



# User manual

## xp.source

**xp.source** is a m4l (max-for-live) device. It comes inside the *xp4l.devices* bundle under the file name *xp4l.source.amxd*.

It is probably the device that you will most often be required to use. It processes both audio signal as well as datas.

### What is a source ?

In spatial audio, a sound source is often compared to a virtual object to represent a sound which is then projected into an immersive audio environment. In XP like in other comparable approaches, the source is represented as a shape in a 3d environnement. This method makes understandable because "visible", the mechanisms of audio reproductions by immersive arrangement of loud speakers, and enable to use geometric wrapping technics to build complex soundscapes.

As a result, a sound source is that virtual object that can be moved within a geometric environnement. In XP, thanks to the Ircam Spat library, the imaginary audio signal born from these movements in the virtual scene is reconstituted using cutting-edge spatialization algorithms whose role is to calculate for each of the speakers the expected signal level according to the scene to render.

XP implementation has been designed in respect of Ableton Live spirit, offering the possibility for these sources to be normal tracks like any other Ableton track, taking audio clips, or benefiting from one of its strength, in the flexible routing it offers. Therefore, sources can take any kind of inputs, from external audio, from return tracks, or from any midi instruments. Thus, the sources of XP can be understood as expanses, like sonic upgrades in the spatial domain, of a classical Ableton live track.

### mono vs stereo

However Ableton works natively with stereo track, choice was made in xp4l to consider sound source only as mono signal. In an immersive context, the base unit is mono signal, and stereo doesn't represent an aesthetic relevance over other formats in this context which should justify prevalence of any kind.

Stereo format exists because it's used for stereo sound system. Nonetheless, If a user desires to reconstruct in its immersive set up the effect of a stereo sound, two sources should be created, each of them playing one side of that stereo sound, which means taking a wide space of the arrangement with basically the same sound.

In XP, there are probably better ways to optimize such feelings. It was conceived as an invitation to search for new forms of sonic expression within the possibilities of spatial sound design, and push back the limits of imagination imposed by stereo constrains.

## Initialization

XP comes with a standard way to instantiate devices loaded in Ableton. It's dynamic, efficient and instinctive as long the basic rule is followed.

Drop the device on an empty Ableton track and the track title will take the name of `<device-type><#id>`.

*Keep titles of the tracks the way they have been instanciated.*

## Loading

Load a `xp.source.amxd` on a new audio track. The title of that track automatically takes the available source number/ID among the 16 available (1 to 16), according to those that have already been created in the project.

Use one `xp.source` per source to control its parameters, and load each instances on a different Live track.

Adding a second `xp.source` device, will entitle the second track 'source2'. And so on.

A maximum of 16 Live tracks shall be used for all of the maximum source number allowed.

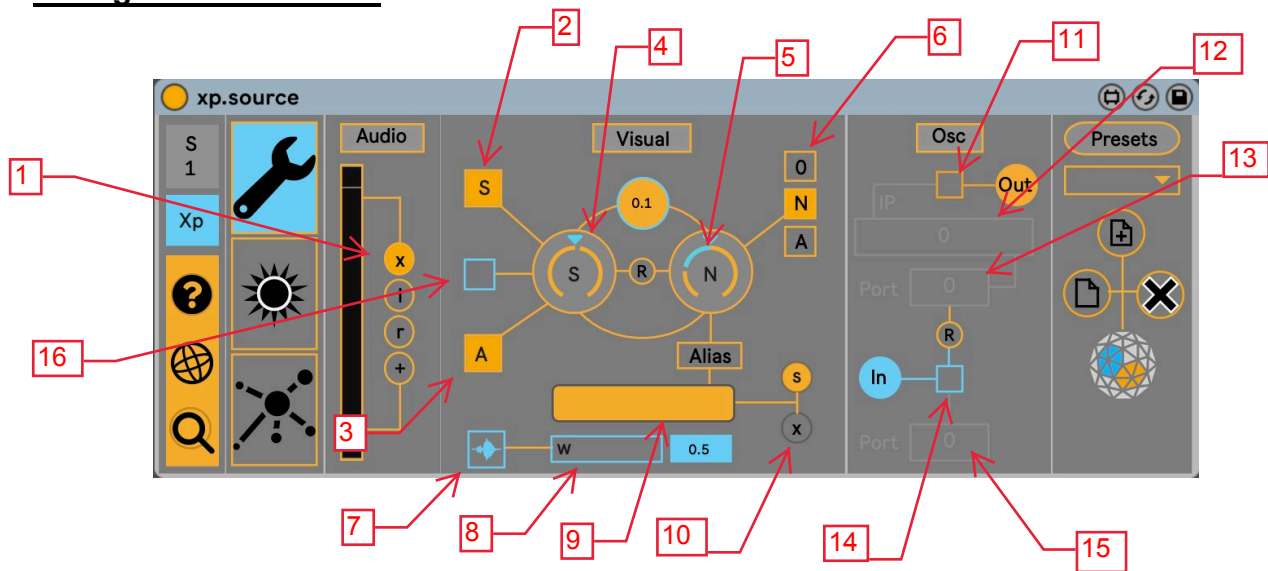
Each time a `xp.source` is loaded on a new track, a sphere appears in the XP standalone view, corresponding to that sound source initialized and positioned in the 3d space.

By default, newly created source are placed at az 0 elev 0 dist 1 (or x 0 y 1 z 0) and signal is muted in the signal interface (read signal interface bellow). A source signal created is sent to the `xp.engine` as an audio channel whose number corresponds to the source id. For instance, `source3` is sent as `input3` in the `xp.engine`, and its signal modulation can be observed in the `xp.engine` input monitor view.

Create as many as 16 sources allowed and observe them in the 3d view. As soon they are correctly instantiated in the view, all source parameters are working in the `xp.source` interfaces.

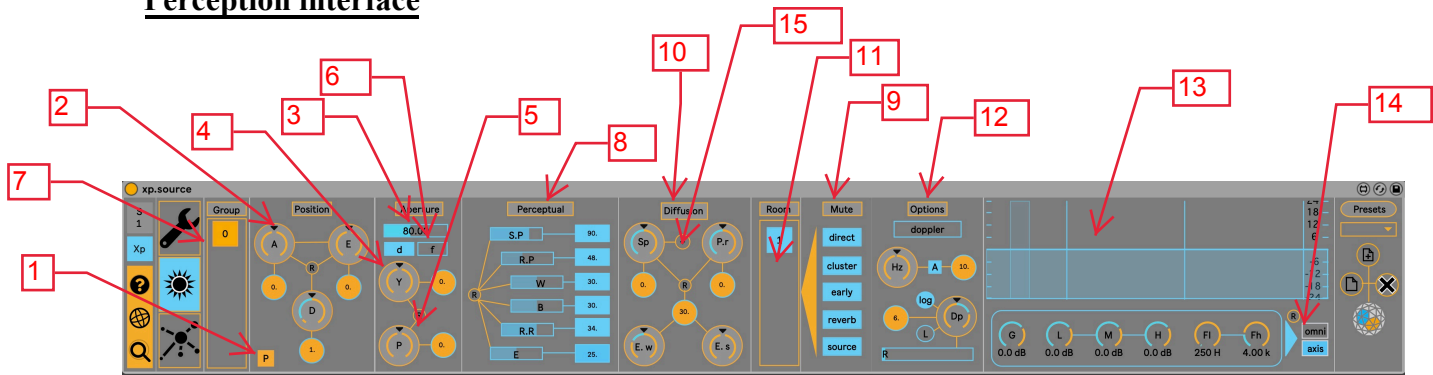
`xp.source` displays a navigation menu on its left to navigate through each of its interfaces : Configuration, Perception, Animation

## Configuration Interface



- 1: select audio channel (off, left, right, sum)
- 2: show/hide source
- 3: show/hide aperture
- 4: sets source appearance size
- 5: sets name appearance size
- 6: select name to display (off, source name, alias)
- 7: show/hide wave form
- 8: amplifying coefficient for wave form
- 9: sets alias for source appearance
- 10: set/reset alias
- 11: Osc out enable
- 12: IP address for osc out
- 13: IP's port for osc out
- 14: Osc in enable
- 15: Port in for for osc in
- 16: source appearance color

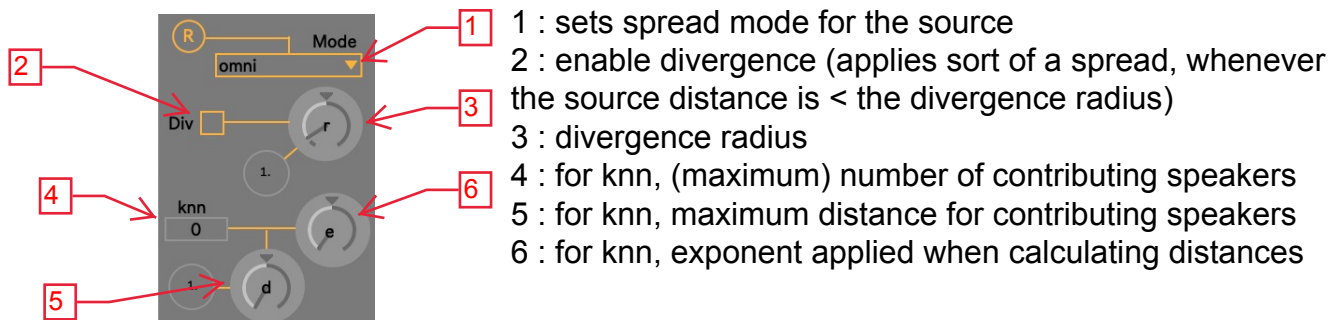
## Perception interface



This interface gives access to parameters adjusting the source perception in its immersive environment. It is a collection of parameters which address simultaneously algorithm configuration of the *Ircam Spat* as well as to the geometric domain in *XP standalone*. As this interface addresses specifically source perception as conceptualized by the *Ircam spat*, new users of that library are strongly recommended to read *Spat* documentation, which provides a comprehensive description for each of the parameters mentioned below.

## Parameters reference

- 1 : Change between Polar and Cartesian coordinate system (aed, xyz)
- 2 : Coordinates parameters value. Move the source position in the 3d space.
- 3 : Aperture of source. Represented in the view by the wire frame shell surrounding the spherical shape
- 4 : Yaw. Source diffusion rotation angle on its z axis
- 5 : Pitch. Source diffusion rotation angle on its y axis
- 6 : Orientation (Default: toward central point / Fixed: disable toward central point)
- 7 : Group. Assign source to a group (see [xp4l.group\\_doc.pdf](#))
- 8 : Perceptual factors : The 6 *Ircam Spat* perceptual factors
- 9 : Mute section.
- 10 : Diffusion section. Spread, Pan Rev, Early width, Early shape.
- 11 : Rooms. Assign source to a Room (see [xp4l.room\\_doc.pdf](#))
- 12 : Option section. Doppler, Air absorption, Air Frequency, Drop, Drop model, Radius.
- 13 : Filtering section. Axis and Omni equalizer.
- 14 : Change between Axis and Omni equalizer
- 15 : show/hide optional diffusion panel (see below)



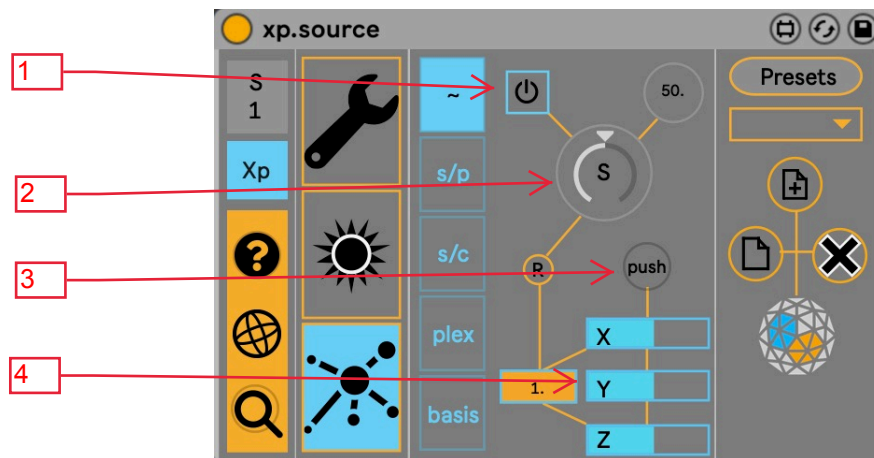
- 1 : sets spread mode for the source
- 2 : enable divergence (applies sort of a spread, whenever the source distance is < the divergence radius)
- 3 : divergence radius
- 4 : for knn, (maximum) number of contributing speakers
- 5 : for knn, maximum distance for contributing speakers
- 6 : for knn, exponent applied when calculating distances

## Animation interface

If moving a source in space should not be the *alpha and omega* of spatial sound design, the fact remains that the animation of sources in 3d space is a critical task that can quickly turn out to become necessary for the production of dynamic creative content.

In *XP*, we wanted to provide several solutions to users so that they are very quickly able to animate their scene while remaining in the instinctive and creative workflow of Ableton. Source can be animated through traditional automation from Clip or arrangement panel. They can also be controlled live from external devices with assignments to the desired parameters. And also, on top of it, with *xp.group*, *XP* comes with an additional unique approach which is an opengl based generative engine accessible from this interface. It offers 4 options that covers a wide range of solutions to quickly animate sound source in space

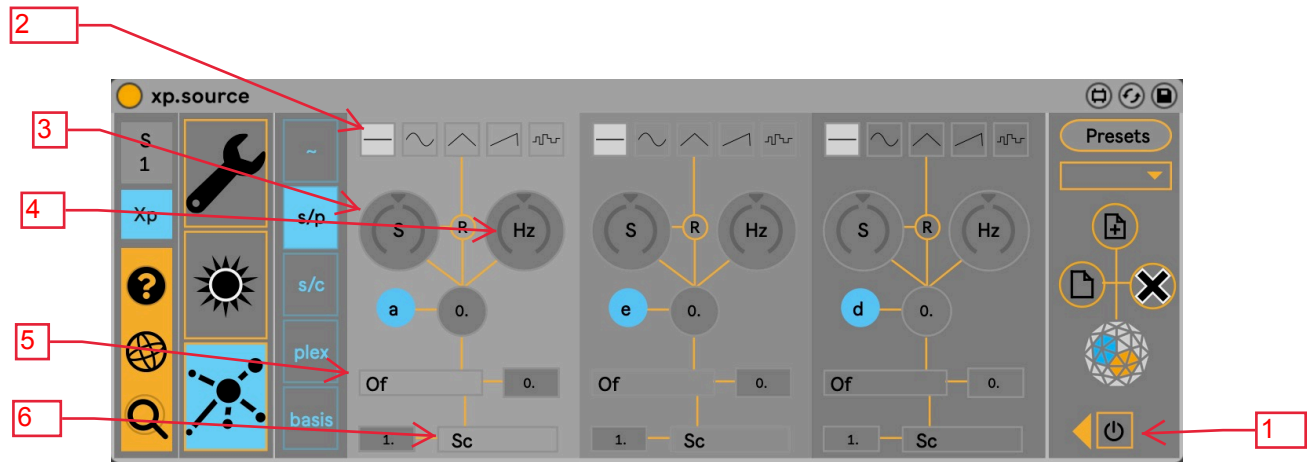
~ :Noise



It works as a noise based randomizer on each xyz dimension

- 1: activate the function
- 2 : speed of the randomize function
- 3: one shot random generator
- 4: factor multiplier for x,y,z dimension

s/p (polar) and s/c (cartesian)



Simple is named this way because it uses a simple generative function based to output float time value controlling coordinates parameters of source. Nonetheless it can already cover a large spectrum of function based motions and trajectories, like for the most known circular orbits. Most of the time at least two dimension should be activated to enable animation.

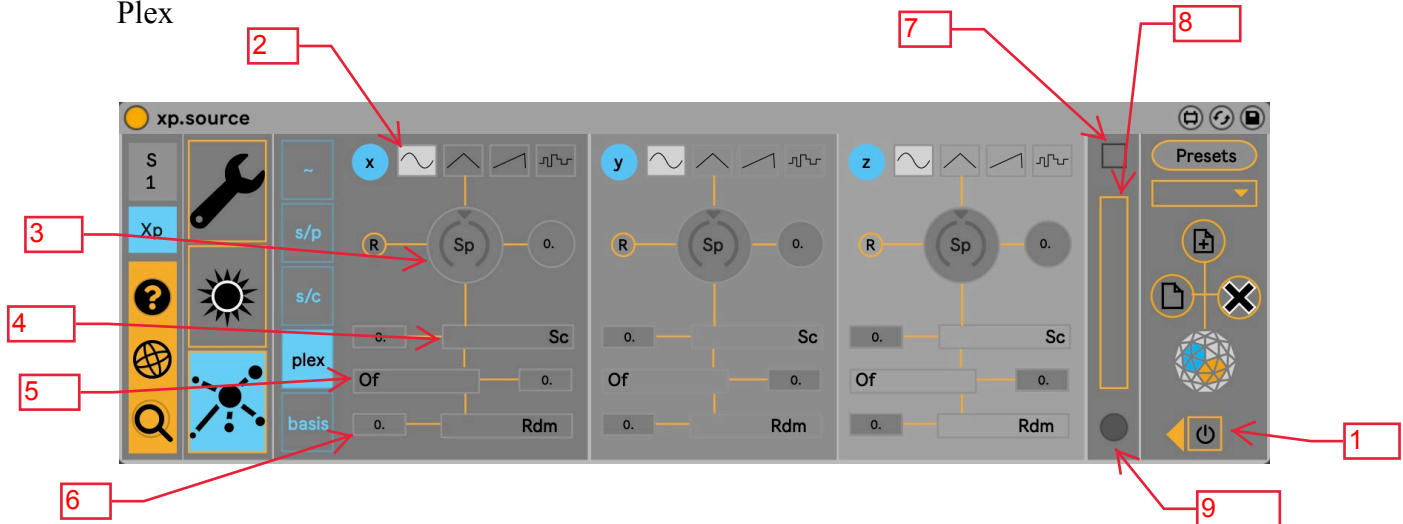
The interface is composed of each dimensions, whether it is set for polar system with Az, Elev, Dist, or for Cartesian with XYZ.

Parameter reference bellow covers only one dimension, but the same applies for each of them

#### Parameters reference:

- 1 : enable function
- 2 : Choose a procedural function to generate floats among Line, Sin, Tri, Saw, Perlin
- 3 : Speed of the function
- 4 : Frequency of the function
- 5 : Offset the function
- 6 : Scale the function

Plex



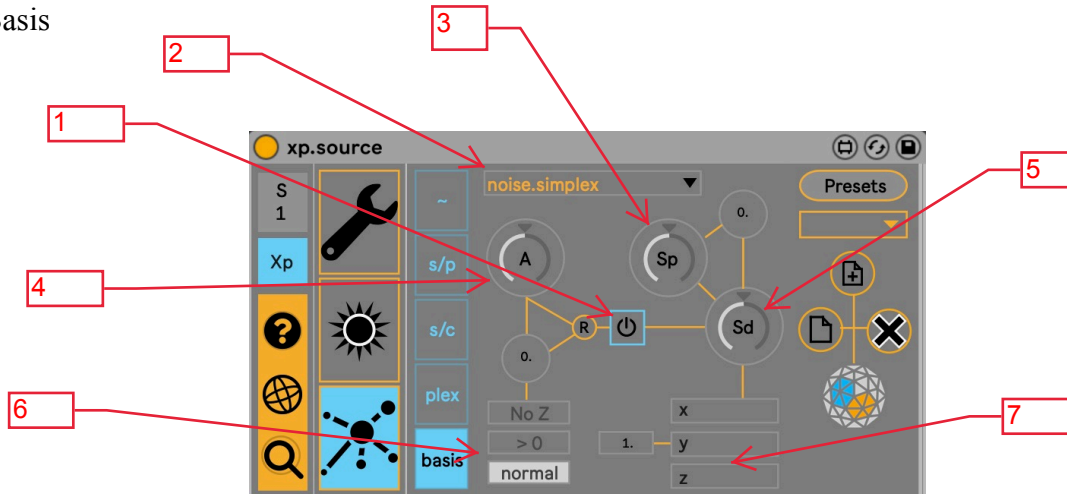
Without getting too much in details, although it can seems similar to simple, it can achieve much more variations than this latter whether in the speed that the sources take, their progression and of course the forms followed.

It has moreover a unique specificity which makes it possible to visualize in the 3d scene, the generated virtual path on which a source is progressing.

#### Parameters reference:

- 1 : Activate the function
- 2 : Choose a procedural function to generate floats among Sin, Tri, Saw, Perlin
- 3 : Speed of the function
- 4 : Scale of the function
- 5 : Offset of the function
- 6 : Random amount of the function
- 7 : Display procedural path
- 8 : Line width of the path
- 9 : Color of the path

## Basis



Behind its false semblance of simplicity, this mode is the one that offers the most fun as it allows sources to take unexpected movement. It uses basis procedural function to evaluate coordinates of the sources.

The special of this mode is to allow to apply constraints to the movement of sources, by preventing them from negative values in the z, or to enter the listening zone. Further updates might include such constrains for the other modes.

### Parameters reference

- 1: activate function
- 2: choose among basis function
- 3: speed of the function
- 4: amplifier coefficient
- 5: pseudo random seed value
- 6: rules, no z (no negative value applied for z dimension), >0 (source is not entering within listening area), normal (no rules)
- 7: scaling weight factors for each dimension