feed back to its own input. Global feedback is the factor which determines how much of one delays output is added to a sum which is feed back into all delay−inputs.

[SoundFX] [Appendix] [FAQ]

[SoundFX] [Appendix] [FAQ]

6.4.9 Operators : Filters

Q: How can I change the band assignments in the Equalize−FFT operator? The first band goes up to 648 hz and the last to 22050 hz, that makes not much sense to me.
A: The current equalizer is based on the fft−algorithm. This splits the whole frequency−space into fixed ranges. The overall area covered is 0 Hz to samplingfreqency/2.
For a future **SoundFX** version I plan to include a full−parametric equalizer (n middle bands with editable gain and width plus one low and one high shelf with editable gain and cut−off).

Q: I could not manage to get TB303 like effect on rhythm. This means to sweep a sharp sound from Low to Hi − it is a classic effect that is used in Trance or House songs when whole rhythm is resonating...
Can I get this in **SoundFX**?
A: I belive you can. At first I recommend to use the Filter−StateVariable as this is fast and powerful. A filter has three important parameters :

1. model : lowpass, highpass, bandpass, bandstop, ... − you would choose lowpass in your case
2. the cut−off frequency : this is the frequency where the amplitude has already dropped by 3db
3. the resonance : this attenuates frequencies around the cut−off

SoundFX allows to modulate most parameters and not keeps them static as most other applications do. Both Cut−Off and Resonance are modulatable. Such a parameter can be controlled in nearly every thinkable way. Therefore you have those blend−shapes (the second line of each parameter). You basically enter a start and end value and the shape alternates between them (see Modulator.
A: (Jan Krutisch) I guess a good way to start here is to use Filter−StateVariable as an effect (as Stefan suggested) and let the cutoff be modulated by the signals amplitude. Since I have not used **SoundFX** for some time now, I could not tell you exactly how this is achieved, but you have to set the modulation to "USER". Then you can choose between frequency and amplitude modulation. The only thing you have to do is to set the two values for cutoff to reasonable values (experimentation rules!!!) and set the resonance to a fairly high value. Voila! Instant jumpy filters attack.

Q: Is there any way to increase resolution when doing FFT analysis. I mean doing zooms and such to find exact peaks?
A: Not really. That is the unfortunate limitation of the FFT. If you are interested in lower frequencies, you can zoom using a trick. Just low−pass−filter the signal and resample it (you can use the builtin aliasing−filter of the Resample−operator). Then start AnalyseSpect−2D.
What I could do, is to try to build an spectrum−analyser on the base of bandpass−filters. This analyser could then (nearly) endlessly zoom−into the signal.

Q: When I create a 1 second noise sample using the Noise operator and then perform a spectrum analysis, the result is anything but flat. What is wrong, the noise algorithm or the FFT?
A: It can not be perfectly flat. Depending on the quality of the random−number generator the noise is more or less "white".

Q: I have recorded a sample from a bigger distance and want to increase the volume. But if I do so, I get a background sound. It seems that this sound is above 14kHz. My sample is speech only, so I think it is no problem to cut this background sound of with **SoundFX**.
A: I assume, that you have recored in 16bit with 44100/48000 Hz. The easiest is to use a low−pass filter. Because you want good cancellation, I suggest to use Filter−FIRLowPass (and not the Filter−StateVariable). Start the filter and

enter 13000 Hz for the cut−off, set modulation to none, as you do not want to create artistic sounds. Number should be something like 64.
Use the AnalyseSpect−2D afterwards to verify that high frequencies have been canceled out. You can even apply the filter several time to increase steepness and dampening.

6.4.10 Operators : Quality

Q: I have a speech sample and want to remove the 50 Hz hum frequency. Which filter should I use and which parameters?
A: This is an easy one. Try it, even though it sounds odd. Use the Delay−FX. There should be a preset "DeHummer_50Hz". It resonates on 50 Hz and suppresses the resonation. Works just wonderful. If there is a remaining hum, just apply it twice (or even more often, which is selden required though).

Q: Has anyone experience with Decrackle of records? I've experimented with Decrackle (Dif. 200 %, Amp. 200 %, Adjust 95 %) and achieved good results on bigger crackles. But I can't find good parameters for Filter−FIRLowPass or DeNoise−FIR to eliminate the permanent silent crackles.
A: I don't believe it is possible to get perfect results by trying to remove crackles from LP automatically. There are many peak wave forms which are part of the music but have very similar characteristics to clicks.
In particular, filters acting in the frequency domain are NOT the way to go. A click or crackle is an impulse signal and therefore contains all frequencies. Removing the high frequencies just spreads it out.

6.4.11 Operators : Synthesis

Q: Do we get to be able to program some FM sounds?
A: Use the Synthesize−FM, it can do everything a Yamaha DX−7 can do plus something more here and there.

6.4.12 Operators : Tuning

Q: What is the difference between Detune, PitchShift and Resample in the sense that all 3 could change the pitch of a sound?
A: Detune and Resample are quite similar. They both just output incoming samples at a different speed, thus they change the length of the sample along with the pitch.
The difference is that Resample is doing this with a constant rate (e.g. output 3 values for each two incoming ones), while Detune can do this with a varying factor. Additionally Resample offers a few gimmicks to cure diseases which can be caused by doing this namely aliasing. Therefore an example : think of a wave containing
+−+−+−+−
where + means maximum positive amplitude and − means maximum negative amplitude. Now you down sample it by factor two (skipping every second value) and you would get
++++
The high frequency has canceled itself or more precise even became a dc−offset. And it gets worser if you down−sample by fractional factors (e.g. 1.5). Then you would get something like
+−−+−

There is a very easy way to see and hear the effect. Use Synthesize−Add and generate a sine−sweep from 1000 Hz to e.g. 100000 Hz but choose a sampling rate of e.g. 44100 Hz. The result should be something like puuuuuiiiiiiieeeee ;−) But it will be something like puuuuiiiiiieeeeeiiieeeeuuueeeeiii.
Use Analyse−Spect−2D to visualize it and you will understand.

Practically you would use Resample if you have a sampled recorded e.g. at 22050 Hz and you want to use it in a software which requires samples recorded at 44100 Hz, as Resample changes the sample so that it sounds the same when played back at 44100 Hz instead of 22500 Hz.
In contrast, with Detune you can do things like simulating the effect of a power−outage or alien voices.

Now about Pitchshift. Its speciality is that it changes the pitch without changing the length, by using a lot of magic (intelligently repeating or skipping small fragments of the sample). There is another related effect which will appear in **SoundFX** sooner or later (I wish sooner). It is called Timestretch and it makes a sample longer or shorter without changing the pitch. Basically this is a PitchShift followed by a resample (thus can be simulated in **SoundFX** already).

[SoundFX] [Appendix] [FAQ]

[SoundFX] [Appendix]

6.5 Support

Please read the chapter FAQ before writing to me. Perhaps there is already a solution to you problem.
If you write to me, please include the following details:

- your computer configuration, preferably use "sys:Tools/ShowConfig"
- which sfx−version and what sfx−cpu−version you use, "version FILE=SoundFX FULL"
- where did you detect the error. The more accurate it is, the easier it is for me to fix it. Please do not send inquiries of the kind "... does not work". Describe as detailed as possible, what you did and what you want to achieve. Every tiny detail may matter.
  You may send samples along with the report (if they may be the cause), but please try to restrict them to a maximum length of 0.5 Mb. If a certain operator causes problems, just send me a preset with the respective settings.
  Finally I may make sense to use tools like Snoopdos, Enforcer, Mungwall, ... and to include their output in the mail.

As mentioned, I try to make the program as good as possible and you could help me. I try to answer every email, but not all letters (I read ALL letter and try to fix the reported bugs).

If you have an internet connection, I recommend to check the bug−tracker at www.sonicpulse.de/phpbt/ and to enter your bug−reports, feature−request or help−request there.

[SoundFX] [Appendix]

© by Stefan Kost 1993−2004 www.sonicpulse.de

[SoundFX] [Appendix]

6.6 Technical Background

SoundFX gets developed using the following tools :

Werkzeug	Beschreibung
SAS C/C++	main compiler
GoldEd	editor on Amiga
JEdit	java based editor, used via network mounted drives
htmldoc	for generation of ps/pdf files

gnu m4	for generation of html files
debug tools, splint, muforce	bug−tracking

Many thanks to the contributing authors!

[SoundFX] [Appendix]