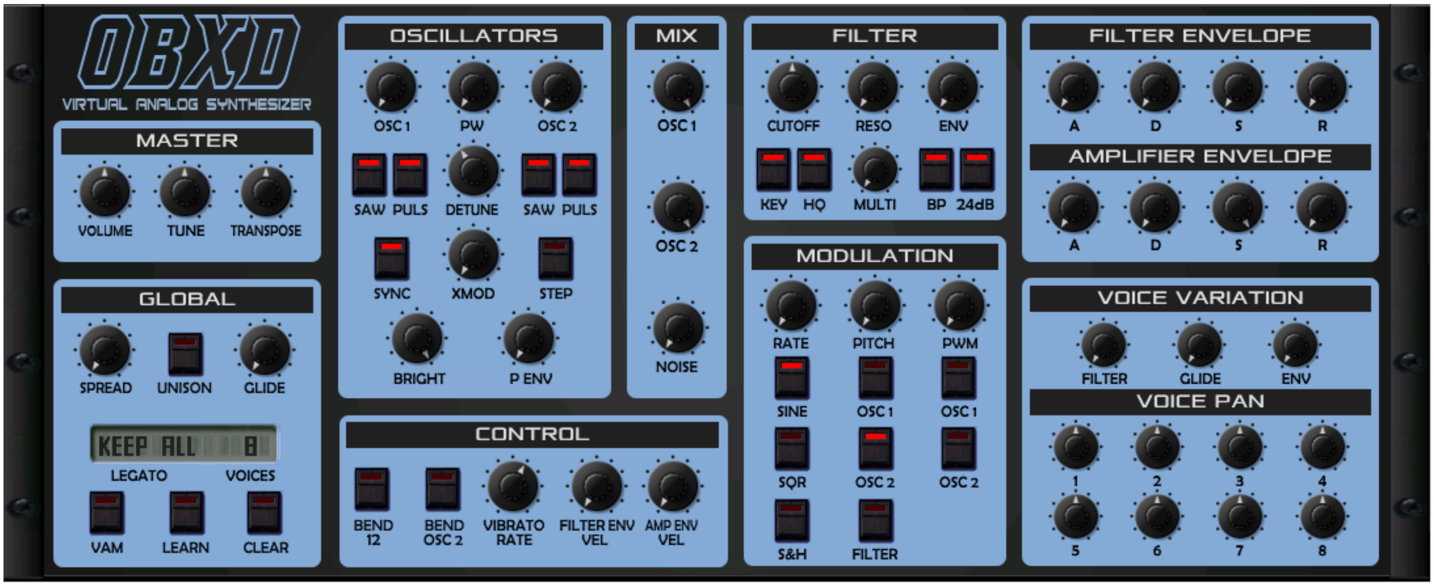
The Datsounds Obxd Virtual Synthesizer

An unofficial manual



Introduction

The Obxd is based on the Oberheim OB-X. It attempts to recreate its sound and behaviors, but as the original was very limited in some important ways a number of things were added or altered to the original design. If you're unfamiliar with the OB-X, its user manual can easily be found from various sources on the net. This manual will make no attempt to explain basic synth programming or the operation of the OB-X, but will discuss modifications from the basic design and in some cases their ramifications.

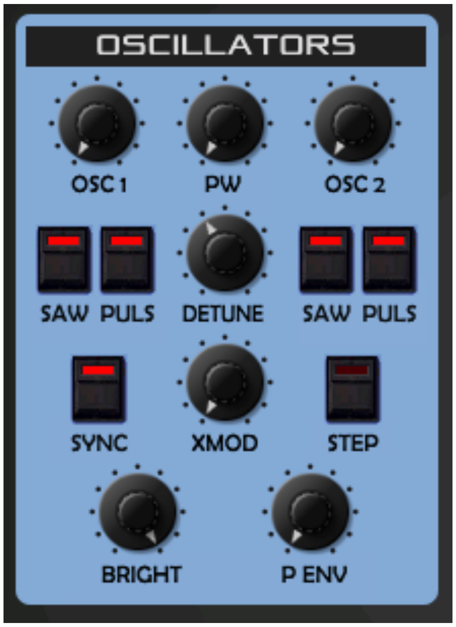
The Obxd was designed to sound as good and as rich as the original. It implements micro random detuning which is a big part of that sound. However, it was not designed as a self-contained completely independent soft-synth. It needs to be contained within a VST framework where things like transposition, automation, layering, arpeggiation, etc, are available. Also, like many synths of the OB-X's generation, the Obxd has no internal effects so its sounds and textures can be greatly enhanced by the use of additional processing like chorus, reverb, delay, etc...

Thanks to 2DaT who created this little marvel and to Layzer who is the primary designer of its GUI. But we would be remiss not to thank all the members of KVR who participated in the Obxd forum thread! While the heart and soul of its design and development is all 2DaT, much discussion and debate went into it's graphic layout and the implementation of some of its features.

So thanks to all who participated in its creation and also to the various people who stepped up to create excellent free patches for the Obxd.

Breeze@KVR - 2014-02-09-V1.11
Note: operation description conforms to the latest Obxd build as of the above date.

Oscillators:



- A mixer was introduced to blend the two oscillators and noise source which is much more flexible than the fixed levels of the original design.

- In the OB-X cross modulation (Frequency Modulation), OSC2 modulated OSC1. On the Obxd this is reversed: OSC1 modulates OSC2. Sounds created by sync and x-mod this way were judged more useful than the inharmonic results of the original modulation path.

- The Step switch allows for precise tuning by semi-tone steps. Note that disengaging this doesn't keep the pitch at the precise semi-tone step but resumes gradual pitch adjustment.

- The Bright control adjusts the clarity of higher harmonics.

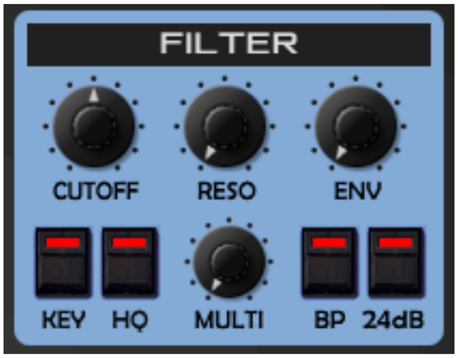
- Detune of OSC2 only adjusts up from the basic pitch (the original was bilateral).

Oscillator related controls (in the Global section)

- **TRANSPOSE** : the transpose control in the Master section only transposes both oscillator pitches in parallel and by octaves. It does not alter the filter cutoff frequency, which allows the filter to change it's harmonic center relative to the fundamental frequency of the oscillators. This is consistent with the original OB-X design. Note that there is no built-in way to transpose the Obxd by steps; when a sound is created and is out of step with concert pitch, the Tune control in the Master section can help tune to the nearest concert note, but to play in pitch external MIDI will have to be transposed.

- **SPREAD** : this control detunes every single oscillator by a random amount, simulating the sound of slightly detuned VCOs (unlike the precision of DCOs).

Filter



The original OB-X had a single 12dB/octave low-pass filter. The Obxd significantly improves on this design:

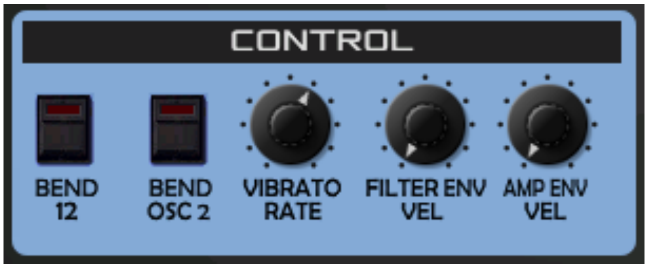
- It implements a Multimode 12dB/octave filter like the Oberheim SEM module. Using the **MULTI** rotary control, you can crossfade between low-pass behavior on the complete left, to either notch (by default) or bandpass (using the BP switch) behavior at 12 o'clock, and high-pass behavior on the far right.

- There is also a 24dB mode which is activated by the 24dB switch. This mode is only low-pass, but the **MULTI** control now allows you to smoothly change the depth of the filter slope from 24dB/octave on the complete left, down to 6 dB/octave on the complete right.

- The **BP** switch has no effect in 24dB mode. It turns on and off to give you something to play with on the GUI which doesn't affect the sound in any way. ;)

- The **HQ** button on the filter turns on "High Quality" mode which results in better interpolation and smoother processing of higher frequency components at the expense of CPU consumption.

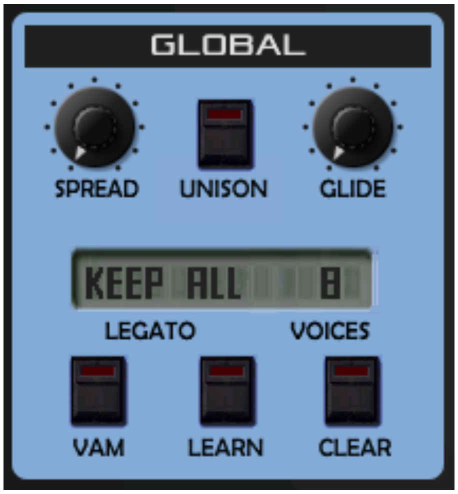
Control Section



- **VIBRATO RATE**: On the original OB-X the Vibrato Depth is assigned to the Modulation Paddle by default. Similarly, on the Obxd Vibrato Depth is inexorably tied to the Modulation wheel as well (CC#1). In order to allow assignment of the Modulation wheel to other parameters without turning on vibrato, turning the Vibrato control to the complete left effectively turns off the vibrato. The vibrato control can be remapped to another controller to control its rate, but CC#1 will ALWAYS affect vibrato depth.

- **FLT ENV VELO** and **AMP ENV VELO** (Filter and Envelope Velocity Modulation) : the original OB-X was not velocity sensitive in any way. The Obxd allows note velocity to alter the envelope depth of the Filter and Amplifier envelopes. Note that altering the filter envelope by velocity will also cause variations in the Pitch when modulating OSC2 (P ENV) from the Filter envelope.

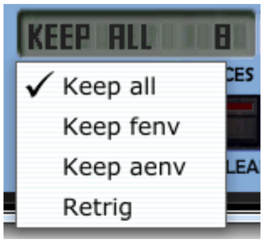
Global Section



- **VAM** (Voice Allocation Mode) : By default the Obxd is a low-note priority synth (like the OB-X) but the VAM switch changes this to last-note priority.

- **VOICES**: unlike the original OB-X, you can select the number of active voices. This can be a CPU saving measure as much it can be a creative choice. Note that CPU use scales linearly as the number of voices is increased.

- **UNISON**: a single-note mode that stacks all the voices available. Spread and Voice Panning are assets in this mode.



●**LEGATO**: there are four behaviors allowed for legato playing. Releasing notes over the maximum number of available voices reactivates the notes still being held, in a manner based on the Legato style.

Note: in all cases, envelopes are only re-triggered once they've reached their sustain phase. This is more noticeable with slow attacks and decays.

Keep all:

• This mode keeps playing notes at their sustain envelope levels;

• Additional notes above the max # of voices will steal held notes, but no envelopes will be re-triggered;

• Exceeding polyphony (max # of voices set in VOICES) and then letting go of these additional notes replays the equivalent number of held notes (based on note history) at the sustain level for the patch (envs not re-triggered).

Keep fenv:

- “Keep Filter Envelope”
- This mode will keep held notes at the Filter envelope sustain level;
- Additional notes above the max # of voices will steal held notes and retrigger the Amp envelope only.
- Exceeding polyphony (max # of voices set in VOICES) and then letting go of these additional notes replays the equivalent number of held notes (based on note history) and only re-triggers the amplifier envelope.

Keep aenv:

- “Keep Amplifier Envelope”
- Same as “Keep fenv” but the Filter envelope is re-triggered while the Amp envelope is not.

Retrig: Same as “fenv” and “aenv” but now both envelopes are re-triggered on new notes.

Note: best way to understand these modes is to reduce the polyphony to 2 or 3 notes and try them.

- **LEARN** and **CLEAR** : Used to bind and unbind Obxd controls to MIDI CC# for automation.

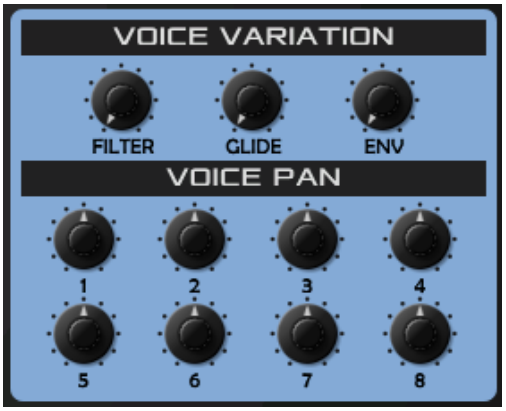
Note: the Obxd does not have it’s own internal automation map, but you can save current assignments by saving a patch bank (*.fxb); this way any assignments you make can be saved and recalled between projects and hosts. Any automation you assign in your DAW will be correctly remembered by your VST host and re-established on project reload.

LEARN: click on Learn, turn or activate the control you want to automate (touching is not enough), and then send the CC message you want to use for it; the LEARN LED will turn off and the controller will be assigned.

CLEAR: to make an Obxd control forget its automation, enable CLEAR, turn the control and send it the original automation it was assigned to; the LED will turn off & the assignment will be forgotten.

Note: it’s possible to re-assign a new controller to a previously assigned control by performing LEARN and sending it a new CC#.

Voice Variation & Panning



The more variation is applied with Voice Variation, the more these parameters will randomly stray from their predictable settings on repeated execution and introduce more analog unpredictability.

Each voice can be panned, but now right on the front panel. :)

FAQ

Q: Where is the FAQ?
A: There is no FAQ.

;)