Automating Ableton Sample Recording

We will try and figure out a way to automate recording samples from Serum and export the settings. Perhaps this could be done in Reaper instead. If I were to do this in Reaper I would have to figure out:

1. How to connect a MIDI instrument and use the Python script to play notes
2. Automate the recording process
3. Automate the preset information export
4. How to add a VST plugin to Reaper

I’ve figured out how to control Reaper with LoopMIDI. I am using the ‘rtmidi’ module to play notes from python into Reaper via MIDI. With the help of ChatGPT, I now know I must create a script with python to automate the recording process. I will iterate through all the files in my preset directory, play a note from each preset, and then record it into an audio file.

C: bool TrackFX\_GetPreset(MediaTrack\* track, int fx, char\* presetnameOut, int presetnameOut\_sz)

EEL2: bool TrackFX\_GetPreset(MediaTrack track, int fx, #presetname)

Lua: boolean retval, string presetname = reaper.TrackFX\_GetPreset(MediaTrack track, integer fx)

Python: (Boolean retval, MediaTrack track, Int fx, String presetnameOut, Int presetnameOut\_sz) = RPR\_TrackFX\_GetPreset(track, fx, presetnameOut, presetnameOut\_sz)

Get the name of the preset currently showing in the REAPER dropdown, or the full path to a factory preset file for VST3 plug-ins (.vstpreset). See TrackFX\_SetPreset. FX indices for tracks can have 0x1000000 added to them in order to reference record input FX (normal tracks) or hardware output FX (master track). FX indices can have 0x2000000 added to them, in which case they will be used to address FX in containers. To address a container, the 1-based subitem is multiplied by one plus the count of the FX chain and added to the 1-based container item index. e.g. to address the third item in the container at the second position of the track FX chain for tr, the index would be 0x2000000 + 3\*(TrackFX\_GetCount(tr)+1) + 2. This can be extended to sub-containers using TrackFX\_GetNamedConfigParm with container\_count and similar logic. In REAPER v7.06+, you can use the much more convenient method to navigate hierarchies, see TrackFX\_GetNamedConfigParm with parent\_container and container\_item.X.

C: int TrackFX\_GetPresetIndex(MediaTrack\* track, int fx, int\* numberOfPresetsOut)

EEL2: int TrackFX\_GetPresetIndex(MediaTrack track, int fx, int &numberOfPresets)

Lua: integer retval, integer numberOfPresets = reaper.TrackFX\_GetPresetIndex(MediaTrack track, integer fx)

Python: (Int retval, MediaTrack track, Int fx, Int numberOfPresetsOut) = RPR\_TrackFX\_GetPresetIndex(track, fx, numberOfPresetsOut)

Returns current preset index, or -1 if error. numberOfPresetsOut will be set to total number of presets available. See TrackFX\_SetPresetByIndex FX indices for tracks can have 0x1000000 added to them in order to reference record input FX (normal tracks) or hardware output FX (master track). FX indices can have 0x2000000 added to them, in which case they will be used to address FX in containers. To address a container, the 1-based subitem is multiplied by one plus the count of the FX chain and added to the 1-based container item index. e.g. to address the third item in the container at the second position of the track FX chain for tr, the index would be 0x2000000 + 3\*(TrackFX\_GetCount(tr)+1) + 2. This can be extended to sub-containers using TrackFX\_GetNamedConfigParm with container\_count and similar logic. In REAPER v7.06+, you can use the much more convenient method to navigate hierarchies, see TrackFX\_GetNamedConfigParm with parent\_container and container\_item.X.

A group of text boxes

Description automatically generated with medium confidence