OB-Xf Manual

Surge Synth Team

Contents

Getting Started

OB-Xf is a virtual synthesizer inspired by the venerable Oberheim OB-X, OB-Xa and OB-8 synthesizers. It is a continuation and modernatization of the last open source release of OB-Xd by 2DaT and later discoDSP.

OB-Xf is free and open source software, which you can use for your music in whatever form you see fit. It is available as a CLAP, VST3, AUv2, and standalone application.

To ask questions, report bugs or issues, or help develop OB-Xf, visit the Surge Synth Team Discord server.

Images in this manual use the **Default** theme.

A PDF version of this manual is also available for download.

Installing or Building OB-Xf

Installing OB-Xf

It is as easy as downloading the installer from the product page and running it!

Windows

On Windows, the installer provides options for installing the VST3 and CLAP formats, as well as standalone application.

Installer is available for both x86 and ARM64/ARM64EC platforms, separately.

macOS

On macOS, the installer provides options for installing the AU (v2), VST3 and CLAP formats, as well as standalone application.

macOS 10.15 or later is required.

Linux

On Linux, the Debian package provides options for installing the VST3 and CLAP formats, as well as standalone application. Individual components are available packaged in a ZIP file

The distribution package is built on Ubuntu 22.04. A list of required packages is listed in the source code and in the .deb itself.

Standalone

If you are running OB-Xf as a standalone application, we would like to divert your attention to the Options menu in the top-left corner, which provides:

- Audio/MIDI Settings in order to specify audio device, sample rate, MIDI input(s) and audio output(s) to be used
- Save and load state of the synthesizer (also known as “presets” or “patches”, although OB-Xf provides its own patch management, which is recommended to be used instead)
- Reset OB-Xf to the default state (reinstantiates the whole synthesizer)

Building from Source Code

To build OB-Xf from source code, steps are outlined in the README of the GitHub repository.

User Interface

OB-Xf user interface is seperated into the following sections: - Programmer - Master - Global - Oscillators - Control - Mixer - Filter - LFO - Filter Envelope - Amplifier Envelope - Voice Variation

Control Types

OB-Xf parameters can be controlled with knobs, toggle buttons, multi-state buttons, menus and text entries.

Knobs

- **Left click** and dragging vertically moves the knob.
- Holding **Shift** while dragging vertically moves the knob with much higher precision.
- **Mousewheel** moves the knob in steps.
- Holding **Shift** while operating the mousewheel moves the slider in small steps.
- For knobs pertaining to pitch adjustment, holding **Alt (⌘ on macOS)** will jump in predefined intervals (4th, 5th, octave, and their multiples).
- **Hovering** above the knob will uncover a popup showing the parameter's value.
- **Double-click** resets the parameter to its default value.
- **Right-click** displays the parameter's context menu

Knob Context Menu The knob context menu provides the full name of the parameter, a text entry field displaying the parameter value (which is also automatically selected and awaiting text input), and and option to reset the parameter to its default value.

Pan knobs in Voice Variation section provide additional options. More about that in Voice Variation section.

Certain hosts can provide additional context menu options for VST3 and CLAP formats, which would show here. These options are usually related to parameter automation, MIDI learn, and so on.

Toggle and Multi-state Buttons

These controls are as straightforward as they can be. Left click will change their value, the only difference is that multi-state buttons have more states than just on and off. They are signified either by existence of two LEDs on them, or by rotating through multiple colors, as showcased below.

Menus

These controls provide multiple options through a context menu, and are stylized to look like 14-segment LCDs for that ultimate vintage synthesizer vibe.

Simply click on the LCD and the options will show!

Text Entry

These controls allow you to input text, which is crucial for specifying important patch information like name, author and license.

Programmer

Patch Organization

Patches are organized in two main overarching sections: Factory and User. To reveal the menu-based patch browser, click on the **Browse** menu - it doubles as patch number, too!

This browser follows the folder structure as-is on your hard drive. For Factory patches, a folder carries the same meaning as sound category. For User patches, an additional folder level is allowed which is called a Project.

An alternative method to browse patches is also available by means of 16 direct access buttons and the **Group** button. This allows browsing in currently focused folder (or rather, sound category) in 16 groups of 16 patches each, effectively allowing quick access to up to 256 patches. A folder can of course contain more than 256 patches, in this case the 16 direct access buttons will only be able to reach the first 256.

Selecting a Patch or a Group

Each of the 16 direct access buttons has two LEDs on them - the yellow LED corresponds to the currently selected group, while the red LED corresponds to the currently selected patch from the currently selected group.

To select a group instead of a patch using the direct access buttons, click on the **Group** button first. Its LED will light up yellow, indicating that you are now in group selection mode. Click the **Group** button again to go back to patch selection mode.

Patch Browser

As mentioned earlier, clicking on the **Browse** menu will open the patch browser, which will outline Factory and User patches, as well as a few housekeeping functions described below:

- **Initialize Patch:** Resets the current state of the synthesizer to the internally specified initial patch, which will output a basic, unfiltered sawtooth waveform. Also available via the **Init** button directly on the user interface.
- **Load Patch…:** Opens an OS dialog for loading `.fxp` files created by OB-Xf. It is also possible to load `.fxp` files created by OB-Xf simply by drag-and-dropping them onto the OB-Xf user interface.
- **Save Patch…:** Opens a dialog for loading `.fxp` files created by OB-Xf. Also available via the **Save** button directly on the user interface.
- **Delete Patch:** Deletes the currently selected User patch and refreshes the Patch Browser. Note that Factory patches cannot be deleted, an error dialog will remind you about this!
- **Copy Patch:** Copies the current state of the synthesizer into the clipboard.
- **Paste Patch:** Pastes the synthesizer state stored in the clipboard, as the current state of the synthesizer.

- **Refresh Patch Browser:** In case you've brought in additional patches from third parties *while* OB-Xf was running, you will need to execute this command in order to be able to browse through those patches.

Save Patch Dialog

This dialog allows you to specify the **Name** of your patch, your own name as patch **Author**, sound **Category** of the patch, which **License** you'd like to assign to your patch, and the **Project** to which your patch belongs to (entirely optionally).

Project is simply an additional folder level inside of which you can store your patches, with categorization by sound type, or without, as you wish. For example, you are working on different songs, you may want to name your project according to the different song names. Or, you are preparing a live gig and you don't want to mix those patches up with any of your produced songs. Or, you have several very different live gigs going on, and you want to group patches according to those. There are likely many more possible usecases for this field, and you're in control here.

Category menu offers a fixed list of sound categories to choose from, but also allows none to be selected. In this case, no subfolder will be created when saving the patch.

Other Facilities

- **Patch Name:** Text entry field which displays the current patch name, **left clicking** it will allow renaming.
- **Prev/Next:** These buttons allow you to cycle through all available patches sequentially.
- **Undo:** Backtracks through your parameter edits. Note that loading a different patch is not directly undoable (all parameter changes which happen upon a patch change will become traversable undo steps)!
- **Init:** Initializes the current patch to a basic saw wave sound.
- **Rand:** Mildly randomizes almost all parameters of the synthesizer, which can lead to unexpected results. Note that Master section, Polyphony, High Quality Unison Voices parameters are **not** randomized!

Master

The Master section simply contains 3 parameters:

- **Volume:** Controls the final output level of OB-Xf. This is very useful for balancing volumes between different patches.
- **Transpose:** Transposes the pitch of all incoming notes in range from -24 to +24 semitones.
- **Tune:** Fine-tunes the pitch of all played notes in range from -100 to +100 cents.

Global

The Global section contains parameters that affect the entire synthesizer.

Poly

This value sets the maximum number of voices that can be played simultaneously, from 1 to 32.

Unison

Enables the layering of multiple voices per note. The number of voices is set by the (Unison) Voices parameter.

(Unison) Voices

This value controls the number of voices played per note, from 1 to 32.

- Only enabled when Unison is switched on.
- Higher values create a thicker sound, but will reduce resulting polyphony.

Detune

Adjusts the amount of pitch variation between voices from 0 to 100 cents. Increasing this control adds imperfection in the tuning between notes, better representing an analog synth that is not perfectly (or desperatly needs to be!) calibrated. Particularly effective when Unison is enabled.

HQ (High Quality)

Enables 2x oversampling of the whole voice structure. This improves audio quality at the cost of increased CPU usage.

Glide

Sets the duration it takes for the pitch to continuously glide from one note to another.

- The pitch gliding approximatively follows a logarithmic curve.
- This value can be set to vary per voice with its associated control in the Voice Variation section.

Env(elope) Legato

Sets which envelopes are going to continue from their current position (i.e. they will NOT be retriggered) when playing legato (overlapping) notes:

- **Both (Both Envelopes):** Both envelopes will continue.
- **Filter (Filter Envelope Only):** Only the Filter Envelope will continue, Amp envelope will retrigger.
- **Amp (Amplifier Envelope Only):** Only the Amp Envelope is will continue, Filter envelope will retrigger.
- **Retrig (Always Retrigger):** Both envelopes are retriggered.

Note Priority

Sets which note is prioritized either when the maximum number of voices is reached, or when Poly is set to the same value as (Unison) Voices (which is effectively monophonic mode):

- **Last:** The most recently played note is prioritized.
- **Low:** The lowest note is prioritized.
- **High:** The highest note is prioritized.

Learn

Enables MIDI learn mode. When enabled, clicking on a parameter will focus it for receiving a MIDI CC message. Once you tweak your wanted MIDI controller, you can click on another parameter to repeat the procedure. If you want to remove an assignment, click the X button to the left of the MIDI CC number shown.

Menu

This button opens a menu with the following options:

File

- **Initialize Patch**: Resets the synthesizer to its default state.
- **Import Patch**: Opens a file dialog to import a patch from a `.fxp` file.
- **Import Bank**: Opens a file dialog to import a bank of patches from a `.fxb` file.
- **Export Patch**: Opens a file dialog to export the current patch to a `.fxp` file.
- **Export Bank**: Opens a file dialog to export the current bank of patches to a `.fxb` file.
- **Copy Patch**: Copies the current patch to the clipboard.
- **Paste Patch**: Pastes a patch from the clipboard to the synthesizer.
- **Open User Directory**: Opens the user directory where patches, banks and themes are stored.

MIDI Mapping

- **Clear MIDI Mapping**: Clears all MIDI CC assignments.
- **Save MIDI Mapping…**: Stores the current MIDI CC assignments to a file for later recall.

Themes

This submenu contains a list of all available Factory and User themes. User themes can be added to the user directory (see Open User Directory above).

Zoom

This submenu contains options to adjust the zoom level of the interface, from 75% to 400%. Some zoom settings may not be available depending on your screen resolution. There is also a submenu containing options for context menu scaling.

Control

The Control section contains a few options that affect how OB-Xf responds to certain incoming MIDI data.

Pitch Bend

- **Down** and **Up:** These menus set the pitch bend amount in semitones, from 0 to 48, separately for positive and negative values of MIDI Pitch Bend messages.
- **Osc 2 Only:** Toggling this button makes the pitch bend wheel affect only Oscillator 2. This can be very useful for hard sync sounds.

Vibrato

This is a dedicated global LFO that modulates the pitch of both oscillators across all voices simultaneously. Vibrato amount is controlled by the modulation wheel (MIDI CC #1).

- **Rate:** Sets the speed of the vibrato LFO.
- **Shape:** Sets the LFO waveform to a triangle or a unipolar square.

Oscillators

OB-Xf contains two independent oscillators, each with its own waveform, pitch, and modulation options.

Tuning

- **Osc 1:** Controls the coarse tuning of Oscillator 1, from -24.00 to 24.00 semitones.
- **Detune:** Controls the fine tuning of Oscillator 2 only, from 0.1 to 100 cents. Like on any analog synth, the two oscillators will never be perfectly in tune with each other, which is why this control won't go down to 0 cents.
- **Osc 2:** Controls the coarse tuning of Oscillator 2, from -24.00 to 24.00 semitones.

Waveform

Two buttons govern the waveform selection for each oscillator. Available options are **Sawtooth** and **Pulse**. If neither of these buttons are enabled, the oscillator outputs a **Triangle** wave, and the triangle wave icon between the two buttons is highlighted.

Pulse Width

- **PW:** Controls the pulse width of Pulse waveform for both oscillators. It is applicable if either oscillator is set to a pulse wave. It ranges from 50% to 97.5%.
- **OSC 2 Offset:** Additionally offsets the pulse width of Oscillator 2 relative to Oscillator 1. It ranges from 50% to 97.5%.

Crossmod and Sync

- **Crossmod:** Adjusts the amount of frequency modulation of Oscillator 2 by the output of Oscillator 1.
- **Sync:** Enables hard synchronization of Oscillator 2 to Oscillator 1. Effectively, the waveform of Oscillator 2 is restarted every time Oscillator 1 is itself restarted (waveform begins a new cycle), creating a more complex waveform.

Bright

This control adjusts the emphasis of high-frequency harmonics in the output of both oscillators. This can add presence and clarity to the sound.

Filter Envelope Modulations

- **Env to Pitch:** Adjust the amount of modulation applied from the Filter Envelope to the pitch of Oscillator 2 only (if **OSC 1+2** is disabled). It ranges from 0.00 to 36.00 semitones.
- **Env to PW:** Adjust the amount of modulation applied from the Filter Envelope to the pulse width of Oscillator 2 only (if **OSC 1+2** is disabled).

Inv(ert) and Osc 1+2

These parameters are available for both Env to Pitch and Env to PW modulations.

- **Inv:** Inverts the polarity of the **Env to Pitch** and **Env to PW** modulations.
- **Osc 1+2:** When enabled, the modulation is applied to both oscillators equallty. When disabled, modulation is applied to Oscillator 2 only.

Mixer

The Mixer section contains the different volume options for the oscillators, ring modulation and noise generator.

OSC 1 and OSC 2

These two knobs control the output level of Oscillator 1 and Oscillator 2 respectively.

Ring Mod

This knob controls the output level for the ring modulation of Oscillator 1 by Oscillator 2. This effect can be useful to create moving or metallic sounds.

Noise

This knob controls the output level of the noise generator. The type of noise can be selected via the multi-state button next to it: the available types are **white**, **pink** and **red**.

Filter

The Filter section contains the different parameters for the synthesizer's multimode filter.

Cutoff and Resonance

- **Cutoff:** Sets the cutoff frequency of the filter, from roughly 33 Hz to 33 KHz.
- **Resonance:** Sets the resonance (or emphasis) of the filter.

Filter Modulation

- **Env Amt:** Sets the amount of unipolar modulation of the filter cutoff frequency by the filter envelope. The polarity will either be positive or negative, based on the setting of the **Invert** button in the Filter Envelope Section.
- **Keytrack:** This knob sets the amount of modulation (bipolar) of the filter cutoff frequency by the note pitch. At 100%, the cutoff frequency will track the pitch of the notes played exactly (1 octave up = a doubling in cutoff frequency).

Filter Modes

There are multiple options to configure the filter, depending on the setting of the **4-Pole** button.

4 Pole Disabled

With the 4-Pole option is disabled, the filter will closely resemble the original SEM's filter, which is a 2-pole (12 dB/octave) continuous state-variable filter.

Mode and BP Button The **Mode** knob allows to seamlessly blend between different filter types: from low-pass to notch to high-pass. The **Band-Pass** button replaces the notch filter with a band-pass filter. The **Mode** knob will therefore blend between low-pass, band-pass and high-pass.

Push The **Push** button increases the maximum resonance of the filter, allowing it to self-oscillate.

4 Pole Enabled

With the 4-Pole option enabled, the filter will have 24 dB/octave topology, with some additional features outlined below.

Mode The **Mode** knob allows to crossfade between the outputs found after each pole of the filter: from 24 dB/octave to 18 dB/octave to 12 dB/octave to 6 dB/octave.

Xpander Switching on the **Xpander** button gives access to a set of different 24 dB/octave filter configurations, still based on the same OB-X filter behaviors. The **Mode** control is replaced with a dropdown menu to select between different filter types:

- **LP4**: 4-pole low-pass filter (24 dB/octave)
- **LP3**: 3-pole low-pass filter (18 dB/octave)
- **LP2**: 2-pole low-pass filter (12 dB/octave)
- **LP1**: 1-pole low-pass filter (6 dB/octave)
- **HP3**: 3-pole high-pass filter (18 dB/octave)
- **HP2**: 2-pole high-pass filter (12 dB/octave)
- **HP1**: 1-pole high-pass filter (6 dB/octave)
- **BP4**: 4-pole band-pass filter (24 dB/octave)
- **BP2**: 2-pole band-pass filter (12 dB/octave)
- **N2**: 2-pole notch filter (12 dB/octave)
- **PH3**: 3-pole phase-shift filter (18 dB/octave)
- **HP2 + LP1**: 2-pole high-pass filter in series with a 1-pole low-pass filter
- **HP3 + LP1**: 3-pole high-pass filter in series with a 1-pole low-pass filter
- **N2 + LP1**: 2-pole notch filter in series with a 1-pole low-pass filter
- **PH3 + LP1**: 3-pole phase-shift filter in series with a 1-pole low-pass filter

LFO

OB-Xf provides two LFOs (or Low Frequency Oscillators), which can be used to modulate various targets. Both LFOs can modulate the exact same targets, however the major difference between them is that LFO 1 is global for all voices (free-running), whereas LFO 2 is per voice (key-triggered and always starting from 0° phase).

Sync and 1 & 2 Buttons

- **Sync:** This button enables synchronization of the LFO rate to the host tempo. When enabled, the rate knob will be quantized to musical note divisions (for example: 1/4, 1/8 triplet, 1/16 dotted).
- **1 & 2:** Switches the view between the two LFOs, allowing you to edit the parameters of either.

Rate

This knob sets the speed of the LFO, ranging from 0 to 250 Hz (or tempo-synced to various note divisions).

Shape Knobs

These knobs allow you to set up a blend of different waveforms at the LFO output:

- **Sine - Triangle:** Set to the left to mix in sine wave, to the right for a triangle wave, and in the middle (DC) for no output from this blend.
- **Pulse - Sawtooth:** Set to the left to mix in a pulse wave, to the right for a sawtooth wave, and in the middle (DC) for no output from this blend.
- **Pulse Width - Tension:** This vertical slider sets the pulse width of the pulse waveform and the tension of the sawtooth waveform respectively.
- **Sample & Hold - Sample & Glide:** Set to the left to mix in a sample & hold waveform, to the right for a sample & glide waveform, and in the middle (DC) for no output from this blend.

Destination Columns

Two columns of destination buttons allow you to select two sets of modulation destinations and their amount.

Mod 1 and Mod 2

These knobs set the amount of modulation applied to the selected destinations for each column.

Destination Buttons

Each button corresponds to a specific modulation destination. When a multi-state button is activated, the LFO will modulate that parameter according to the amount set by the corresponding **Mod** knob. Each button has a positive and negative polarity option, represented by their left and right LEDs, respectively (green and red for LFO 1, yellow and blue for LFO 2).

The available modulation destinations include:

- **Mod 1**:
  - **Osc 1:** Modulates the pitch of oscillator 1.
  - **Osc 2:** Modulates the pitch of oscillator 2.
  - **Filter:** Modulates the cutoff frequency of the filter.
- **Mod 2**:
  - **PW 1:** Modulates the pulse width of oscillator 1.
  - **PW 2:** Modulates the pulse width of oscillator 2.
  - **Volume:** Modulates the post-filter amplifier output level.

Envelopes

Filter and Amplifier Envelopes

OB-Xf features two ADSR envelopes per voice: one for the filter and one for the amplifier. Both envelopes have a common set of parameters.

Common Parameters

- **Attack:** Sets the time it takes for the envelope to rise from silence to its maximum level after a key was pressed. The attack time can range from very fast (1 ms) to very slow (up to 60 seconds).
- **Decay:** Sets the time it takes for the envelope to fall from its maximum level to the sustain level after the attack phase is complete. The decay time can range from very fast (1 ms) to very slow (up to 60 seconds).
- **Sustain:** Sets the level at which the envelope will hold after the decay phase, as long as the key is held (or a sustain pedal is depressed). The sustain level can be set from 0% to 100%.
- **Release:** Sets the time it takes for the envelope to fall from the sustain level back to silence after a key is released. The release time can range from very fast (1 ms) to very slow (up to 60 seconds).
- **Curve:** Sets the curvature of the Attack stage. At minimum, the curve is logarithmic, at maximum, the curve is linear.
- **Velocity:** Sets how much the envelope output will be scaled by the velocity of incoming MIDI notes. Particularly useful for creating vivid sounds that respond to user's playing dynamics in both timbre and volume.

Invert

This parameter is only available for the Filter Envelope. It simply inverts the output of the envelope, which as a result modulates the target parameters below their current value, rather than above.

Voice Variation

The Voice Variation section contains parameters that affect the overall character and differences between each voice in OB-Xf.

Per-Voice Parameter Spread

- **Glide**: Sets the amount of variation in portamento time between voices. This can notably create a more detuned or chorused effect when multiple voices are played in unison and glided between.
- **Filters**: Sets the amount of variation in filter cutoff frequency between voices. Higher values result in a wider spread of cutoff frequencies, leading to a richer and more complex sound.
- **Envelopes**: Sets the amount of variation in envelope time parameters (attack, decay, release) between voices. Increasing this value can add subtle differences in how each voice is articulated.
- **Levels**: Sets the amount of variation in amplitude levels between voices. Higher values can create a more dynamic sound. This is applied at the final output stage of each voice (post-filter).

Pan Knobs and Voice Indicators

The 8 pan knobs allow panning of each individual voice in the stereo field. This can create a wider and more immersive sound.

To the right of each pan knob, an LED indicator shows when the corresponding voice is active. If fewer than 8 voices are used (see Poly), only the corresponding number of pan knobs will be applied and have an LED next to it. If more than 8 voices are used, the pan knobs will cycle through the voices (e.g., with 10 voices, pan knobs 3-8 control voices 3-8, and pan knobs 1-2 control voices 1-2 and 9-10). Additional LEDs of different colors will be displayed to show the activity of these extra voices.

Theme Authoring

This guide explains how to create a custom skin (theme) for OB-Xf. A theme lives in its own folder and consists of a `theme.xml` layout file plus a collection of graphical assets — either PNG bitmaps or SVG vectors.

Where Themes Live

Themes are stored in the OB-Xf user data directory under:

`<UserData>/Surge Synth Team/OB-Xf/Themes/<ThemeName>/`

To create a new theme, make a new folder with your theme name and place your `theme.xml` and assets inside it.

Default Bitmap Theme — Installed Location

The built-in **Default** bitmap theme is installed by the OB-Xf installer and is a good reference for bitmap-based skins. After running the installer you will find it at:

Platform	Path
macOS	`/Library/Application Support/Surge Synth Team/OB-Xf/Themes/Default/`
Windows (system-wide install)	`%ProgramData%\Surge Synth Team\OB-Xf\Themes\Default\`
Windows (per-user install)	`%LocalAppData%\Surge Synth Team\OB-Xf\Themes\Default\`
Linux (system-wide install)	`/usr/share/Surge Synth Team/OB-Xf/Themes/Default/`
Linux (per-user install)	`~/.local/share/Surge Synth Team/OB-Xf/Themes/Default/`

> **Tip (Linux):** The exact data directory OB-Xf is using on your system is shown in the plugin's **About** screen. If the path differs from the defaults above (e.g. because `CMAKE_INSTALL_PREFIX` or `XDG_DATA_HOME` is set), use the path shown there instead.

Vector Theme — Downloading from GitHub

The **VectorTheme** is the reference for SVG-based skins. It lives in the OB-Xf source repository on GitHub. You can download it without cloning the entire repo using `git sparse-checkout`, or simply browse and download the folder directly from GitHub.

Option 1 — Download the folder via the GitHub web UI:

1. Go to (OB-Xf Github)[https://github.com/surge-synthesizer/OB-Xf]

2. Navigate to `assets/binary/VectorTheme/`
3. Use the **Download ZIP** button (or a tool such as DownGit) to download just that folder.
4. Extract the contents into a new folder inside your OB-Xf Themes directory, e.g. `UserData/Surge Synth Team/OB-Xf/Themes/MyVectorTheme/`.

Option 2 — Sparse clone with Git:

```
git clone --filter=blob:none --sparse \
    https://github.com/surge-synthesizer/OB-Xf.git ob-xf-src
cd ob-xf-src
git sparse-checkout set assets/binary/VectorTheme
```

The theme files will then be at `ob-xf-src/assets/binary/VectorTheme/`. Copy that folder (or its contents) into your OB-Xf Themes directory.

theme.xml — Layout File

Every theme must contain a `theme.xml` at the root of the theme folder. This file tells OB-Xf where every widget is positioned on screen and which graphical asset it uses.

File Structure

```
<?xml version="1.0"?>
<obxf-theme>
    <widget name="volumeKnob"   x="52"  y="127"              d="40"  pic="knob"     />
    <widget name="hqModeButton" x="128" y="233" w="23" h="35"        pic="button"   />
    <widget name="slider-h"     x="844" y="110" w="50" h="13" fh="13" pic="slider-h" />
    <!-- ... -->
</obxf-theme>
```

The root element is `<obxf-theme>`. Each child `<widget>` element describes one control.

Widget Attributes

Attribute	Applies to	Description
`name`	all	Unique widget identifier — **must not be changed**.
`x`	all	Left edge of the widget in pixels (at 1× zoom).
`y`	all	Top edge of the widget in pixels (at 1× zoom).
`w`	buttons, menus, sliders, labels	Width in pixels.
`h`	buttons, menus, sliders, labels	Height in pixels.

Attribute	Applies to	Description
`d`	knobs (PNG filmstrip)	Diameter of the knob in pixels. Also used as the track length for SVG layered sliders (see below).
`fh`	sliders, labels	Frame height — the height of a single frame within the filmstrip.
`pic`	most widgets	Filename of the asset to use, **without extension**.

Widget Types

Knobs A widget becomes a **knob** when you supply the `d` attribute instead of `w`/`h`:

```
<widget name="filterCutoffKnob" x="616" y="48" d="40" pic="knob" />
```

`d` sets both the display diameter and (for PNG themes) the frame size within the filmstrip.

Buttons Buttons use `w` and `h`. The asset is a vertical filmstrip where each frame is `h` pixels tall:

```
<widget name="hqModeButton" x="128" y="233" w="23" h="35" pic="button" />
```

Sliders A widget becomes a **slider** when you use `w`, `h`, and `fh` (frame height) instead of `d`. OB-Xf automatically determines drag direction from the aspect ratio: if `w > h` the slider is horizontal; if `h > w` it is vertical.

```
<!-- Horizontal slider -->
<widget name="filterEnvAttackCurveSlider" x="844" y="110" w="50" h="13" fh="13" pic="slider-h"

<!-- Vertical slider -->
<widget name="someVerticalSlider" x="100" y="200" w="13" h="50" fh="13" pic="slider-v" />
```

Menus Menus use `w` and `h` but do **not** use a `pic` attribute — their appearance is drawn by the host/OS. You only control their position and size:

```
<widget name="polyphonyMenu" x="56" y="235" w="31" h="31" />
```

Labels / Decorative Images Static images (waveform icons, LED indicators, background panels, etc.) use `w`, `h`, and optionally `fh`:

```
<widget name="osc1TriangleLabel" x="299" y="170" w="16" h="10" pic="label-osc-triangle" />
```

The `pic` Attribute and Default Names

Most widgets have a **hardcoded default filename** that is used when `pic` is omitted. You only need to supply `pic` when you want to use a differently-named asset. The comments in the reference

`theme.xml` files indicate which widgets have hardcoded names (shown as commented-out `pic` attributes).

```
<!-- hardcoded name - pic attribute is optional here -->
<widget name="polyphonyMenu" x="56" y="235" w="31" h="31" /> <!-- pic="menu-poly" -->

<!-- explicit pic - useful when sharing one asset across multiple widgets -->
<widget name="volumeKnob" x="52" y="127" d="40" pic="knob" />
```

Knob-to-Slider Conversion

Any knob can be turned into a slider (and vice versa) purely through the attributes you supply:

- Provide `d` □ treated as a **knob**.
- Provide `w`, `h`, `fh` □ treated as a **slider**.

This lets you redesign the interaction style of a control without changing its `name`.

Bitmap (PNG) Themes

In a bitmap theme every asset is a PNG file. OB-Xf supports three zoom levels and selects the appropriate file automatically based on the display scale factor.

Zoom Levels / HiDPI Variants

Suffix	Scale	Example filename
(none)	1×	`knob.png`
`@2x`	2×	`knob@2x.png`
`@4x`	4×	`knob@4x.png`

You must supply all three variants for every asset. The `@2x` image should be exactly twice the pixel dimensions of the base image, and `@4x` exactly four times.

Knob Filmstrips

A knob asset is a **vertical filmstrip**: a single PNG that contains every rotation frame stacked top-to-bottom. The frame size equals the knob diameter `d` × `d` pixels.

Default theme example — `knob.png`:

- Diameter (`d`): 40 px
- Frame size: 40 × 40 px
- Total frames: 140
- Image size: 40 × 5 600 px (`40 × 140 = 5600`)
- `knob@2x.png`: 80 × 11 200 px

- `knob@4x.png`: 160 × 22 400 px

Frame 0 (top) is the fully counter-clockwise position; the last frame is fully clockwise. OB-Xf selects the frame that corresponds to the current parameter value.

Button Filmstrips

Buttons are also vertical filmstrips. Each frame is `w` × `h` pixels. OB-Xf uses two different frame-layout conventions depending on the button type.

Standard Toggle Buttons (2 states, 4 frames) Most buttons are simple on/off toggles. Their filmstrip has **4 frames**:

Frame	State
0	Off — not pressed
1	Off — pressed (mouse held down)
2	On — not pressed
3	On — pressed (mouse held down)

Default theme example — `button.png`:

- Widget size: 23 × 35 px
- Total frames: 4 (image height 140 px = 4 × 35)
- `button@2x.png`: 46 × 280 px

Multi-State Buttons (N states, N×2 frames) Some buttons cycle through more than two states. The noise colour button (`button-slim-noise`) is the primary example — it has **3 states** (e.g. white noise, pink noise, off) and therefore **6 frames**.

The frame layout for an N-state button is:

```
frame = (stateIndex * 2) + mouseButtonPressed
```

So for a 3-state button the filmstrip is:

Frame	State
0	State 0 — not pressed
1	State 0 — pressed
2	State 1 — not pressed
3	State 1 — pressed
4	State 2 — not pressed
5	State 2 — pressed

Default theme example — `button-slim-noise.png`:

- Widget size: 18 × 13 px (`h="13"` in `theme.xml`)
- Total frames: 6 (image height 78 px = 6 × 13)
- `button-slim-noise@2x.png`: 36 × 156 px

  Tip: To find out how many states a particular button has, count the frames in the Default theme filmstrip (`imageHeight / h`) and divide by 2. If the result is 2 it is a standard toggle; if it is 3 or more it is a multi-state button.

Slider Filmstrips

Sliders use a filmstrip where each frame is `w` × `fh` pixels (for horizontal sliders) or `fh` × `h` pixels (for vertical sliders). The filmstrip encodes every thumb position.

Default theme example — `slider-h.png` (horizontal):

- Widget: `w="50" h="13" fh="13"`
- Frame size: 50 × 13 px
- Total frames: 190 (image height 2 470 px = 190 × 13)

Default theme example — `slider-v.png` (vertical):

- Widget: `w="13" h="50" fh="13"` (example)
- Frame size: 13 × 13 px (one frame per position)
- Total frames: ~477 (image height 6 200 px)

Asset Checklist for a Bitmap Theme

For every asset referenced by `pic` in your `theme.xml`, provide:

```
<name>.png
<name>@2x.png
<name>@4x.png
```

Also provide the background panel:

```
background.png
background@2x.png
background@4x.png
```

Vector (SVG) Themes

In a vector theme, assets are SVG files. Because SVGs scale losslessly, you do **not** need `@2x` or `@4x` variants — a single SVG file serves all display densities.

Simple SVG Assets

For static images (backgrounds, labels, button faces, menu decorations) a single SVG file is used directly. The filename matches the `pic` attribute in `theme.xml`, with a `.svg` extension:

```
button.svg
button-slim.svg
label-osc-triangle.svg
background.svg
```

The SVG's `width` and `height` attributes should match the logical pixel dimensions used in `theme.xml`.

Layered SVGs for Knobs

Knobs in a vector theme are rendered from **two SVG layers**:

File	Role
`knob-layer1.svg`	Static background (body, shadow, ring) — does not rotate.
`knob-layer2.svg`	Rotating indicator (pointer, dot, line) — rotated by OB-Xf.

OB-Xf composites layer 1 underneath layer 2, rotating layer 2 around the centre of the knob to reflect the current parameter value. Both SVGs should be the same size (matching the `d` attribute in `theme.xml`).

Design tips:

- Keep `knob-layer1.svg` as the knob body with no directional indicator.
- Put only the pointer/indicator in `knob-layer2.svg`, centred on the SVG canvas so rotation works correctly around the middle.
- The rotation range is typically –135° to +135° (270° total sweep), matching the classic OB-X style.

Layered SVGs for Sliders

Sliders follow the same two-layer pattern:

File	Role
`slider-h-layer1.svg`	Static horizontal track.
`slider-h-layer2.svg`	Thumb / handle that moves horizontally.
`slider-v-layer1.svg`	Static vertical track.
`slider-v-layer2.svg`	Thumb / handle that moves vertically.

For SVG sliders the `d` attribute in `theme.xml` is repurposed as the **travel distance** in pixels — the number of pixels the thumb moves from its minimum to maximum position. The `w` and `h` attributes set the overall bounding box of the widget.

Asset Checklist for a Vector Theme

```
background.svg
knob-layer1.svg
knob-layer2.svg
slider-h-layer1.svg
slider-h-layer2.svg
slider-v-layer1.svg
slider-v-layer2.svg
button.svg
button-slim.svg
button-alt.svg
button-slim-alt.svg
button-clear.svg
button-clear-red.svg
button-clear-white.svg
button-dual.svg
button-dual-alt.svg
button-group-patch.svg
button-slim-noise.svg
button-slim-vibrato-wave.svg
label-osc-triangle.svg
label-osc-pulse.svg
label-lfo-wave2.svg
label-filter-mode.svg
label-filter-options.svg
label-bg-save-patch.svg
label-led1.svg
label-led2.svg
label-led3.svg
label-led4.svg
menu-poly.svg
menu-voices.svg
menu-legato.svg
menu-note-priority.svg
menu-pitch-bend.svg
menu-patch.svg
menu-categories.svg
menu-xpander.svg
```

Tips and Workflow

1. **Start from a reference theme.** Copy the Default (PNG) or VectorTheme (SVG) folder and rename it. Edit `theme.xml` and replace assets one at a time.
2. **Keep widget `name` values unchanged.** The names are how OB-Xf identifies controls internally. Changing them will break the layout.

3. **Match logical pixel dimensions.** All coordinates and sizes in `theme.xml` are in logical (1×) pixels. For PNG themes, the `@2x` and `@4x` images must be exact integer multiples of the base image.
4. **Filmstrip frame count matters.** OB-Xf derives the number of animation frames from the image height divided by the frame height (`fh` or `d`). Make sure your filmstrip height is an exact multiple of the frame size.
5. **SVG canvas size.** Set the SVG `width`/`height` to match the logical pixel size of the widget. For knob layers, both SVGs must be the same size.
6. **Test at multiple zoom levels.** Run OB-Xf at 100%, 200%, and 400% UI scale to verify that PNG assets look sharp and SVG assets render correctly at all sizes.
7. **The `pic` attribute is optional for most widgets.** If you omit it, OB-Xf falls back to the hard-coded default filename. You only need `pic` when you want to use a custom name or share one asset across multiple widgets.