ReaClassical Core Manual

Generated 2026-01-14 23:54:37 UTC

Table of Contents

Welcome!

> Thank you for your interest in ReaClassical Core. This project is driven by a passion to create a flexible classical music editing tool for musicians, composers, and sound engineers. Your support and feedback are essential for its improvement.
>
> Whether you're new to ReaClassical Core or have used it from the start, your trust inspires continued innovation. I hope it enhances your classical music production and supports your creative vision.
>
> 

Preliminaries

It is important to note that if you already own REAPER then the world of classical editing including source-destination editing (aka 2-, 3-, and-4-point editing) is available at no extra cost to you via the freely available ReaClassical Core package. There's no need to spend any of your hard-earned money on Sequoia, Pyramix or SaDiE in order to make editing precise and efficient. As a classical engineer myself, I can say with certainty that what I am about to share with you covers all my editing needs. Indeed, I couldn't return to the old way of working at this point. Your mileage may vary and I'd love to hear from you if there are functions that you feel might be missing.

ReaClassical vs ReaClassical Core

ReaClassical Core serves as an accessible entry point to the complete recording, editing, mixing, and mastering world of ReaClassical. It's a great way to get started editing the *ReaClassical* way inside of a regular REAPER instance. If you like *Core*, consider trying the full ReaClassical ecosystem - it's also completely free and open source! For more information see https://reaclassical.org.

Compatibility

ReaClassical Core runs on any system that is compatible with REAPER (nine architectures!). This includes 64-bit and 32-bit versions of Windows, MacOS and Linux (including Raspberry Pi).

This Manual

This online manual, located at https://reaclassical.org/core, serves as the official documentation for ReaClassical Core. The structure of the manual is designed to take the user through preliminary remarks, then a detailed look at editing workflows. After some brief closing remarks, there follows the appendices (descriptions of all the ReaClassical Core functions and recommended keyboard shortcuts). I highly recommend doing a complete read of the short manual and becoming very

familiar with the appendices.

ReaClassical Community

The community thread plays an important role in the development of ReaClassical. Not only is it a place for users to suggest feature requests and point out bugs but also discuss more general classical music recording, mixing and mastering techniques. It also serves as something of a development blog as I not only announce the regular releases but also document the under-the-surface details for those that are interested.

Relatively new is the ReaClassical Discord server. This is a great place for live support, general chat, proposing feature requests, workflow discussion, and letting the community know about albums or individual pieces you have created with the help of ReaClassical or ReaClassical Core.

Ways to Contribute

The most important way users can contribute to the development of ReaClassical or ReaClassical Core is to actually use the tools! It makes me happy to know that engineers can make whole professional-sounding and technically accurate masters from ReaClassical. Another is to suggest features or let me know about bugs. You can either do this on the thread or via the Issues page on the ReaClassical GitHub. Finally, I'd be glad of any monetary donations. You can use PayPal to do so.

Spread the Word!

If you've enjoyed using ReaClassical Core in your projects, I'd be incredibly grateful if you could mention it in your album booklets, social media video descriptions, or anywhere you typically include session details. A simple acknowledgment alongside your usual credits helps spread the word and supports the continued development of the tool.

Buy ReaClassical Merch

Get print-on-demand ReaClassical merch like shirts, mugs, totes, pins, stickers and more through my TeePublic store!

Source Code

The source code for ReaClassical and ReaClassical Core, the mastering grade ReaClassical plugins, this manual and the website can all be found here. ReaClassical and ReaClassical Core are GPL-3.0 licensed.

Development Style

Due to working on GitHub and releasing the functions via ReaPack, I have the ability to push

bugfixes and new features very quickly into an existing ReaClassical install. Often bugfixes happen within minutes or hours of receiving the report. When I dream up new features, the development often happens in rapid fashion over the course of a few days. However, now that ReaClassical has what I consider a mature feature set, I foresee maintenance and occasional bugfixes becoming more central to the process. This will give me more opportunity to work on this documentation, a complete video tutorial series etc. Part of development is also ensuring that ReaClassical continues to operate as expected with the latest REAPER versions. That's not to say there won't be new features appearing! As the REAPER developers add more new features, I will always check to see what might be useful for ReaClassical.

Versioning

Unlike the full ReaClassical system, ReaClassical Core will not receive as many updates as the full ReaClassical system and these will be mostly limited to bugfixes where necessary, and ensuring that the functions continue to work as expected in future REAPER versions. Therefore, a simpler MAJOR.MINOR versioning has been selected.

Tools and Languages

ReaClassical Core works on top of REAPER, the digital audio workstation and utilizes ReaPack and SWS Extensions. ReaClassical Core functions are coded using Lua. All coding is done either in REAPER's ReaScript Development Environment, vscode or kate on Linux. The online manual is written in AsciiDoc. The ReaClassical Core logo and banner were created in GIMP.

Changelog

The changelog for ReaClassical Core functions can be found by double-clicking on the ReaClassical Core package in ReaPack and navigating to the `History` tab. Whenever you sync ReaPack via **Extensions › ReaPack › Synchronize packages**, this information should also appear automatically.

Installing ReaClassical Core

Requirements: *ReaPack, SWS/S&M extension*

To install ReaClassical Core:

1. Download ReaPack from https://reapack.com/. Read the instructions at https://reapack.com/user-guide for how to install the plugin, add repositories, add packages etc.
2. Import my repository into ReaPack by copying and pasting this link into ReaPack.
3. Search for "SWS" and mark "SWS/S&M extension" for install.
4. Search for "core" in the ReaPack and mark "ReaClassical Core" for install.
5. Click "Apply".

Assigning Keyboard Shortcuts

Unlike the full ReaClassical ecosystem, ReaClassical Core does not provide any keyboard shortcuts and leaves it up to the user to assign as they see fit. The appendix includes suggested shortcuts. Open up the actions menu via keyboard shortcut `?` or by navigating to **Actions › Show action list...** to see the ReaClassical Core functions (you can filter at the top of the screen). You can add your own shortcuts via the box in the bottom left of the actions menu dialog. For reference (from REAPER's own user guide), here is a table of equivalent modifier keys for the different OSes:

Table 1. Shortcut Equivalents

PC (Windows or Linux) Key	Mac (MacOS) Key Equivalent
Shift	Shift
Control (Ctrl)	Command (Cmd)
Alt	Option
Windows	Control

Editing with ReaClassical Core

Recording Audio

If you are recording your music in REAPER, a good idea would be to set the recorded filename pattern to use take numbers. Go to **Options › Preferences** or use `Ctrl` + `P` and set "Recorded Filenames:" to something like `$tracknumber-$track-T$recpass` or `$tracknameornumber-T$recpass`. This way, you can easily identify your takes and also use the ReaClassical Core Find Takes function to search for takes based on underlying filename.

Importing Audio

Read this section if you or someone else recorded using a different DAW or portable recorder.

Either drag the audio into the REAPER window or select **Insert › Media file...**.

For classical music editing purposes outside of the full ReaClassical ecosystem, you should choose a *horizontal* approach by inserting all the takes of a particular channel sequentially on its own track. It makes sense to have the main stereo pair(s) at the top of the project followed by any section pairs and individual spots.

Marking Edits on your Scores

This is best done using a physical, photocopied score by the conductor or lead musician. I advocate for a "T" system where a large T is inserted into the score at the intended edit point. Either side of the T stem, and under the crossbar, the outgoing and incoming take numbers are written. A wavy crossbar indicates some leeway for where the edit point can be placed. Further notes can be

attached underneath the T such as directions for tightening the gap etc.

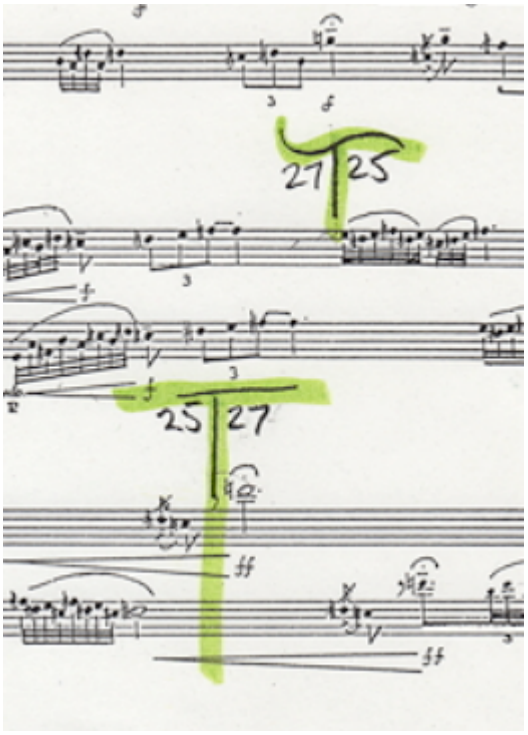

Figure 1. Editing a physical score

ReaClassical Core Preferences

Before starting to edit using ReaClassical Core, it is worth opening the ReaClassical Core preferences window. Here you can set default crossfade length when using the S-D edit functions, the safety length for alerting the user when placing markers close to existing edits, and the option to add S-D markers at mouse hover location (to enable, change the 0 to a 1).

Ready...Steady...Go!

Listed keyboard shortcuts are suggestions based on the full ReaClassical key map.

1. Set destination IN and OUT markers (`1` and `2`)
2. Set source IN and OUT markers (`3` and `4`)
3. Run the S-D Edit function (`5`) to make the source-destination edit (2-, 3- or 4-point edit depending on how many markers you set), or make a time-stretched S-D edit (`F4`) (requires all 4 markers), or if only using destination IN marker and both source markers, an "assembly" or 3-point "insert" edit (`F3`)
4. To finesse the edit, open the REAPER crossfade editor (`F`)

Multi-Project Tab S-D Editing

If you'd like to S-D edit between various open project tabs you can set both the source and destination *project* markers. The markers can exist anywhere on the tab's timeline but perhaps the very beginning or end would be good to keep them out of the way. The S-D editing itself then works just as for a single tab other than any source markers that are set are not deleted to aid quickly

undoing in the destination tab and being ready to reapply the edit. The only rule when using S-D project markers is to ensure that source or destination markers should be paired with the corresponding source or destination project marker. This workflow would, for example, allow you to have multiple project tabs (perhaps one for each symphony movement plus a final *destination* tab), allowing for both internal editing per tab but after setting the S-D project markers compiling the final edit in the *destination* tab. To delete all the S-D project markers run the Delete S-D Project Markers function. Also, in multi-tab S-D editing, when regular markers are placed, any other existing versions in other tabs are automatically deleted to ensure that only one version of the marker exists at a time across all open project tabs.

Further Source-Destination Insights

4-Point Editing

For this operation, set all four markers. Make the edit with the regular ReaClassical Core S-D edit function. This is the most useful edit when dealing with classical music or other acoustic music performed without a metronome.

3-Point Editing

For this operation, set any combination of three markers. Again, run the regular ReaClassical Core S-D edit function. The missing marker is placed according to the distance set by the existing complete pair.

2-Point Editing

For this operation, there are two possibilities: 1) Set one source marker and one destination marker. Make the edit with the regular ReaClassical Core S-D edit function. Any missing IN markers are set to the beginning of the timeline and any missing OUT markers are set to the end of the source or destination material. 2) Set both source markers and no destination markers. Make the edit with . Here, the destination markers are set at the exact same positions on the timeline as the source markers. Obviously this operation is only useful in a vertical editing workflow when you can select source material from a different track group. The usefulness of this second option is further reduced if the takes are not vertically aligned and not virtually identical in tempo. On the other hand, it could be an incredibly quick method for editing takes of a hybrid classical piece that is performed to a click track or other recorded steady beat.

Insert with Time-Stretching

Using the ReaClassical Core_Insert with time-stretching function, you can complete a 4-point edit where the material between the source markers is time-stretched to fit the length of time between the destination markers. This is really useful when the source material has to fit the destination span exactly, for example when working with visual cues. The time-stretch algorithm used will be the one set in REAPER project settings. When there are multiple items in between the source markers, the function will glue the items together before time-stretching. Note that this function can also be used in multi-tab S-D editing mode (see above).

3-point Insert "Assembly Line" Editing

Sometimes you don't necessarily have a best overall take and it is desirable to build the perfect performance linearly, section by section, measure by measure. In this case, set the destination IN markerand set both source markers. Run the ReaClassical_Core 3-point Insert Edit function. A 3-point insert operation will occur and the destination IN marker will jump to the end of the pasted item, ready for the next edit. This means that in order to compile further sections, you now only need set the source markers. If you accidentally move the location of the destination IN marker in the middle of assembly line editing, the function will let you know and offer to move the marker back to the right edge of the latest item in the edit. This will even allow you to do some regular 3- or 4-point editing earlier in the sequence before continuing with the assembly line edits. Just place the destination IN marker anywhere in the project and answer **[No]** when the message box appears. Note that this function can also be used in multi-tab S-D editing mode (see above).

Delete / Delete with Ripple

While perhaps not used as often as 3- and 4-point edits, I have created two functions for deletion of material. Delete & Ripple will delete the material between source IN and OUT markers and ripple material to the right backwards with a short crossfade (`backspace`). Delete Leaving Silence will also delete but maintain the silence without rippling backwards (`ctrl` + `backspace`).

Delete S-D markers

Delete all regular S-D markers!

Crossfade Editor

Now that you've made your precise edits using S/D workflow or razor editing (no worries if it's a bit rough!), it's time to check things through with the help of the crossfade editor view.

Since v7.40, REAPER includes an excellent professional two-lane crossfade editor similar in nature to the specialist classical DAWs such as Sequoia and Pyramix. With the improved crossfade editor, users can see the continued "ghost" waveforms of the items beyond the crossfade they enter and likewise the previous waveforms of the items that exit the crossfade. The ability to visually align transients and then position the crossfade just before it is absolutely critical (and fun when you have the tools to do it!). Select the right-hand item of a fade, use whatever you have set up as the REAPER crossfade editor shortcut (or via **Edit › Crossfade Editor**) and the crossfade window is opened. Note also that you are automatically centered on the crossfade and can use the mouse wheel to zoom in and out. Press the same keyboard shortcut to exit the window.

So, now you are in the crossfade editor mode, my own preferred method of getting the perfect crossfade is to move the transient I want on the left (upper) item to just after the crossfade by dragging on the ghost waveform. Then I drag the "ghost" waveform of my right (lower) item so that the two transients align. That's it! Drag on the active part of the waveform to also move the location of the crossfade. You can just move or resize the crossfade by hovering directly over the lower portion of the crossfade and dragging either the edge or the shaded rectangle.

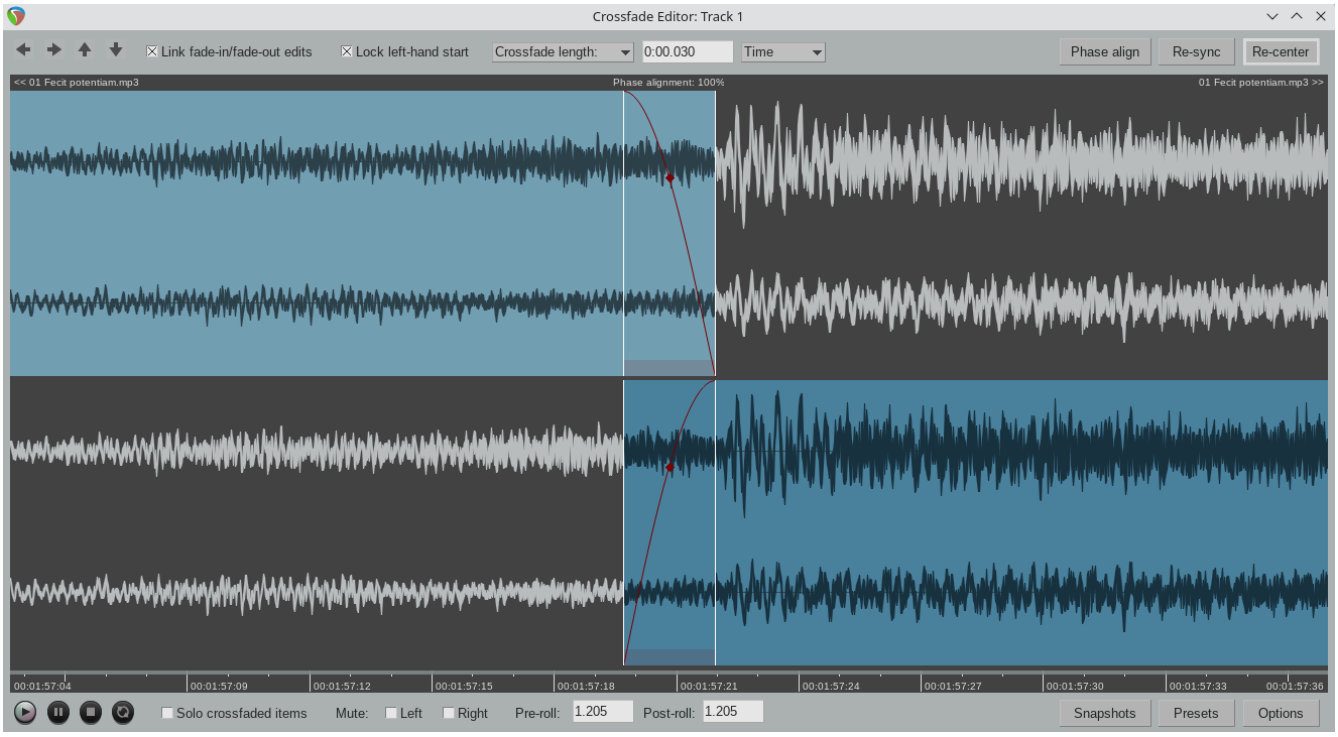

Figure 2. REAPER Crossfade Editor View

In reality, this process can be just a few seconds to achieve the perfect edit. To ensure that previous xfades are unaffected, make sure that *Lock left-hand start* is checked:

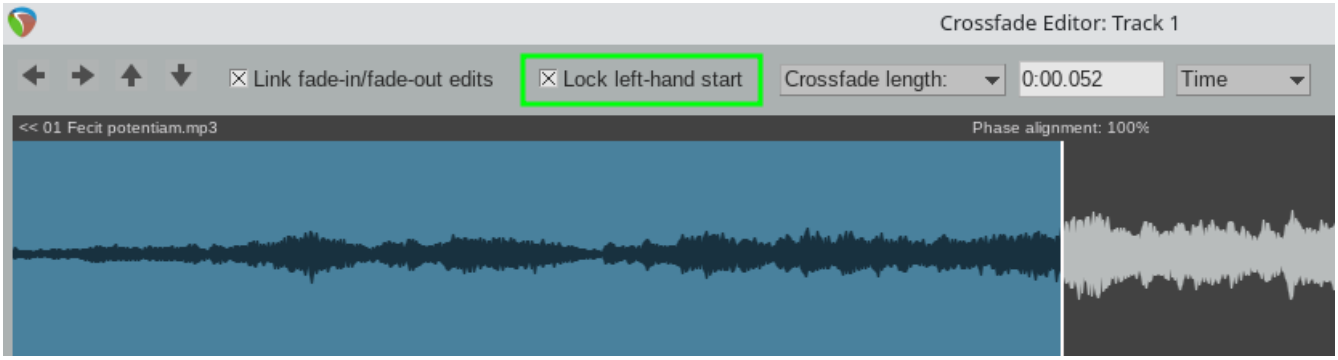

Figure 3. Locking start of left item

For auditioning of material inside the fade editor, phase alignment and all sorts of other advanced view options, please see the REAPER manual for more details of what is possible.

Beyond editing

Once you have successfully edited your material, export your audio stems for your mixing/mastering engineer using the REAPER render dialog. Or, if your engineer is an enlighened being, simply send over a zip of your project folder (ensuring all audio is first moved or copied in via the REAPER project bay window) for them to import into ReaClassical.

End Matter

Thanks

I am appreciative of the collective contributions of the REAPER community with regards the early source-destination actions (Pelleke, in particular), MPL, X-Raym, BirdBird, RCJacH, Sai'ke, Meo-Ada Mespotine, cfillion and many more. Finally, many thanks to Justin and Schwa for such an amazingly versatile DAW.

Closing Thoughts

I hope you enjoy the ability to do serious classical editing with ReaClassical Core on REAPER and that my efforts go some way to making things better, easier and more efficient. Remember that if you find any oddities with the functions or have a bright idea for something new and shiny, please add your thoughts to the dedicated thread on the forums and/or create an issue or start a discussion on my GitHub repository located at https://github.com/chmaha/ReaClassical. If you've had fun using ReaClassical Core, don't forget to check out the full recording, editing, mixing and mastering solution at https://reaclassical.org.

Appendix

Description of ReaClassical Core Functions

Preliminary Note: I include the default ReaClassical keyboard shortcuts as recommendations.

ReaClassical Core Project Preferences `F5`

Description: Set S-D crossfade length, S-D Destination Marker Check Range (ms), Add S-D Markers at Mouse Hover.

Notes: The user can change the checking range (distance beyond an item edge/fade or crossfade) when placing destination IN and OUT markers. Setting to 0 would just check if the marker would be placed *inside* a item fade or crossfade. You can also choose to add S-D markers at the mouse hover position vs edit cursor.

Source-Destination Markers `1`, `2`, `3`, `4`

Description: Add source-destination markers ready for making the edit.

Notes: `1`, `2` = destination IN/OUT; `3`, `4` = Source IN/OUT. In between the destination markers is where the edit will actually happen. In between the source markers is where the material is taken from. Use all four markers for 4-point editing. Use any combination of 3 markers for 3-point editing. Use either 1 source and 1 destination (more useful) or just the source markers for 2-point editing (less useful). By default, the markers are added at the edit cursor location. Additionally, you can set the placement of S-D markers at mouse hover position via ReaClassical Preferences (`F5`). The benefit, other than speed, is that you don't have to worry about making sure source tracks are selected by clicking on the item - simply hover over the desired item and add the source markers!

If you attempt to set one of the destination markers inside of an existing crossfade or within 500ms of a crossfade or item edge, the function will alert you (pressing **[OK]** places the marker anyway).

This helps avoid awkward *sliver* edits that can happen especially if you are zoomed out and placing markers by ear. You can set the check range via ReaClassical Core Project Preferences. The check range (in milliseconds) is the distance beyond an item edge, fade or crossfade. For example, setting to 0 would only check for placement *inside* a fade or crossfade.

S-D Edit `5`

Description: Make the source-destination edit (2-, 3- and 4-point editing)

Notes: Once you have placed your S-D markers use this function to make the edit. Whatever the number of markers you used, the edit will be crossfaded using a short equal power fade (you can change this in REAPER via **Preferences… › Media Item Defaults**) and when using 4 markers, the S-D markers removed ready for the next edit. The edits are perfectly crossfaded and the user only needs to focus on the selection of source material. For any S-D edit you can, of course, undo the action and return to the previous state. The desired crossfade length is set via ReaClassical Core Project Preferences. Note that this function can be used in multi-tab S-D editing mode (see below).

3-point *Assembly Line* Edit `F3`

Description: Uses both source markers and destination-IN marker to compile edits from start to finish.

Notes: The destination-IN marker is retained and placed at the end of the edit so that for subsequent operations the user only has to set the source markers.

If you accidentally move the location of the destination IN marker in the middle of assembly line editing, the function will let you know and offer to move the marker back to the right edge of the latest item in the edit. This will even allow you to do some regular 3- or 4-point editing earlier in the sequence before continuing with the assembly line edits. Just place the destination IN marker anywhere in the project and answer **[No]** when the message box appears. Note that this function can also be used in multi-tab S-D editing mode (see below).

Insert with Timestretching `F4`

Description: Makes the source material fit the the space in between the destination markers using time-stretching vs rippling later material.

Notes: Requires placing all four S-D markers. This is less useful for classical music editing but perhaps very useful for video work where replacement material has to fit exactly into the time used by the original. Crossfade length is set via ReaClassical Project Preferences. Note that this function can also be used in multi-tab S-D editing mode (see below).

Delete with Ripple `Backspace`

Description: Deletes material between the source markers, rippling all later material.

Notes: This only affects the folder group that was highlighted when the source markers were placed which will almost always be the destination group (the top folder in the project). Crossfade length is set via ReaClassical Core Project Preferences.

Delete Leaving Silence `Ctrl` + `Backspace`

Description: Deletes material between the source markers, no rippling of later material.

Notes: This is similar to *Delete with Ripple* but instead leaves a gap where the original source material existed.

Add Source Project Marker `Ctrl` + `Alt` + `3` or `Ctrl` + `Alt` + `4`

Description: Enables multi-tab editing

Notes: To work in multi-tab S-D edit mode, the user is required to use the S-D project markers as a pair, one in the *destination* tab and the other in the *source* tab. In multi-tab S-D edit mode, any source makers placed by the user are not deleted in order to aid a quick undo in the destination tab ready to reapply a tweaked edit.

Add Destination Project Marker `Ctrl` + `Alt` + `1` or `Ctrl` + `Alt` + `2`

Description: Enables multi-tab editing

Notes: To work in multi-tab S-D edit mode, the user is required to use the S-D project markers as a pair, one in the *destination* tab and the other in the *source* tab. In multi-tab S-D edit mode, any source makers placed by the user are not deleted in order to aid a quick undo in the destination tab ready to reapply a tweaked edit.

Delete all S-D Markers `Ctrl` + `Delete`

Description: Deletes source and destination markers.

Notes: Useful if you want to abandon a S-D edit, however, there's also no problem leaving the markers for another time given that the source/destination IN/OUT marker functions replace any existing version of the marker.

Delete all S-D Project Markers `Shift` + `Delete`

Description: Deletes both source and destination project markers

Notes: In order to switch back to regular single-tab S-D editing, it is important to run this command otherwise the S-D edit functions will continue to attempt to find markers elsewhere.

REAPER Crossfade Editor `F`

Description: Opens a two-line crossfade editor view for precise finessing of S-D edits.

Notes: Since v7.40, REAPER includes an excellent professional two-lane crossfade editor similar in nature to the specialist classical DAWs such as Sequoia and Pyramix. Drag on the "ghost" waveforms to align transients and drag on the lower area or edge of the crossfade to move or resize the crossfade itself. Please see the REAPER manual for more details of what is possible including phase alignment and various view options.

You can shuttle between crossfades using the keyboard arrow keys.