

---

# **Metrasynth Documentation**

*Release 2016*

**Matthew Scott**

November 30, 2016



<b>1</b>	<b>Projects</b>	<b>3</b>
<b>2</b>	<b>Vision</b>	<b>5</b>
2.1	Leverage SunVox . . . . .	5
2.2	Expose SunVox . . . . .	5
2.3	Embed SunVox . . . . .	5
2.4	Explore SunVox . . . . .	5
2.5	Extend SunVox . . . . .	5
2.6	Respect SunVox . . . . .	6
<b>3</b>	<b>Contributors</b>	<b>7</b>
3.1	Matthew Scott . . . . .	7
<b>4</b>	<b>Indices and tables</b>	<b>9</b>



*Metrasynth* is a collection of tracker/synthesizer tools based on the venerable SunVox audio engine.

*Metrasynth* is a portmanteau of “meta tracker/synthesizer”.

Contents:



---

## Projects

---

**The Complete Guide to SunVox** A free book for beginning to advanced users, and developers incorporating SunVox DLL into their projects.

**Orbitant** MIDI clock and musical timing utilities for Python.

**Radiant Voices** Create, read, modify, and write SunVox files.

**Solar Flares** Sound design and performance tools for SunVox.

**Solar Sails** SunVox Augmentation and Interactive Live-coding System.

**SunVOSC** Bidirectional OSC bridge for SunVox DLL.

**sunvox-dll-python** Python wrapper for the SunVox DLL.

**VoxPlex** VoxPlex synchronizes files to/from SunVox for iOS.





---

## Vision

---

The long-term vision of Metrasynth is to create new ways to interact with the SunVox app, the SunVox DLL, and the SunVox file format.

We'll do so by following these principles:

### 2.1 Leverage SunVox

Whenever SunVox can do something by itself, let it do so.

### 2.2 Expose SunVox

Whatever the SunVox DLL and file format can do, you should be able to access it using Python in expressive object-oriented and/or functional ways.

### 2.3 Embed SunVox

Whatever `sunsynth` file you can create, you should be able to load as an instrument in a DAW using a freely available VST or similar plugin.

### 2.4 Explore SunVox

Reach the limits of SunVox's capabilities. Report these findings back to the community, and internalize them in Metrasynth tools.

### 2.5 Extend SunVox

Create tools that use the SunVox format and DLL, but provide novel capabilities that are not feasible using the standard SunVox UI, nor using SunVox via another pre-existing DAW.

## 2.6 Respect SunVox

Show respect for Alexander Zolotov, creator of SunVox, who graciously provides his expertly-crafted audio engine as a beautiful cross-platform UI, as well as a commercially useful, zero-cost DLL.

Show respect for all members of the SunVox community, which has grown a healthy culture of sharing and experimentation.

---

**Contributors**

---

### **3.1 Matthew Scott**

Matthew got his introduction to electronic music circa 1993, primarily through the tracker scene.

In 2016, after trying simulators such as SchismTracker, he discovered SunVox, and promptly wondered why he hadn't discovered it sooner!

Once he tried an experiment using Python and the SunVox DLL, he became ecstatic and decided to embark on a mission to not only make the most of that resource, but to also share as much as possible with the SunVox community.



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`