RJS Sampling Suite - User Guide

How to use - Quick Start

The ideal workflow consists of running the three core scripts in sequence:

1. Chop Samples
2. Detect Sample Pitch
3. Arrange Samples for Export

In reality you need to monitor and control the process more closely:

1. Place your source audio on the first track in Reaper.
2. Run 'Chop Samples'.
3. Check the chopping results:
   - Listen to the 'Cut-outs' track.
   - Use 'Vertical View'.
   - Adjust manually or redo chopping.
4. Run 'Detect Sample Pitch'.
5. Check the pitch detection results:
   - See the labeling colors.
   - Listen to the MIDI notes together with the samples.
   - Make corrections manually.
6. Prepare the samples for arranging:
   - Choose which samples to keep (if there is excess).
   - Place the samples in ***the right order****.
7. Run 'Arrange Samples for Export'.
8. Render the samples using the following rendering settings (or change the file naming):
   - Source: 'Selected media items via master'
   - File name: '**$parenttrack\$parenttrack_$region_$track**'

*** "*The right order*":**

➢ Round robin samples next to each other (within each note label).
➢ Velocity layers from low to high.

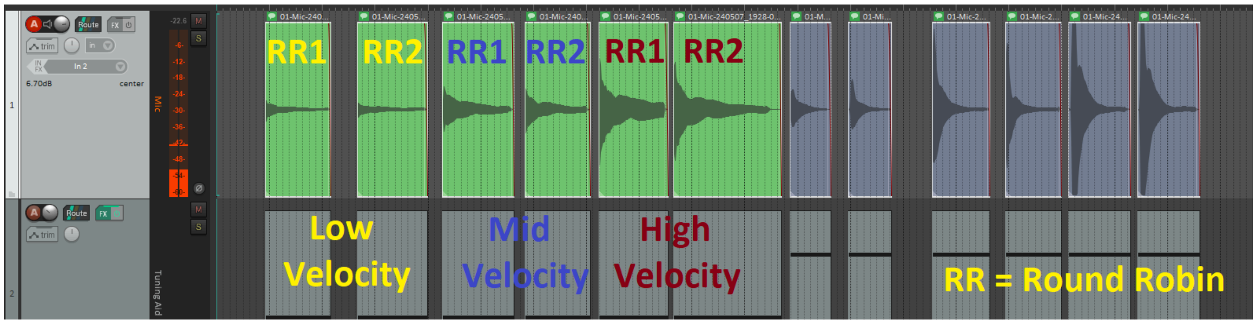


Table of contents

RJS Sampling Suite

A script collection for Reaper

Introduction

What is RJS Sampling Suite?

RJS Sampling Suite is a collection of scripts that help create sample libraries. The scripts automate tasks such as chopping recordings into samples, labeling samples based on pitch, and arranging them for rendering. The scripts are ReaScripts that work in Reaper DAW. ReaScripts are essentially custom actions for Reaper.

RJS Sampling Suite produces samples with filenames that indicate which MIDI velocities and MIDI notes the samples occupy. Thus, the samples can be automapped by a sampler program.

Why use it?

Making a sample library involves a lot of audio editing. Even smaller libraries with tens or hundreds of samples take a significant time investment and effort to create. And the core of the problem isn't even the work amount, it is the nature of the work. Manual sample editing is tedious and boring. It destroys artistic and creative flow. It is the major obstacle that discourages a sample instrument maker to follow through with a creative idea.

RJS Sampling Suite aims to eliminate the worst parts of sample instrument creation. With RJS Sampling Suite you can:

- Save time and energy
- Achieve precision
- Test ideas quickly
- Prototype iteratively
- Take on larger projects
- Manage projects effectively and flexibly

How to use

How to prepare a project and start working

1. Place the source audio on the first track in Reaper.
    - **The core scripts always process the items on the first track.**
2. Follow the workflow.

How to process a multi mic recording

1. Place the source audio of the main microphone on the first track in Reaper.
    - The scripts make cuts, pitch detection, etc. based on this audio item.
    - A close mic recording usually has the clearest transient and signal overall.
2. Place the source audio of the other mics on the tracks below.
3. Align the source audio items.
4. Group the source audio items using Reaper’s grouping function [Select items + G].
5. Work on the source audio on the first track → Changes get copied to the other items.
    - After arranging and rendering the mic number is included into the sample name.
    - To configure the mic naming more precisely, use ‘Configure Sample Naming’ or manually change the name of the tracks where you see the word ‘Mic’.

How to process percussions

1. Chop normally.
    - You might have to use a short ‘Minimum sample length’ value.
2. Use ‘Assign Midi Note Numbers’ to label the samples.(See the script file for instructions)
    - Here you have to decide how the samples get mapped in the sampler program.
3. Arrange and render.
    - You might want to use the ‘Stretch Policy’ setting ‘nostretch’ (value: 4).
        - Each sample occupies only the MIDI note they were assigned to.
        - No stretching to close the gaps between the samples.

How to deal with inaccurate chopping (of a noisy recording)

- Option 1: Lower the sensitivity of the chopping. See the script file about 'noisy modes'.
- Option 2: Use the EQ'd "sidechain" trick:
    1. Put an EQ on the source audio track and set up a high pass filter.
        - Other effects to enhance the signal can be used as well.
    2. Render the EQ'd track into a stem track.
    3. Move the new stem track on top (if it is not placed there already by Reaper).
    4. Group the audio items on the original track and on the stem track [Select + G].
    5. Run 'Chop Samples'
    6. Delete the stem track, move the original track back on top, and continue.

How to adjust sample tails (and starting points)

1. Listen to the ‘Cut-outs’ track to determine where audible tails end.
   Look at zoomed-in waveforms to visually detect possible noises and where the tails end.
   - Zoom in: Shift + Up Arrow (on Windows).
2. Option 1: Use ‘Decrease/Increase Tail Length’ to adjust the selected samples.
   Option 2: Use the mouse to adjust the tails manually.

- If needed, use a similar process for sample starting points.
  - Use ‘Decrease/Increase Leading Pad Length’ or the mouse.
  - An alternative: Accuracy problems of the starting points are usually resolved by changing chopping algorithm parameter values. (See also this.)

How to use ‘Vertical View’

1. Run ‘ENTER Vertical View’.
2. Check the sample starting points for inconsistencies.
3. Make manual adjustments by dragging the contents inside audio items.
   - ALT + Drag (on Windows)
4. Run ‘EXIT Vertical View’.

How to correct the pitch detection results manually

- Change the note of a MIDI item to the correct pitch and run 'Update Sample Info'.
- If a MIDI item is missing, create one by recording or copying, create/modify a note, and run 'Update Sample Info'.

How to tune samples

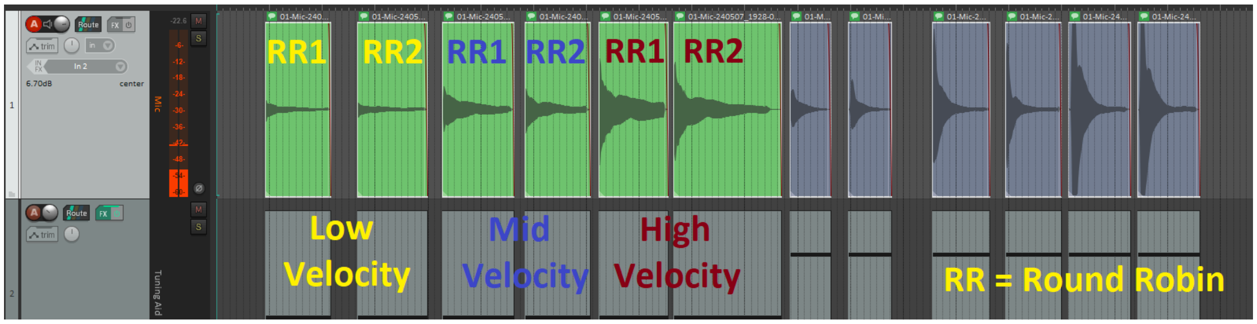
1. Make sure samples have been labeled correctly.
2. Select the samples on the first track you want to tune.
3. Option 1: Run ‘Tune Selected Samples’.
   Option 2: Use ‘Tune Selected Samples UP/DOWN by 1 cent’ scripts to tune manually.
   - Use the MIDI synth on the ‘Tuning Aid’ track to assess the tuning by ear.

How to take a sample inventory

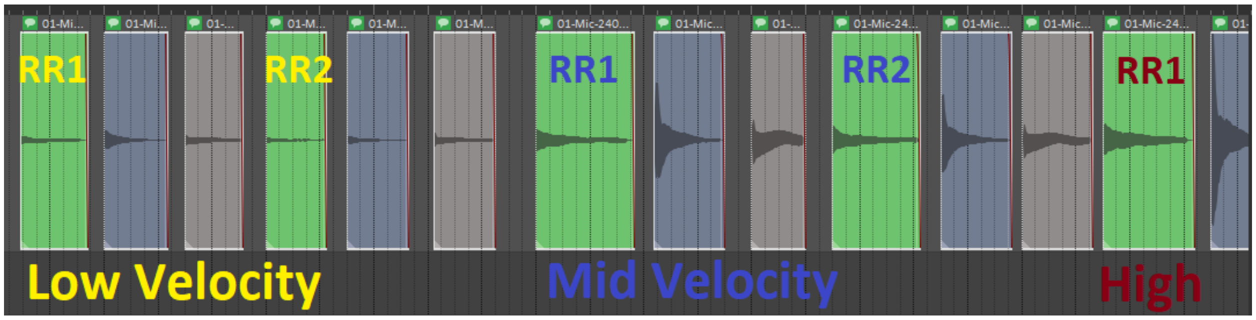
1. Run ‘Take a Sample Inventory’ to open a dialog.
2. Input your planned amount of round robins and velocity layers.
3. See the report if the actual sample amounts match the plan.
   - No match: Add/Remove/Correct the samples pointed out and repeat the process.

How to prepare the samples for arranging

1. Choose which samples you want to include in your instrument and remove the rest.
    - Taking an inventory of the samples helps spot excess samples and other problems.
    - Ordering the samples and having them close together can help make decisions about round robin samples and velocity layers.
        - Run 'Sort Items Based on Note and Energy' to sort the samples.
        - Use 'Line up Items on the Selected Track' if you don't want to change the order of the samples but want them to be closer to one another.
2. Place the selected samples in ***the right order*** (within each note label).
    - Round Robin samples of each note must be next to each other.
    - Velocity layers must grow from low to high.
    - The order of the musical notes (such as C, D, etc.) **does not matter.***



***Note:** The samples above could also be recorded/ordered alternating between musical notes.
- For the arrangement algorithm this makes no difference as long as the samples are in the right order **within each note label** (color). The arrangement result will be the same.

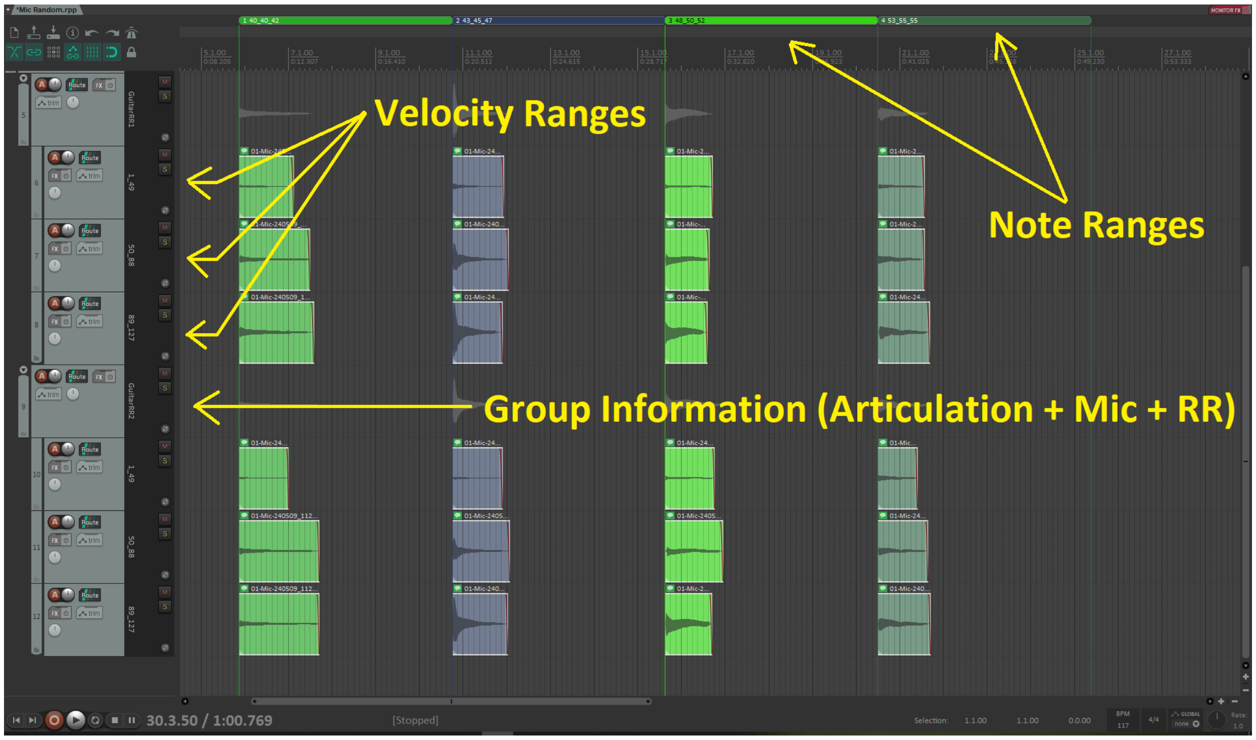


How to change the sample naming convention

1. Run 'Configure sample naming' to open a dialogue.
2. Form a name structure using the keywords available.
    - See the script file for more information about the keywords.
3. Render using the wildcard '**$item**' for the file names:
    - Source: 'Selected media items via master'
    - File name: '$parenttrack**$item**'

What is the arrangement that enables exporting?

Running the script ‘Arrange Samples for Export’ builds the structure of the sampled instrument. Each sample is given a note range and a velocity range, which are calculated based on user input. The arrangement information is then stored in the region markers and track names created by the script. Sample filenames are formed with this information during exporting.



About filenames and automapping

A lot of the value that RJS Sampling Suite produces is in the filenames of the rendered samples. Collectively the filenames encode the structure of the sampled instrument. To map a sample into a sampler program, five parameters are needed:

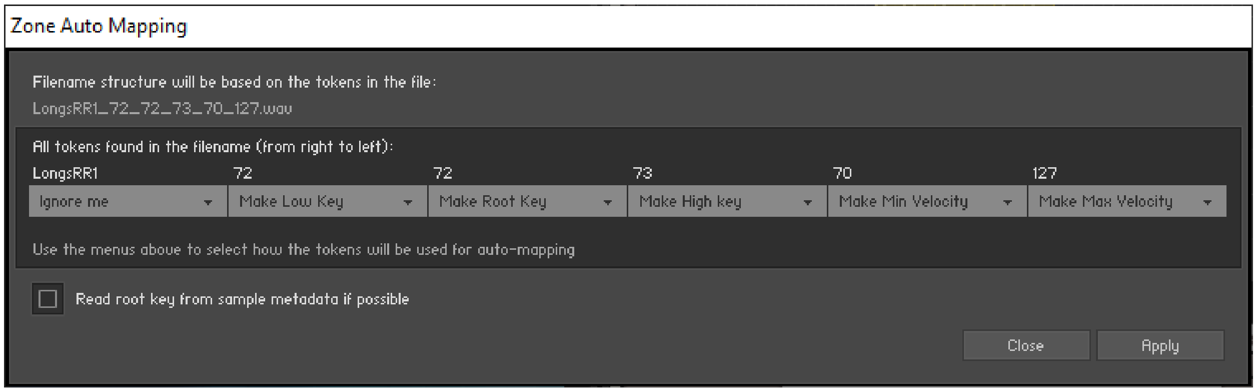
- Root Key: The MIDI note the sample is “anchored” to.
  - The key on the keyboard that produces the sample’s sound with its original pitch.
- Low Key: The MIDI note that is the lower limit of a stretched sample.
  - The lowest key that triggers the sample.
- High Key: The MIDI note that is the higher limit of a stretched sample.
  - The highest key that triggers the sample.
- Min Velocity: The lowest MIDI note velocity value that triggers the sample.
- Max Velocity: The highest MIDI note velocity value that triggers the sample.

These five parameters are encoded into the filenames produced by the RJS Sampling Suite. It is also possible to reorder, add, omit, or modify information by changing the naming convention. For instance, it is possible to convert MIDI note numbers into note letter symbols and velocity values into velocity layer ordering numbers.

Here’s an example of a filename produced by RJS Sampling Suite:

- Guitar_36_43_48_40_100.wav

Here are Kontakt’s automapping settings that can map based on the filename:



Zone Auto Mapping

Filename structure will be based on the tokens in the file:
LongsRR1_72_72_73_70_127.wav

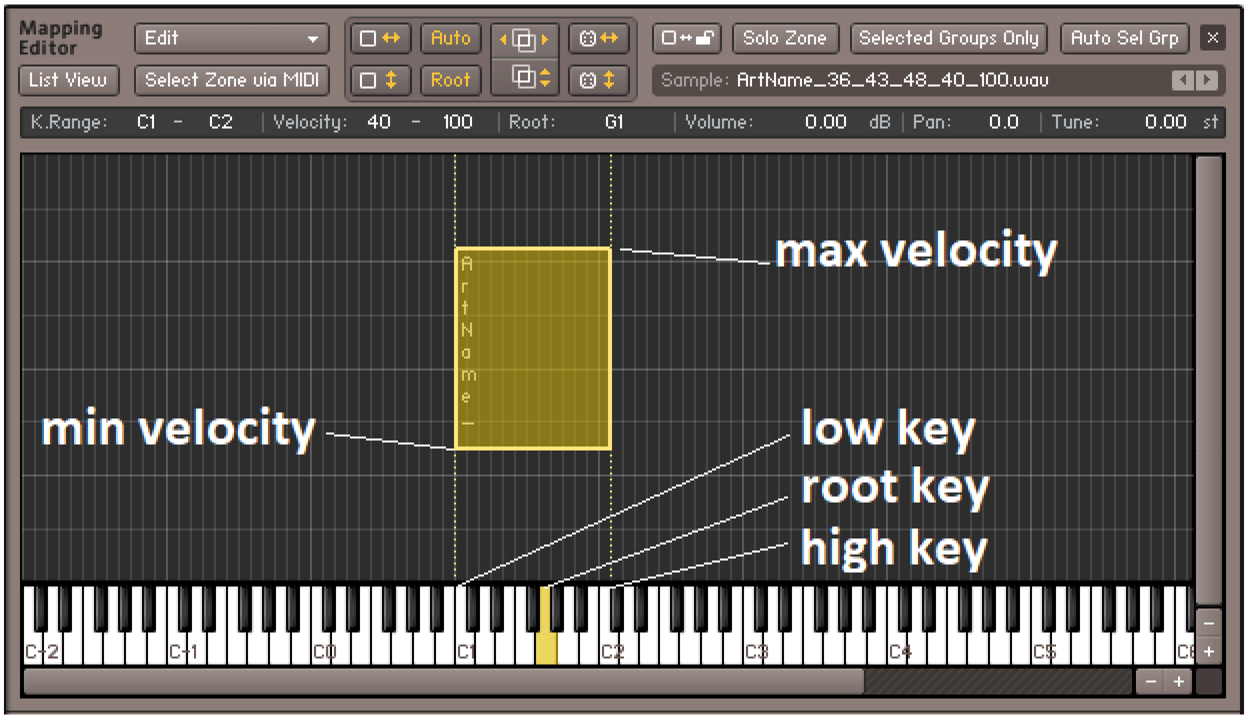
All tokens found in the filename (from right to left):

LongsRR1	72	72	73	70	127
Ignore me	Make Low Key	Make Root Key	Make High key	Make Min Velocity	Make Max Velocity

Use the menus above to select how the tokens will be used for auto-mapping

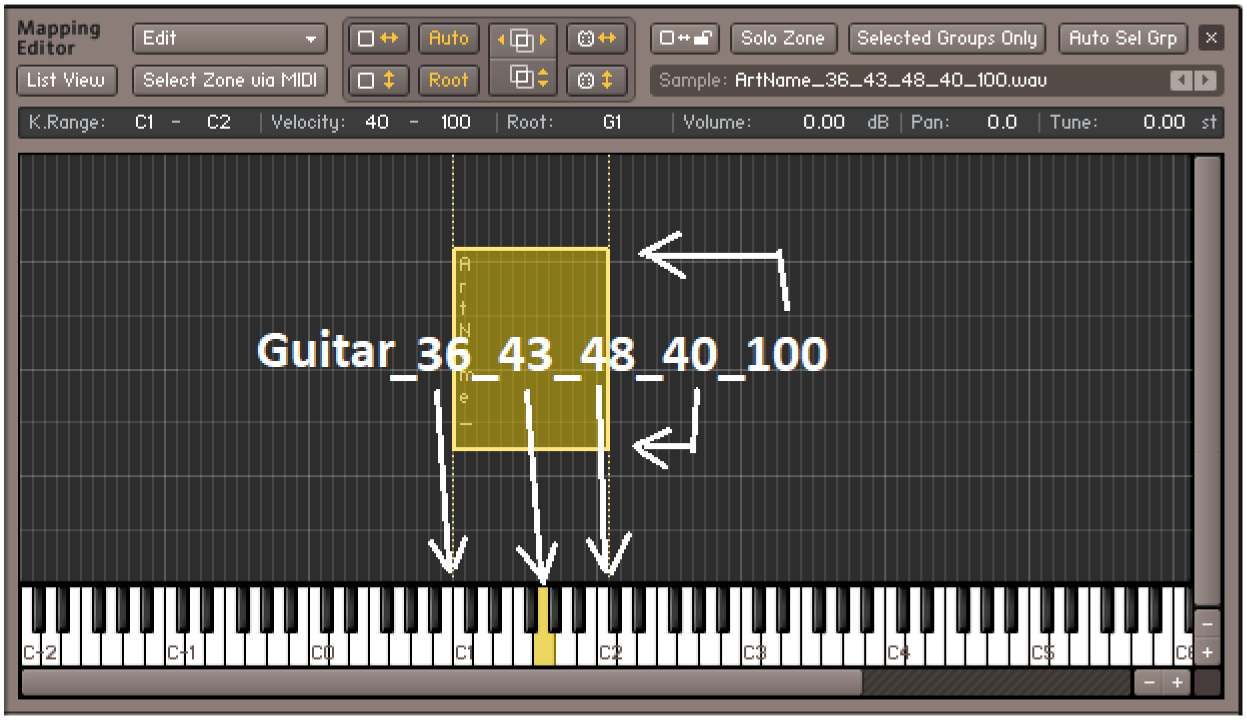
☐ Read root key from sample metadata if possible

Here’s how the sample file above would be mapped in Kontakt:



Sample: ArtName_36_43_48_40_100.wav

K.Range: C1 - C2 | Velocity: 40 - 100 | Root: G1 | Volume: 0.00 dB | Pan: 0.0 | Tune: 0.00 st



Sample: ArtName_36_43_48_40_100.wav

K.Range: C1 - C2 | Velocity: 40 - 100 | Root: G1 | Volume: 0.00 dB | Pan: 0.0 | Tune: 0.00 st

Miscellaneous

Tips for managing projects in Reaper

- Save the project with a new name at important stages of the process, and especially before running 'Arrange Samples for Export'.
- Zoom in [Shift + Up Arrow (on Windows)] to see the audio waveform better.
- Place effects on the master track to process the samples during rendering.
    - A convenient way to render different versions of the sample set, e.g. raw samples, tuned samples, EQ'd samples etc..

Tips for learning to use the scripts

1. Create simple instruments with a few samples using only the core scripts.
    - Don't pay too much attention to the quality of the work.
2. After getting familiar with the basic workflow, start monitoring and improving quality.
    - Try different chopping sensitivity and sample length settings.
        - Look at the results using 'Vertical View'.
    - Try using pitch detection on various sounds and correct potential mislabelings.
    - Try arranging the same sample set using different arrangement settings.
3. See what kind of helper scripts there are and learn to use the ones you need.

Known challenges

- Pitch detection has trouble working on really low and high notes.
- Pitch detection has trouble working on sounds with strong/metallic overtone content.
    - Example: Kalimba
- Long slowly fading sound tails are not perfectly catched by the chopping algorithm.
    - Example: Guitar, piano, etc..
    - Use your ear and a zoomed-in waveform to adjust the tail length.
- Sounds with slow and soft attacks are not perfectly catched by the chopping algorithm.
    - Example: Strings, shakers, etc..
    - Use your ear and a zoomed-in waveform to adjust the starting point.
- (Inaudible) Low frequency rumble/noise can cause problems for the chopping algorithm.
    - Symptoms: Cuts start too early, and/or lack of cuts between samples.
    - Mitigate using the 'EQ'd "sidechain" trick'.

Changing the default settings of the scripts

- Some of the scripts have default settings that can be changed by modifying the .lua file.
- Open a script file in a text editor and look for a section called 'Default Settings.'

Additional information and technical reference

- Each .lua script file includes instructions and additional information specific to that script.
- Script files can be opened and modified in a regular text editor.

The Scripts

Core Scripts

There are three scripts that are essential in the workflow of the RJS Sampling Suite.
In an ideal case, these are the only scripts needed.

Chop Samples

- Cuts the source audio into pieces/samples
- Places the silent parts separating the actual samples on a new track

Precise chopping is fundamental to the quality of a sample library.
This script also offers a tool to assess the quality of the chopping.

Detect Sample Pitch

- Detects the pitches of the samples and labels them accordingly (with different colors)
- Creates MIDI items/notes underneath each sample on a new track with a synthesizer

Labeling is needed for arranging and naming the samples later.
This script also offers a tool to assess and correct the labeling result.

Arrange Samples For Export

- Calculates the MIDI velocity layers for the library
- Calculates the MIDI note spread for each sample
- Creates new tracks and timeline regions that carry the calculated information
- Places the samples on the newly created tracks to enable rendering/exporting

Arranging makes it possible to render the samples into automatically named WAV files.
The filenames include the information about the sample library structure.
A sampler program (such as Kontakt) can 'automap' the samples based on the filenames.

Helper Scripts

The helper scripts can be used between the core scripts.
The helper scripts offer tools to assess and correct the results of the core scripts.

Assign Midi Note Numbers

- Labels samples based on manually created timeline markers

This script can be used instead of 'Detect Sample Pitch' to label samples.
The script is needed when making percussion libraries.

Configure Sample Naming

- Creates sample names based on user input

This script makes it possible to form custom naming for the samples.
Custom naming can help target various samplers and their automapping features.

Decrease/Increase Leading Pad Length

- Changes the starting point of the selected samples
- Changes the cut-out parts of the audio to reflect the change of the samples

These scripts can be used to finetune the chopping result.

Decrease/Increase Tail Length

- Changes the end point of the selected samples
- Changes the cut-out parts of the audio to reflect the change of the samples

These scripts can be used to finetune the chopping result.

ENTER Vertical View

- Places each sample on a new track
- Zooms in the view to highlight the starting point of the samples

This script makes it easy to assess the precision of the sample starting points after chopping.

EXIT Vertical View

- Places each sample back on the their original position on the first track
- Zooms out the view back to its original state

This script must be run after using 'Vertical View'.

Line up Items on the Selected Track

- Moves the samples close to one another

This script can help keep things organized when preparing for the arrangement phase.

Sort Items Based on Note and Energy

- Calculates signal energy of each sample
- Reorders the samples from low note to high note
- Reorders the samples from low energy to high energy within each note label

This script can be used when preparing for the arranging of the samples.
Sorting makes it easier to assess which samples to include into the library when there are excess samples.
Sorting is required if the notes/sounds in the source audio were not recorded in a consistent order in terms of loudness (velocity layers).

Take a Sample Inventory

- Counts the samples by labels
- Tells if the count matches the amounts needed for the planned library

This script helps count the samples before arranging.
The script exposes excess samples, unlabelled samples and "garbage".

Tune Selected Samples UP/DOWN by 1 cent

- Changes the pitch of the selected samples by 1 cent.

Tune Selected Samples

- Changes the pitch of the entire sample by a fixed amount to be closer to the ideal pitch
- No continuous tuning like autotune

Update Sample Info

- Updates the sample labeling

This script must be run after correcting sample labeling manually.