

Handbook

This page contains some ideas on how one can use PyFLP for automating tasks (*to a certain extent*) which can only be done via FL Studio.



A basic-to-intermediate level of Python knowledge is assumed. No prior knowledge of PyFLP is required for any of the sections below.

These all are written from a dev's POV. I would ♥ to get more ideas and hear about different use cases.

Exporting to a ZIP

Imagine you had a folder structure like this:

```
└─ Samples
   ├── 🥁 kick.wav
   ├── 🥁 clap.wav
   └── 🎵 toms.wav
└─ MyGreatSong.flp
```

For the purpose of simplicity, assume that `` MyGreatSong.flp`` uses only the samples from `` Samples`` and all ****sample file names are unique****.

The code below will create a ZIP containing all the samples used

```
from zipfile import ZipFile
import pyflp

project = pyflp.parse("MyGreatSong.flp")

with ZipFile("MyGreatSong.zip", "x") as zp:
    zp.write("MyGreatSong.flp")

    for sampler in project.channels.samplers:
        if sampler.sample_path is not None:
            zp.write(sampler.sample_path)
```

Caution

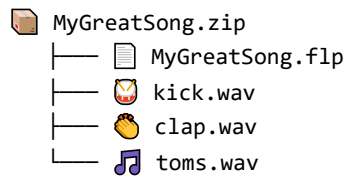
Missing samples

The above code assumes that all the samples exist at the paths the FLP has stored. If any of the samples aren't found, there will be an error.

FL Studio doesn't give up this easily. It searches up a lot of paths, including but not limited to the recursive scanning of folders which are:

- Current directory.
- Added to the sample browser.
- Containing previous samples / missing samples.

This will create a ZIP file of the structure:



Hint

FL Studio stock samples

While this will work for 3rd party samples *unless there's 2 samples with the same name*, FL Studio doesn't store the full path inside an FLP for stock samples. See [pyflp.channel.Sampler.sample_path](#) for more info.



Unlocking demo version FLPs

Caution

This doesn't work for FL Studio 21 projects. See #146 <<https://github.com/demberto/PyFLP/discussions/146>>

FLPs saved with a trial version of FL Studio cannot be reopened again without saving in a registered version. The state of demo versions of native plugins' is not retained either.

Hint

This section **doesn't** explain how to make 3rd party plugin demos recall their state. They have their own mechanisms for doing that.

It is possible to undo both of these:

See also

[Project.licensed](#) and [_PluginBase.demo_mode](#).

```
import pyflp

project = pyflp.parse("/path/to/myflp.flp")

# Unlock the FLP itself
project.licensed = True

# Unlock trial version native plugins
for instrument in project.channels.instruments:
    instrument.plugin.demo_mode = False

for insert in project.mixer:
    for slot in insert:
        if slot.plugin is not None:
            slot.plugin.demo_mode = False

pyflp.save(project, "/path/to/myflp_unlocked.flp")
```

Note

An unregistered version of FL Studio will roll back these changes once you save an FLP in it (even previously registered ones), so you need to repeat this process everytime.