Tutorial Abunch (version 048)

A. Installation of Pure Data and Abunch

Before you can start using Abunch, you'll have to install Pd (Pure Data). You can download Pd at www.puredata.info/downloads .

Installing Pd inWindows:

If you have downloaded an installer (a file ending on .exe), just click on this file and Pd will be installed in the normal place C:\Program Files\Pd. If the file is a zipfile, unzip it and put the whole folder (with the name Pd) in your Program Files folder. Go to C:\Program Files\Pd\bin in explorer and click on 'pd.exe' to launch Pd (you can make a shortcut to this file and put in on your desktop as this is the file that you will always need to start Pd).

Installing Pd in Mac OS: double click on the downloaded file. A file like 'Pd-0.42-2.app' will be the result (in the same location as the folder where you the downloaded file was). Cut and paste this file to 'Applications' (where all your programs are stored). Double click on this 'Pd-0.40-2.app' file (or the Pd icon) to launch Pd.

If you want **to test if the sound in Pd is working,** go to 'Media' in the menu of Pd, click on 'test audio and midi' and click on the -40 or -20 button under 'test signal'. You should hear a sound now. You can open a file by choosing 'File' and 'open' in het main menu.

Abunch can be downloaded at www.hansroels.be/abunch.htm

It is a zip file that you should unzip. Leave all Abunch files in the same folder (called Abunch)!

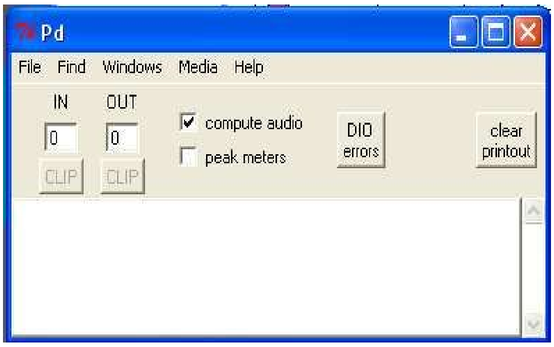
B. Basics

If you want to start using Abunch immediately, just read remarks 1 to 5 and remark 15. Next try out the example files in the Abunch folder, their file names start with 'ex' (like ex1a-playsample.pd). Every example file has a short explanation on how to use them.

This tutorial not only explains how to use Abunch but also Pd (as the two are closely connected).

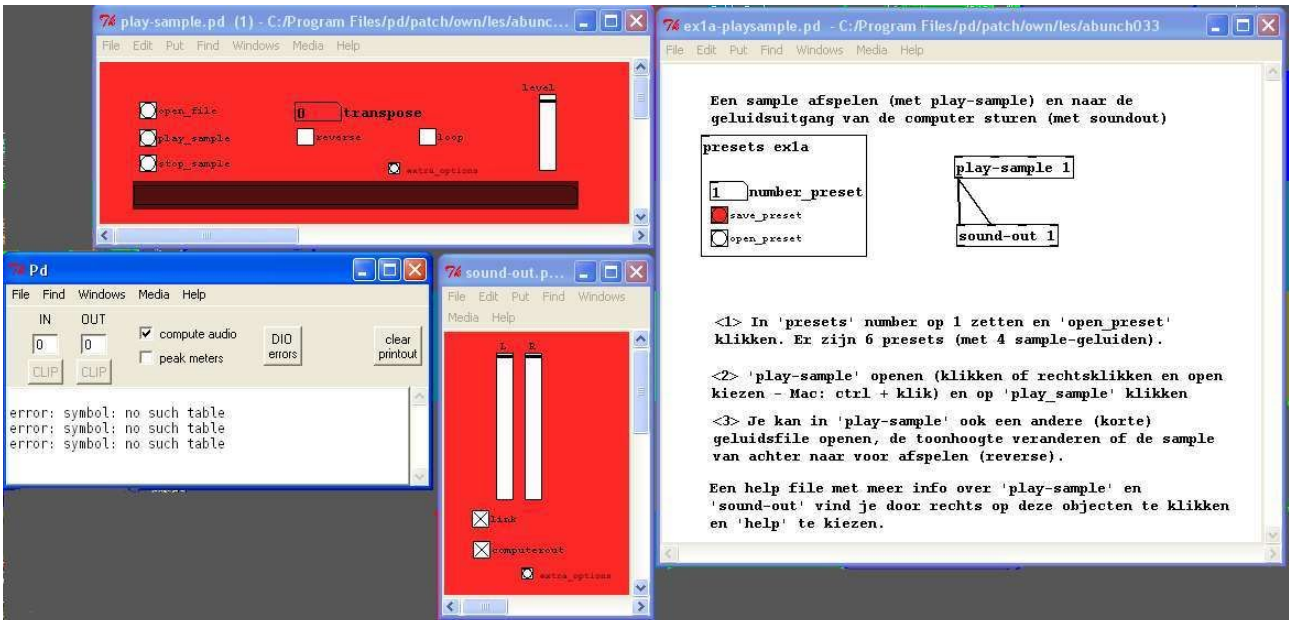
1. Abunch is a collection of objects made in Pd (Pure Data). To use Abunch you should launch Pd.

2. You can only hear sound if '**compute audio**' is switched on in the main Pd window.



The main window of Pd with the sound switched on.

3. Choose 'File' and 'open' in the main menu of the **Pd main window** and open the file 'ex1a-playsample.pd' in the Abunch folder. The window that is opening now is the **file window** (its menu has extra items like 'Edit' and 'Put'). If you click on [play-sample 1] or [sound-out 1] in the 'ex1a-playsample' file a subwindow opens with faders, knobs and switches to perform with these Abunch objects. These two **subwindows** originate from the two objects in the 'ex1a-playsample.pd' window. As long as this main file is opened you can close and open the subwindows of [play-sample 1] and [sound-out 1]. If you close the main file 'ex1a-playsample.pd' these subwindows are also closed.



play-sample.pd (1) - C:/Program Files/pd/patch/own/les/abunc...
File Edit Put Find Windows Media Help
open_file
play_sample
stop_sample
0 transpose
reverse
loop
extra_options
level
Pd
File Find Windows Media Help
IN
OUT
0
0
CLIP
CLIP
☑ compute audio
☐ peak meters
DIO errors
clear printout
sound-out.p...
File Edit Put Find Windows Media Help
L
R
☒ link
☒ computerout
extra_options
ex1a-playsample.pd - C:/Program Files/pd/patch/own/les/abunch033
File Edit Put Find Windows Media Help
presets ex1a
1 number_preset
save_preset
open_preset
play-sample 1
sound-out 1

```
error: symbol: no such table
error: symbol: no such table
error: symbol: no such table
```

Een sample afspelen (met play-sample) en naar de
geluidsuitgang van de computer sturen (met soundout)

<1> In 'presets' number op 1 zetten en 'open_preset'
klikken. Er zijn 6 presets (met 4 sample-geluiden).

<2> 'play-sample' openen (klikken of rechtsklikken en open
kiezen - Mac: ctrl + klik) en op 'play_sample' klikken

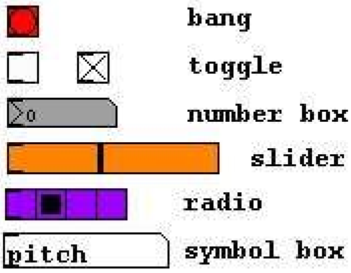
<3> Je kan in 'play-sample' ook een andere (korte)
geluidsfile openen, de toonhoogte veranderen of de sample
van achter naar voor afspelen (reverse).

Een help file met meer info over 'play-sample' en
'sound-out' vind je door rechts op deze objecten te klikken
en 'help' te kiezen.

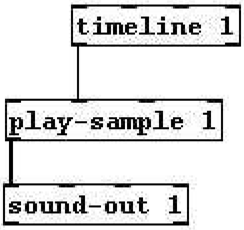
The main window of Pd ('Pd'), the window of the main file ('ex1a-playsample.pd') and the two subwindows in red ('play-sample' and 'sound-out') .

4. The objects [play-sample 1] and [sound-out 1] have **inlets at the top and outlets at the bottom** (the small dark blocks at their border), this applies to any object that has in- or outputs. The sound and computations flow from the top to the bottom. In the example 'ex1a-playsample' [play-sample 1] plays a sample and this sound is sent to [sound-out 1] (the sound output of the computer).

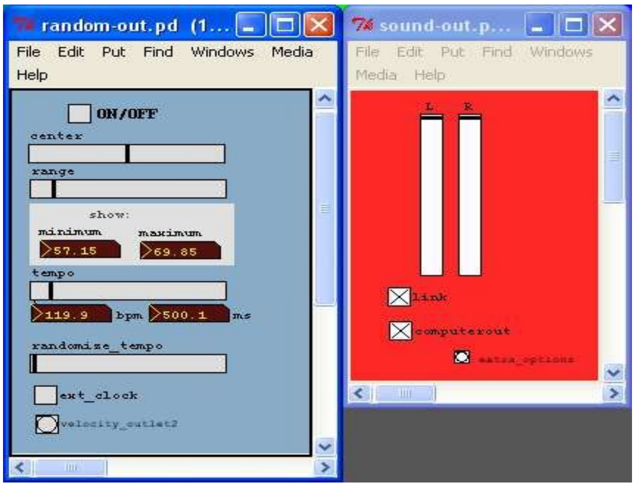
5. You can use the faders and knobs in the opened windows of [play-sample 1] and [sound-out 1]. The size and color of these '**graphical user interface**' (GUI) elements can be different depending on the window that you have opened but the operation is always identical: you should click on a **bang** button (changing the color of the center of this button). You should also click on a **toggle** and then this toggle can be switched on (with a cross) or off (empty). You should click and drag on a **number box** to change the numbers (if you hold the 'shift' key at the same time you can change the numbers in smaller steps). You should also click and drag on a **slider**, in the example picture on the small vertical stripe in the middle (as with the 'number box' you can change in a smaller resolution by holding the 'shift' key while clicking and dragging). You should click on the **radio** row of knobs (a selector) to select one square (in the picture the second square is selected). You should also click on the **symbol box**, type a word and press the 'enter' key ('symbol' in Pd is a synonym for letter).



6. There are **two kinds of connections** in Pd. Thick connections (between [play-sample 1] and [sound-out 1] in the picture) are **continuous audio streams** (44100 samples per second if the sampling rate is 44.1 KHz). From the moment the audio is switched on in Pd (see 2.) numbers are sent at a very high speed (even if you don't hear anything). The thin cables (between [timeline 1] and [play-sample 1] in the picture) are **sporadic control data**, they are only sent if it is necessary.

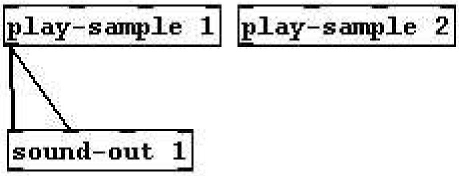


7. All objects in Abunch have **a blue or red background color**: blue stands for objects that mainly compute and send out control data (thin connections), red for objects that send audio streams (thick connections).
The GUI elements (fader, number box,...) with a dark brown background color just show something. It is useless to try to change the values in these boxes by clicking and dragging.

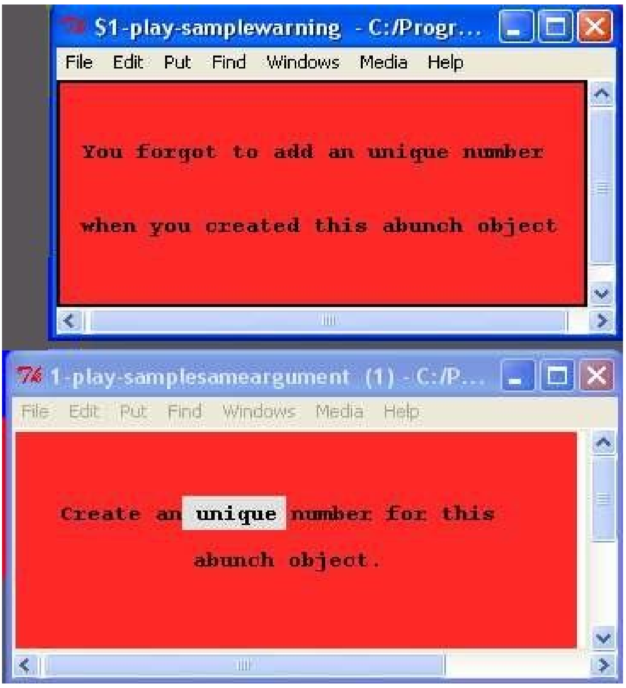


8. Pd and thus Abunch have two operating modes: in '**edit mode**' you can change all kinds of things in the structure, in '**play mode**' you can operate the GUI without changing the underlying structure. With the keys **Ctrl + e** you can switch from one mode to another. In 'play mode' your mouse has an normal look (an arrow), in 'edit mode' an small hand appears on the place where your mouse is standing).

9. You can **make a new object** by choosing 'Put' and 'Object' in the menu of the window of the opened main file and typing a name, space and an **unique number**, like [play-sample 2]. No other object with the same name should have the same number. In remark 20 you can read where to find an overview of all the objects in Abunch. If you make a new object in this way the 'edit mode' (see 8.) is automatically switched on.



10. If you forget to add an unique number (and just type [play-sample] in the empty object), a **warning message** pops up saying: “you forgot to add an unique number when you created this abunch object”. If there is an object with the same name and number (like [play-sample 1]) anywhere in a patch, another warning message pops up with this text: “Create an *unique* number for this abunch object”. The name of this 'double' object is shown in the Pd main window. Adding an unique number to the name of a new Abunch object is the best way to prevent these warning windows.

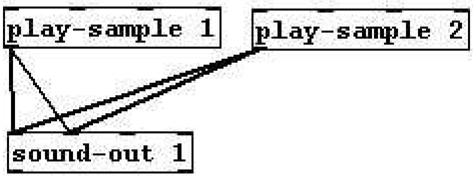

$1-play-samplewarning - C:/Progr...
File Edit Put Find Windows Media Help

You forgot to add an unique number when you created this abunch object

1-play-samplesameargument (1) - C:/P...
File Edit Put Find Windows Media Help

Create an unique number for this abunch object.

11. You can **connect** two objects by moving to the outlet of the first object with the mouse (a small circle appears), clicking and dragging to the inlet of the second object. You can thus connect [play-sample 2] to [sound-out 1] (as in the picture) to hear the sound from the second sample player.



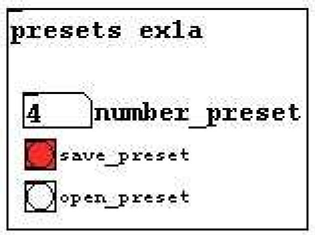
12. Caution: you **cannot connect** in- or outlets for thick connections (audio streams) with in- or outlets for thin connections (control data). A red object can't be connected to a blue one (see 7.).

13. To make life easier there are some **conventions in Abunch**. If an object processes an audio stream (a red object) the inlet at the left is the one for the sound to be processed. If the object is a synthesizer (like 'syna-fm' or 'grain-synth') the first outlet (at the left) is for pitch values and the second for velocity values.

14. If you have a look at the files in the Abunch folder you will notice a file like 'play-sample.pd'. In our example this file is opened within the main file 'ex1a-playsample.pd'. Only if the main file and the Abunch object files are in the same folder, these Abunch objects can be placed and opened within the main file (for the exception see 16). If you want to make a **new file** and put Abunch objects in it, you have to take care that this new file is in the Abunch folder. That's why there is an empty file in the Abunch folder (file.pd). You can open it, put all preferred Abunch objects in them and save it under a new name. A file that is meant to be opened within another file in Pd, is called an **abstraction**.

15. Because Abunch objects in two main files could have the same name and number and conflicts could arise, you should **avoid opening two main files** (like 'ex1a-playsample.pd' and 'ex1b-recandplay.pd'). Always close the pevious main file.

16. If you save a file (choosing 'File' and 'save as' in the main menu), you only save the objects and connections in this file. The position of faders, knobs and the sound files that are used in some objects (like [play-sample]) aren't stored. If you want to save these data you should save a **preset**. This is the way to save presets in Abunch: make a new object (see 9.) and in this empty object type presets, a space and the name for the presets in one word, in the example picture this is [presets ex1a]. The presets object is the only one in Abunch that doesn't need an unique number but a (unique) word as a name for the presets.



17. The **presets** object saves the position of faders and knobs **globally** -for all opened Abunch objects- . Sometimes it can be handy to open a preset of one object (a local preset) without changing something in other opened objects. That's why this local feaure is built into some of the Abunch objects ('syna-fm', 'keyboard-azerty',...). Have a look at the help file (see 19) of an Abunch object for more detailed information.

18. The '**presets**' object stores the presets as a **text file** (.txt) in the folder 'presets' in the Abunch folder. The name of this preset file is exactly the same as the name that you have given to the presets (thus in the example of 16. a file 'ex1a.txt' is generated).
The **preset system in Abunch before version 035** was different. You can convert your old presets by placing the object 'presets-old <name>' and a second object 'presets <name>' in your file. Open the presets one by one in the 'presets-old' object and restore them in the 'presets' object.

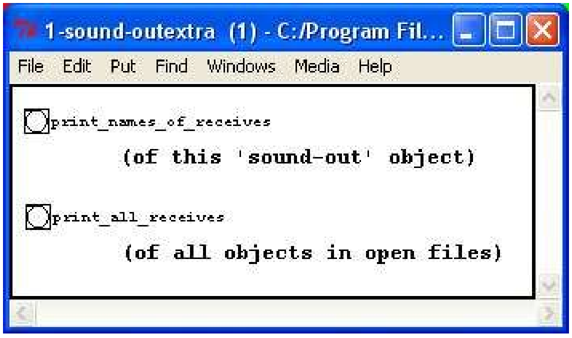
19. If you want to **automatize or control** a certain fader in an Abunch object A by another object B, you should send the control values from B to the right inlet of A. To know which inlet is used to control a fader you need to look at the help file of the Abunch object A (see 20). Also if you want to connect a clock signal to a certain object you have to look at the help file to know which inlet is fit for the clock signal. The range of the control values in Abunch is normalized between 0 and 127. *(also look at ' ex1m-control.pd' in the Abunch folder)*

20. You can open the **help file** by right-clicking ('ctrl' + click on Mac) on an object and choosing help. The help file gives you more information about this particular object. A help file in Abunch is always structured in the same way: a the top there is a general description of the object, next there is a part labelled 'panel' about the function of every fader, knob or switch in the object, next a part with the title 'inlet' about what kind of data the in- and outlets send or receive and at last a part with

extra remarks. You can find out which inlet to use to control a certain fader or knob or to receive a clock signal in the 'inlet' part.

21. If you want to see an **overview of all the objects in Abunch,** you can open the file 'objects.htm' with your internet browser. This file is in the 'doc' folder of the Abunch folder.

22. Every 'graphical user interface' element (faders, knobs,...) in Pd has a name to which you can send values, this is called a 'receive' name. Sometimes it can be handy to use these **receives** because some objects in Abunch (like [8band-filter]) have so many GUI elements that it is impossible to control all of them via inlets. If you also want to use the [text-score] object in Abunch, you need these receives because you have to insert them to declare the destination of the values in your score file. In almost all Abunch objects there is a button '**print_receives**' that prints out all the names of these receives in the main window of Pd. This button is in the 'extra_options' window of the Abunch object.


1-sound-outextra (1) - C:/Program Fil...
File Edit Put Find Windows Media Help

○print_names_of_receives

(of this 'sound-out' object)

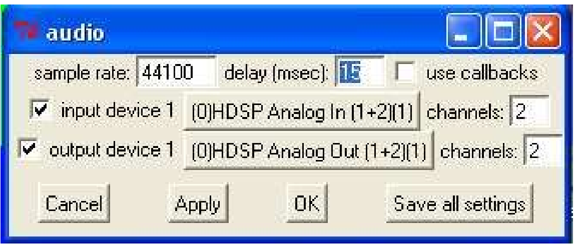
○print_all_receives

(of all objects in open files)

'extra_options' window of the 'sound-out' object in Abunch

23. If you send a value directly to the **receive** of a fader or other GUI element, you can't use the normalized range of 0 - 127. This range can only be applied if you control an object via the inlets (see 18.). Therefor the **range** is mentioned between brackets if you print the receives (see 21.). This could be a possible outprint in the Pd main window: level[0-1]: 1-samrlev. This means that you can send values within the range 0 to 1 to the fader with the panel name 'level', the receive name to which you send these values is '1-samrlev'. Some objects have more parameters that you can control with this receives system than GUI elements. These 'receives' are **hidden.** If you have a look at the outprint of the Abunch object 'waves', you will read: (hidden)_glide_time_for_level[10-xxx]: 1-wavglamp. The 'hidden' stands for a parameter that can be manipulated with the receives system but that doesn't have a GUI, in this case it is the transition time in milliseconds to move from one position of the fader 'level' to another one. If this is set to 300, it will create smooth transitions. [10-xxx] means that you can use a value between 10 and an infinitely high number.

24. Every computer needs time to process sound. This extra time is called '**latency**', 'buffer' or 'delay'. In the main menu of Pd (Windows: in 'Media' and 'Audio Settings', Mac: in 'Pd' 'Preferences' and 'Audio settings') you can find a small input window called 'delay(msec)' that shows the latency in milliseconds. You have to double this latency time if your file is also working with sounds from the computer sound input because your computer needs this time to compute the audio at the input and at the output. So if you are using a microphone or electric guitar as input sound and you want to use a (rhythmic) delay effect in Abunch, you have to fill in the latency time (as shown in the main menu) in the 'extra_options' window (also see help file of 'simple-delay')



sample rate: 44100 delay (msec): 15 □ use callbacks

☑ input device 1 (0)HDSP Analog In (1+2)(1) channels: 2

☑ output device 1 (0)HDSP Analog Out (1+2)(1) channels: 2

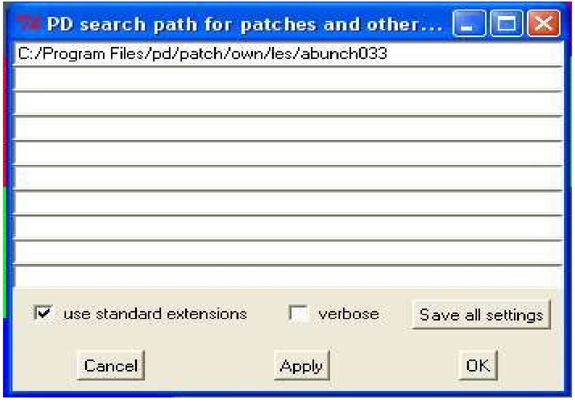
audio settings window in Pd

25. All files in the Abunch folder which names start with '**ex**' are **example files**. The 'ex1' files are simple examples with maximum 2 or 3 Abunch objects. The 'ex2' and 'ex3' files gradually increase in complexity. The example files starting with 'ex5' show how to combine Abunch objects with the built-in Pd objects. The latter are more low-level and fundamental compared to the ready-to-use Abunch objects. If you want to adjust an Abunch object to your own wishes or if you want to create a totally new object, it is handy to learn more about Pd and its basic objects.

26. If you add the Abunch folder to the '**search path**' of Pd, you don't always have to work in the Abunch folder to use these Abunch objects (as explained in 14.) You can add the Abunch file to any (new) file that is opened. Pd will automatically search for these objects in the folders of your search path. This is the way to add Abunch to the Pd search path:

In Mac Os X: in the main menu go to 'PD', 'Preferences' and 'Path' and type the location of your Abunch folder (in Pd-extended there is a button that opens the folder that you want to add). Click on 'save all settings' and OK. If you close and restart Pd the Abunch objects will automatically be found.

In Windows: in the main menu go to 'File' and 'Path' and type the location of the Abunch folder in the first empty row (like C:/Program Files/pd/patch/abunch033). Click on 'save all settings' and OK. If you close and restart Pd the Abunch objects will automatically be found.


PD search path for patches and other...

C:/Program Files/pd/patch/own/les/abunch033

☑ use standard extensions
□ verbose

27. Some of the features and procedures that were summed up in the previous remarks are typically Abunch (and don't work with any Pd object), others aren't. To avoid confusion this is an overview:

typical for Abunch:
-the red and blue background color (see 7)
-a new object needs an unique number (see 9)
-the warning windows (see 10)
-avoid 2 opened main files at the same time (see 15)
-the global preset system with the 'presets' object (see 17)
-control of GUI elements via the inlets is normalized between 0 and 127 (see 18)

standard in Pd and thus in Abunch:
-switch on audio (see 2)
-system with main windows and subwindows (see 3) and abstractions (see 13)
-inlets at the top, outles at the bottom of objects (see 4)
-thick and thin connections between objects (see 6)
-edit and play mode (see 8)
-way to connect objects (see 11 en 12)
-right-click on an object to open a help file with informaion about this object (see 20)
-system with a 'receive' name to which you can send values (see 21 en 22)
-possibility to add folders to the Pd 'search path' (see 25)

standard in Pd but not applied in Abunch:
-the name of an audio object (as opposd to a control object) ends with a ~ (inlet~, outlet~,...)

28. This is a list of some basic objecs in Pd (not Abunch) than can be easily and handily combined with Abunch objects. You can find the full list of all the built-in Pd objects by right-clicking on an empty, white place in a file and choosing help.

[spigot] : a switch: if the right inlet receives a positive number, the left inlet is passed to the outlet.

[moses] : takes numbers and outputs them at the left output if they're less than the control value, at right if they are greater or equal to it.

[select] : looks for identical numbers or words (symbols). If the left inlet is identical to the right inlet, a 'bang' message is sent to the left outlet, otherwise it is sent unchanged to the right outlet.

[<] [>] [==] : (smaller than – higher than – equal to) compares and sends a 1 (true) or 0 (false) to the outlet.

[random] : produces a random number between 0 and the number at the right inlet every time a 'bang' message enters at the left inlet.

[trigger] (or **t**) : sends out anything in a chronological order to the different outlets (right outlet first, left outlet last).

[print] : prints anything in the Pd main window.

[key] : sends the number of a key of the computer keyboard.

[keyname] : sends out a 1 to the left outlet if you press a key and a zero if you release it.
[notein] : sends out the pitch and velocity values from a midi keyboard.
[makenote] : produces a midi note with pitch, velocity and duration.
[ctlin] : (controller in) sends out the incoming (hardware) midi controller values.
[osc~] : produces a sine tone at the outlet, the left inlet is for the frequency.
[noise~] : produces noise.
[dac~] : audio outlet of the soundcard of the computer (digital to analogue converter).
[adc~] : audio inlet of the soundcard of the computer (analogue to digital converter).

29. If you click on the button 'print' in the 'presets' object, a list with all the necessary files (for the main file) is shown in the main window of Pd. This is the list of all abstractions (see 14.) and the preset file. If you place all these files (together with the main file) in a new folder, create a folder 'presets' in this folder and place the presets text file in this 'presets' folder, you can open the main file and all abstractions without needing the whole Abunch folder.

30. Some Abunch objects weren't originally developed by the author Hans Roels but are **adaptations of existing files**:

phaser : based on H15.phaser.pd by Miller Puckette (in the Pd doc files)
reverb : based on rev3~ .pd by Miller Puckette (in the Pd extra folder)
simple-pitchshift : based on G09.pitchshift.pd by Miller Puckette (in the Pd doc files)
old-vocoder : based on 5.vocoder.pd by Miller Puckette (in the Pd doc files)
al-disto : based on mbass v0.3 by Tristan Chambers
waves-add : based on 'wave-designer.pd' by Frank Barknecht
wa-synth : based on E07.evenodd.pd by Miller Puckette (in the Pd doc files)
spectrum : based on E01.spectrum.pd by Miller Puckette (in the Pd doc files)
analyse : based on sigmund~ by Miller Puckette (in the Pd extra folder)