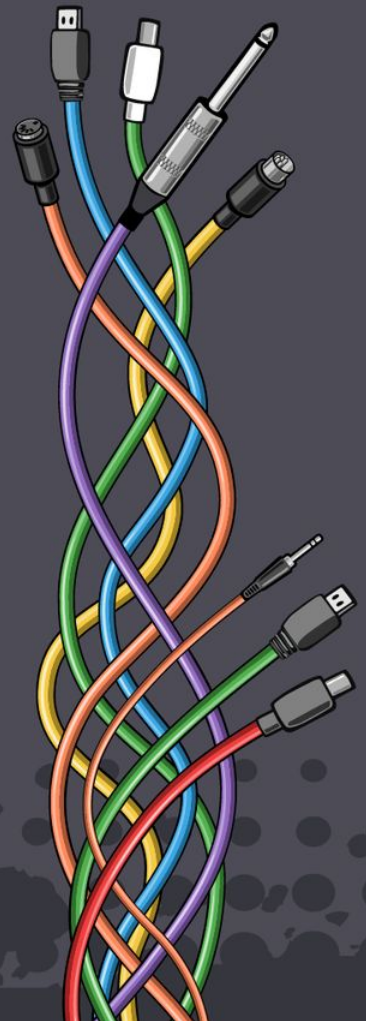


CREATING FROM LEGACY CODE

A CASE STUDY OF PORTING LEGACY CODE FROM EXPONENTIAL AUDIO

HARRIET DRURY

ADC25 Bristol

Introduction

Hello! My name is Harriet 🏴 🏳️‍🌈

- Software Engineer at Native Instruments
- Working on iZotope Products for 2 years
- DSP/ML Focus
- Guitarist
- Can pronounce Cynefin Correctly

Reverbs, Quickly...

http://articles.ircam.fr/textes/Jot97b/

Jean-Marc Jot

Efficient Models for Reverberation and Distance Rendering in Computer Music and Virtual Audio Reality

https://valhalladsp.com/2010/12/21/the-reverb-beard/

The Reverb Beard - Valhalla DSP

Something that I find rather curious, is that many of the reverb pioneers sported some seriously impressive beards.

https://ccrma.stanford.edu/~jos/pasp/pasp.html

PHYSICAL AUDIO SIGNAL PROCESSING

FOR VIRTUAL MUSICAL INSTRUMENTS AND AUDIO EFFECTS

Reverbs, Quickly...

Reverbs, Quickly...

Reverbs, Quickly...

Reverbs, Quickly...

Reverb | Shop for used & new music gear... | Sell your gear | Favourites | Basket | Sign up | Log in

Guitars | Pedals and Amplifiers | Keyboards and Synths | Recording Gear | Drums | DJ and Audio Gear | More Categories | Brands | News | Explore | Artist shops | Reverb Gives | Help Centre

Vintage 1958 EMT 140 valve plate reverb

Second-hand – Good

£4,995

+ £350 Delivery

Buy it now

Add to Basket

Make an offer

Listed: a month ago **Views:** 195 **Watchers:** 24 **Offers:** 0

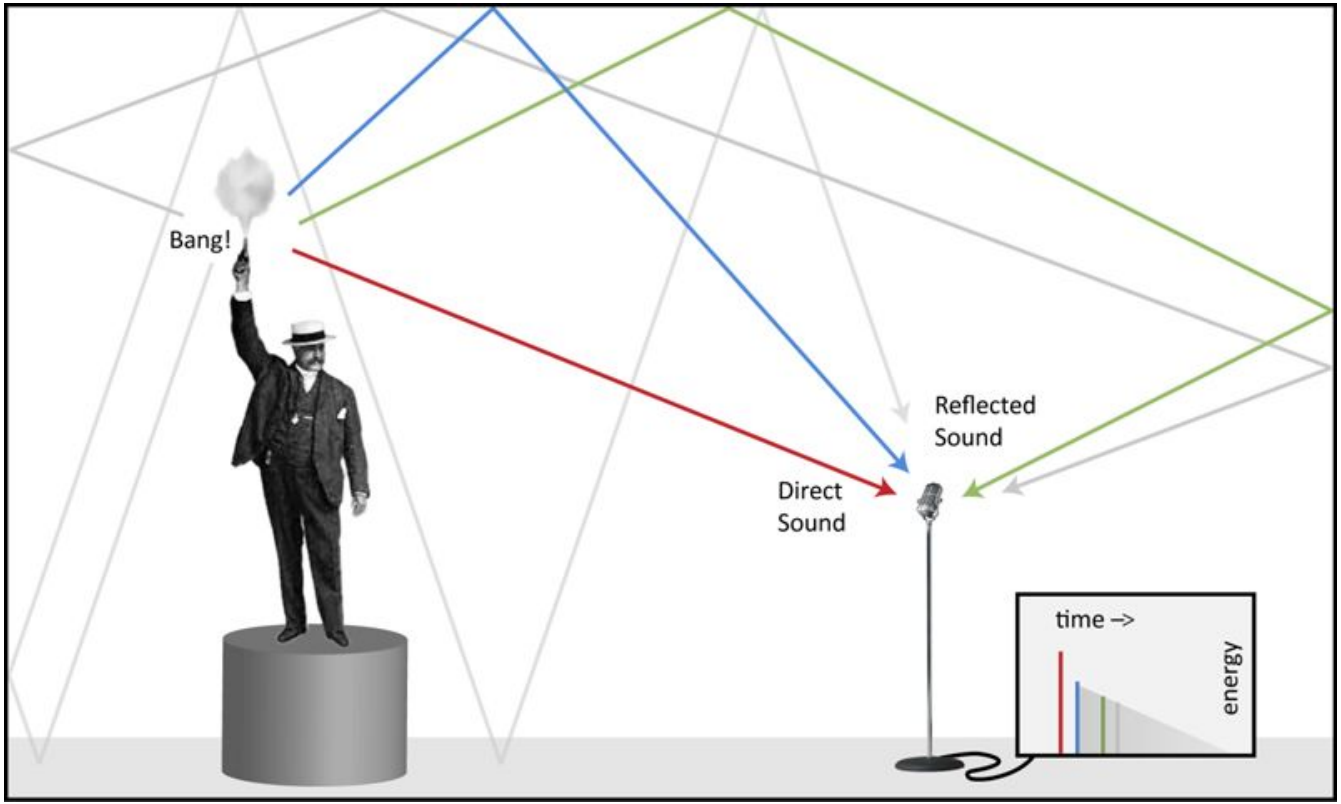
Vintage things ★★★★★

London, United Kingdom

Our users rate Reverb

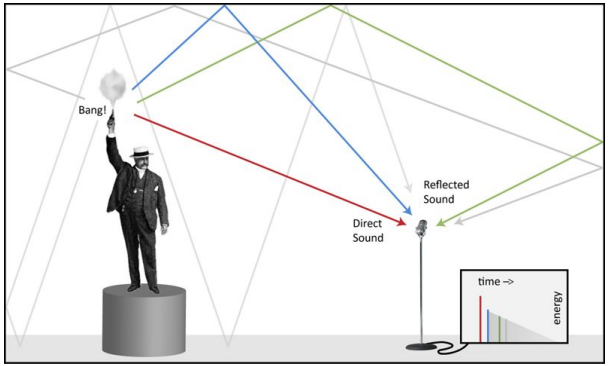
Excellent ★★★★★ Trustpilot

Digital Modelling - Capture Impulse Response (IR)



From: https://www.prosoundweb.com/what-is-an-impulse-response/

Digital Modelling - Capture Impulse Response (IR) & Convolve



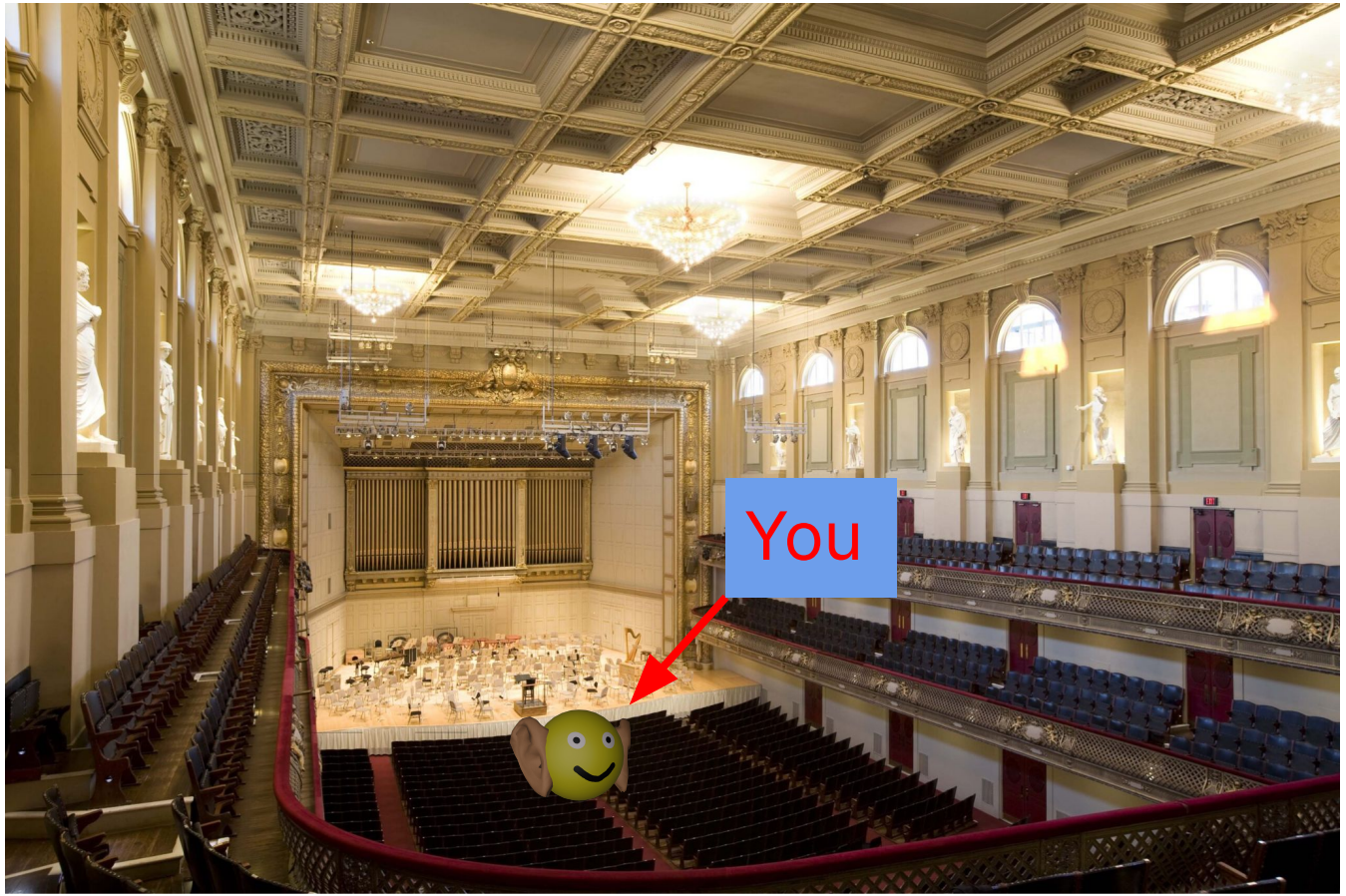
$$y(n) = (h * x)(n)$$

Example Impulse Response Plugin - Altiverb

Example Convolution Plugin - Trash

Digital Modelling - Impulse Response

Digital Modelling - Impulse Response



Digital Modelling - Impulse Response

Digital Modelling - Impulse Response

Digital Modelling - Algorithmic Reverbs

Digital Modelling - Algorithmic Reverbs

An algorithmic reverb is based on a mathematical model that simulates the behavior of a physical space.

- Reflections

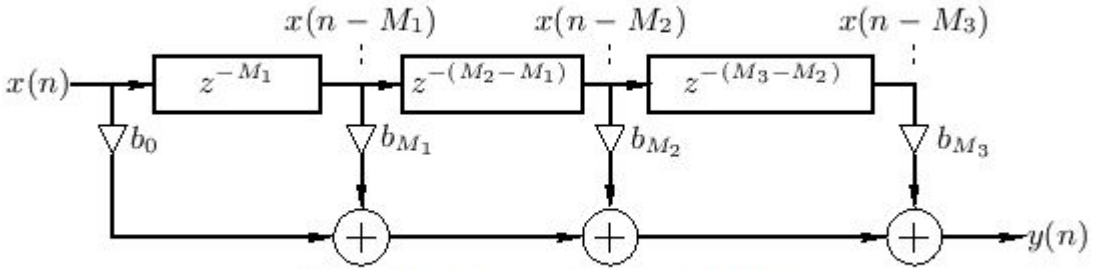
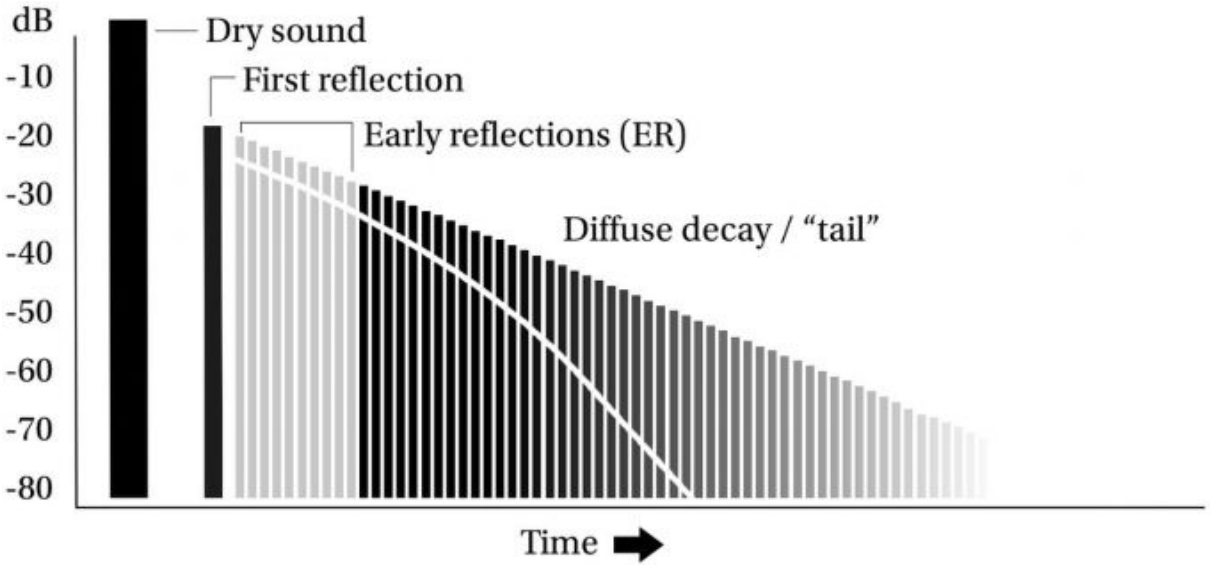
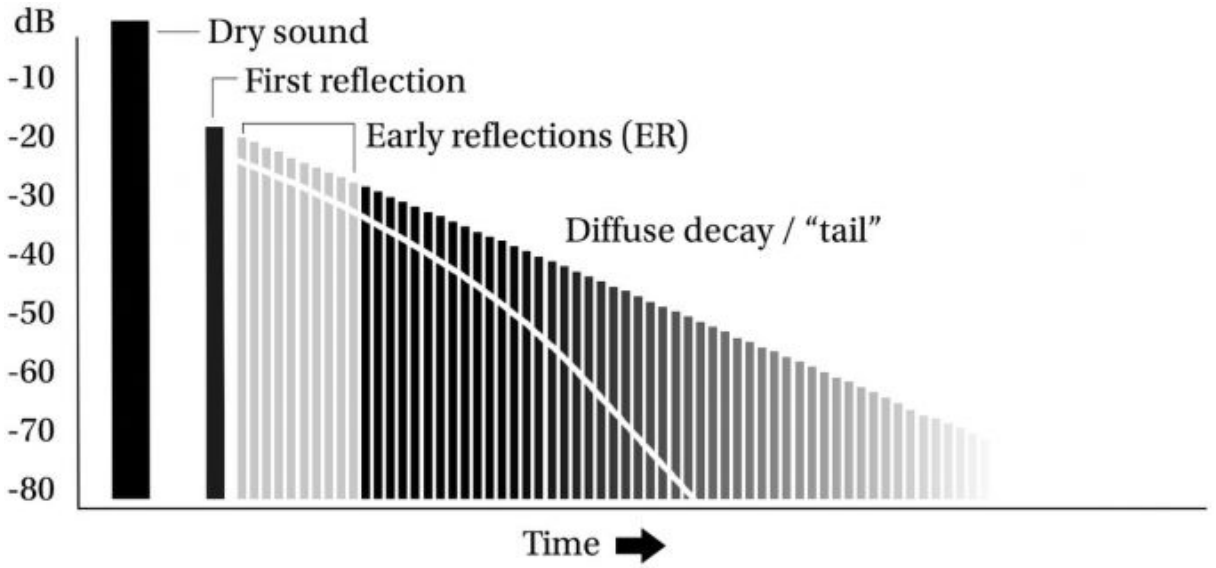


Figure 2.19: Tapped Delay Line (TDL).

- Reflections
- Tail



- Reflections
- Tail
- Diffusion



Lexicon Hardware

Exponential AUDIO

Founded by Michael Carnes in the 2010s

Reverbs based on Lexicon Hardware

Stereo & Surround Plugins

Michael Carnes

Musician, Composer, Engineer, Maker of Great Things

Studied at Boston University and M.I.T

Now retired from engineering

EXPONENTIAL AUDIO

AN iZOTOPE COMPANY

Acquired by iZotope in 2019

M1 support added with UI refresh in 2022

1st Generation

2nd Generation - 2017 Onwards

Stratus

Goals

Purity, no noticeable modulation, completely natural decay.

- Early Reflections similar to Phoenixverb/ R2

Symphony

Goals

Fat tail, character, modulation as a feature

- Ability to ‘Freeze’ a buffer, in perpetuity
- Chorusing Junction

Equinox

Released April 2025

Two Tank Based reverb engines (Stratus & Symphony)

Built by incorporating Exponential Audio DSP into a shared lib in iZotope’s codebase

Equinox

2. **Header**: Contains global controls including the Preset selector.

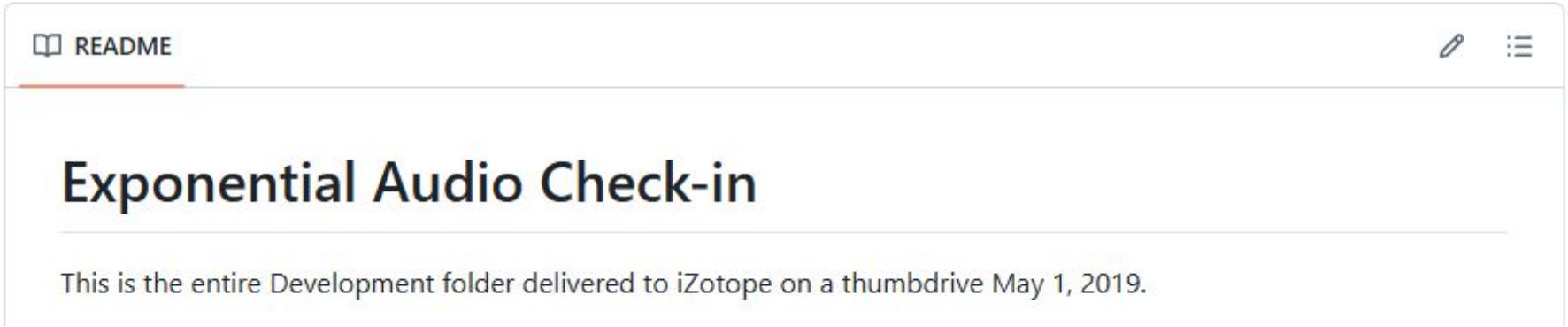
1. **Main controls**: Contains the most important settings.

3. **Additional controls**: Provide detailed settings for the reverb engine.

4. **EQ section**: Lets you adjust the filtering for the input signal, the early reflections, and the tail of the reverb.

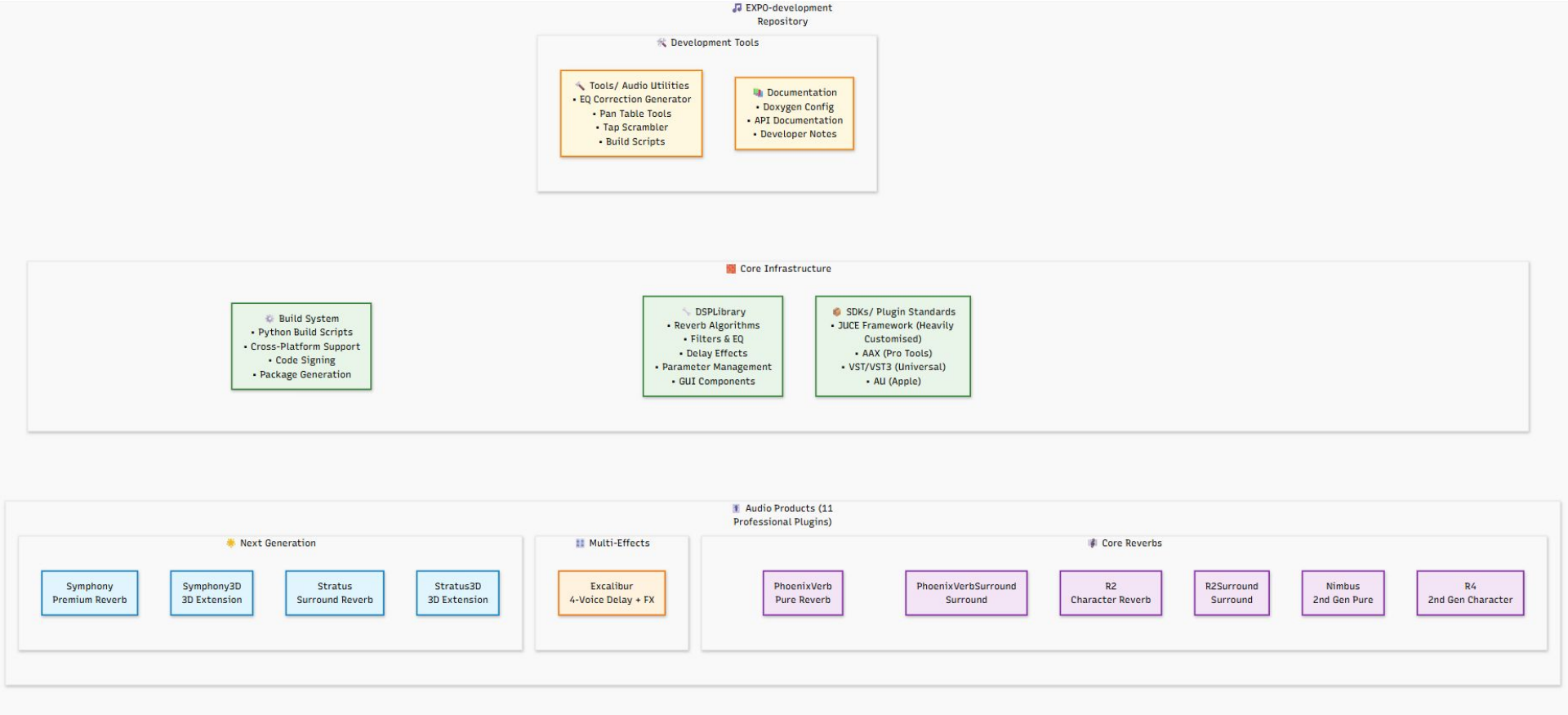
5. **Dynamics Controls**: Lets you apply a saturation, a compression, and a gate to the reverb sound.

Code maintained by someone other than the original author

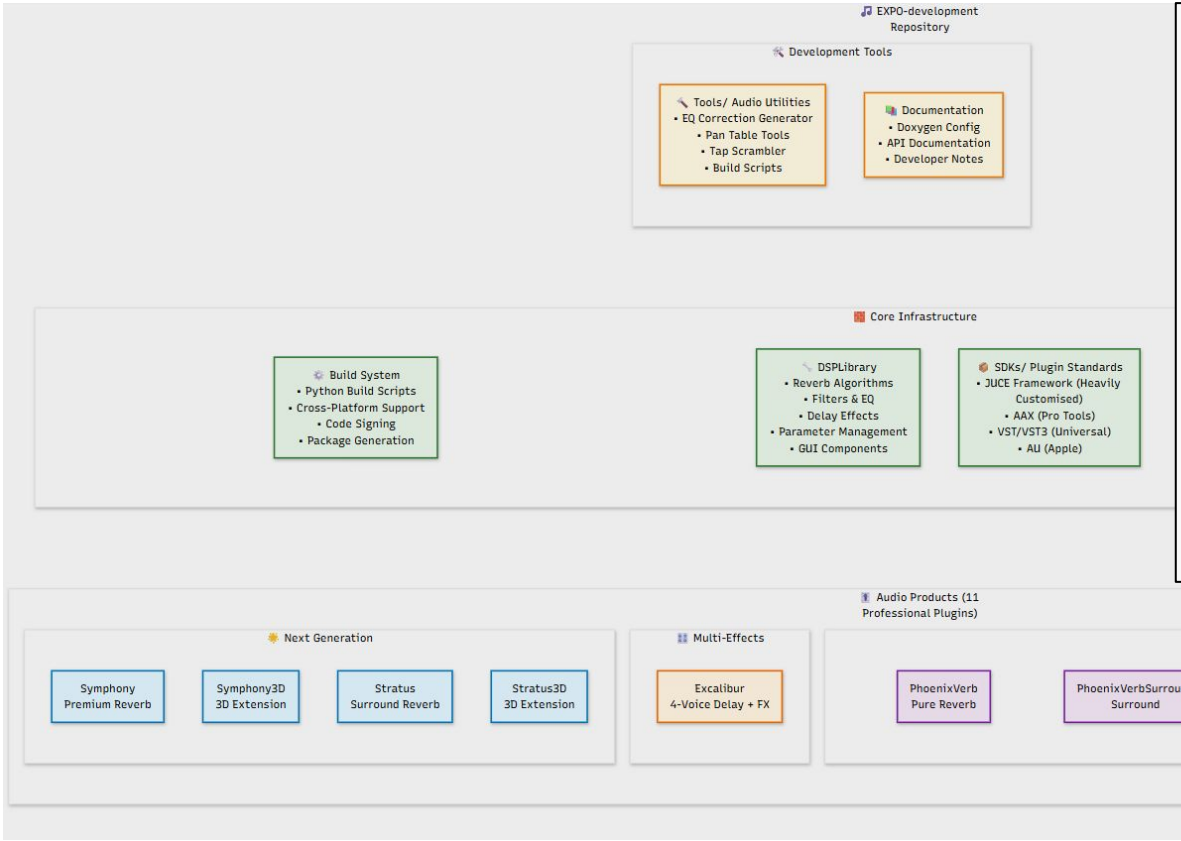


Code maintained by someone other than the original author

Legacy Code


EXPO-development Repository
Development Tools
Tools/ Audio Utilities
• EQ Correction Generator
• Pan Table Tools
• Tap Scrambler
• Build Scripts
Documentation
• Doxygen Config
• API Documentation
• Developer Notes
Core Infrastructure
Build System
• Python Build Scripts
• Cross-Platform Support
• Code Signing
• Package Generation
DSPLibrary
• Reverb Algorithms
• Filters & EQ
• Delay Effects
• Parameter Management
• GUI Components
SDKs/ Plugin Standards
• JUCE Framework (Heavily Customised)
• AAX (Pro Tools)
• VST/VST3 (Universal)
• AU (Apple)
Audio Products (11 Professional Plugins)
Next Generation
Symphony Premium Reverb
Symphony3D 3D Extension
Stratus Surround Reverb
Stratus3D 3D Extension
Multi-Effects
Excalibur 4-Voice Delay + FX
Core Reverbs
PhoenixVerb Pure Reverb
PhoenixVerbSurround Surround
R2 Character Reverb
R2Surround Surround
Nimbus 2nd Gen Pure
R4 2nd Gen Character

Legacy Code



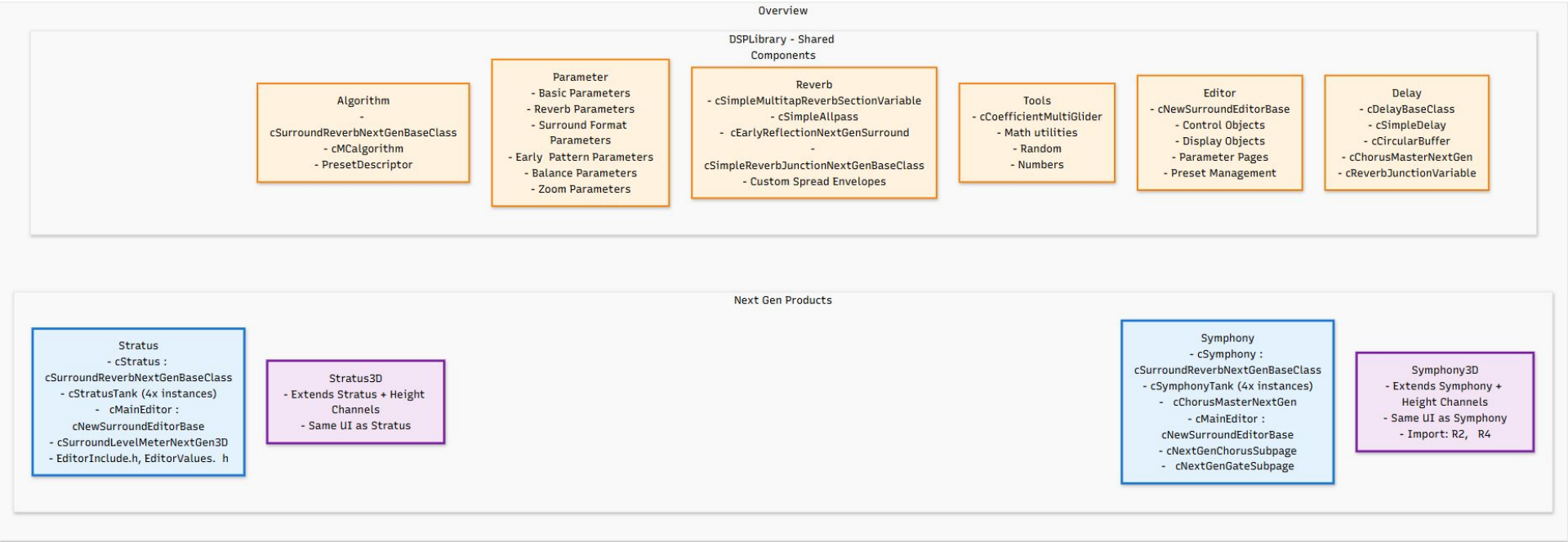
Customised JUCE Dependency

JUCE

Originates from circa 2016

- Custom changes to surround sound routing
- Internal Deps all frozen
- Led to a UI bug when Apple changed an API

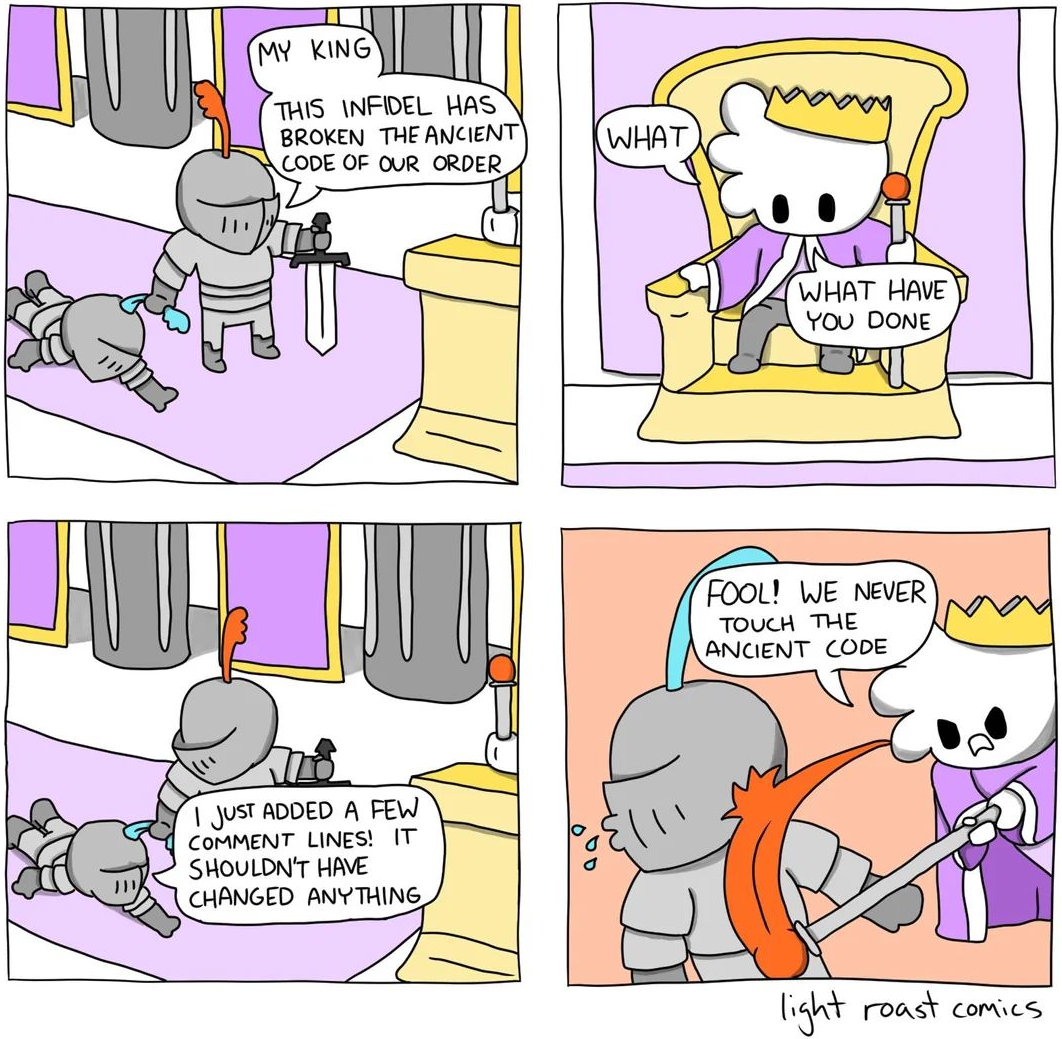
Legacy Code


Overview
DSPLibrary - Shared Components
Algorithm
-
cSurroundReverbNextGenBaseClass
- cMCalgorithm
- PresetDescriptor
Parameter
- Basic Parameters
- Reverb Parameters
- Surround Format Parameters
- Early Pattern Parameters
- Balance Parameters
- Zoom Parameters
Reverb
- cSimpleMultitapReverbSectionVariable
- cSimpleAllpass
- cEarlyReflectionNextGenSurround
-
cSimpleReverbJunctionNextGenBaseClass
- Custom Spread Envelopes
Tools
- cCoefficientMultiGlider
- Math utilities
- Random
- Numbers
Editor
- cNewSurroundEditorBase
- Control Objects
- Display Objects
- Parameter Pages
- Preset Management
Delay
- cDelayBaseClass
- cSimpleDelay
- cCircularBuffer
- cChorusMasterNextGen
- cReverbJunctionVariable
Next Gen Products
Stratus
- cStratus :
cSurroundReverbNextGenBaseClass
- cStratusTank (4x instances)
- cMainEditor :
cNewSurroundEditorBase
- cSurroundLevelMeterNextGen3D
- EditorInclude.h, EditorValues. h
Stratus3D
- Extends Stratus + Height Channels
- Same UI as Stratus
Symphony
- cSymphony :
cSurroundReverbNextGenBaseClass
- cSymphonyTank (4x instances)
- cChorusMasterNextGen
- cMainEditor :
cNewSurroundEditorBase
- cNextGenChorusSubpage
- cNextGenGateSubpage
Symphony3D
- Extends Symphony + Height Channels
- Same UI as Symphony
- Import: R2, R4

Legacy Code - Summary

- One Monolithic Repo
- Customised Framework Dependencies
- DSP shared between products

Legacy Code



Source:
https://www.reddit.com/r/ProgrammerHumor/comments/9xat04/the_ancient_code/

How can we future proof the DSP written by Michael Carnes?

Legacy Code

1. Continue Building as is

Legacy Code

1. Continue Building as is

Pros

- Keeps revenue stream alive
- Branding stays consistent
- Respected code - if it ain't broke...

Cons

- Slow release cycles
- obsolete technology
- Bugs/ instability
- There are only certain team members who understand it

2. Build as a Shared Lib?

Legacy Code

2. Build as a Shared Lib?

Pros

- Quick(ish)
- Clear API Boundary

Cons

- Version Hell
- You have to deploy a DLL
- There is no ‘clean’ DSP Layer
- No User Benefit

3. Extract DSP, into a new, shared repo

Legacy Code

3. Extract DSP, into a new, shared repo

Pros

- Preserve existing DSP/ Value
- Future Proof the Technology
- Cross Product Potential
- Performance Optimisation Opportunities
- Looked at By More Devs

Cons

- The Most Time Consuming Option
- Risk: Untangling 15+ years of coupled code introduces bugs
- API Design Complexity - how *do* you cleanly abstract an inherited class?

Legacy Code

3. Extract DSP, into a new, shared repo

Pros

- Preserve existing DSP/ Value
- Future Proof the Technology
- Cross Product Potential
- Performance Optimisation Opportunities
- Looked at By More Devs

Cons

- The Most Time Consuming Option
- Risk: Untangling 15+ years of coupled code introduces bugs
- API Design Complexity - how *do* you cleanly abstract an inherited class?

Legacy

3. Extra

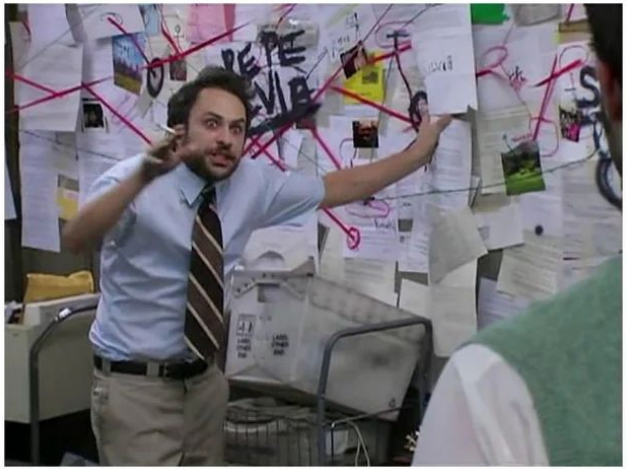

Neoverb
1-MIDI - Neoverb
Reverb
Reflections
31
Space
Plate
1.10
Space
Hall
2.50
Space
-3.0 dB
-3.0 dB
Aurora
Solo Reverb
Bypass
Presets
Hall
0
Reverb
10
Pre-Delay
2.00
Decay
Width
Dark
Tone
Bright
dB
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
Unmask
10 ms
Attack
300 ms
Release
50 %
Sensitivity
20 Hz
50
100
200
500
1k
2k
5k
10k
Pre EQ
Reverb EQ
Hz
100
300
1k
3k
10k
Freq 100 Hz
Gain 0.0 dB
Q 0.7
Dialogue Match
Snapshots
REFERENCE
1
Select a reference and click Capture
or
Load a reference profile
Capture
Clear
APPLY TO
2
Select a clip to apply processing to and click Capture.
Capture
Clear
EQ
100
Amount
Reverb
-inf
Wet
0
Dry
Ambience
0
Trim
Advanced

Phoenixverb, the DSP repo

README

PhoenixVerb

Me explaining my variable naming scheme to the other devs



In this repo, we have reimplemented Exponential Audio Reverbs with minimal changes using iZotope DSP APIs.

Named after the first reverb to be ported, phoenixverb

DM-3: Commit Initial PhoenixVerb boilerplate #1

Merged

evan-lynch_ntvinst merged 5 commits into `master` from `DM-3-setup-shared-dsp-phoneixverb-repo` on May 9, 2019

Active development since 2019

Contributions from many NI devs over the last 6 years

Phoenixverb, the DSP repo

Increase mantainability of ported expo DSP

Usable across product repos

Testable, being able to use iZotope testing frameworks

Create an interface to expo DSP that matches pre existing iZo ideas

Replace old school, C styled code

Utilise modern C++ best practices

Create a bypass on the iZo side

Aux outputs

What did we have to Audit to add expo features?

Legacy code for 1 person vs the company code

Why Port?

Old JUCE Version (Lead to a UI Bug)

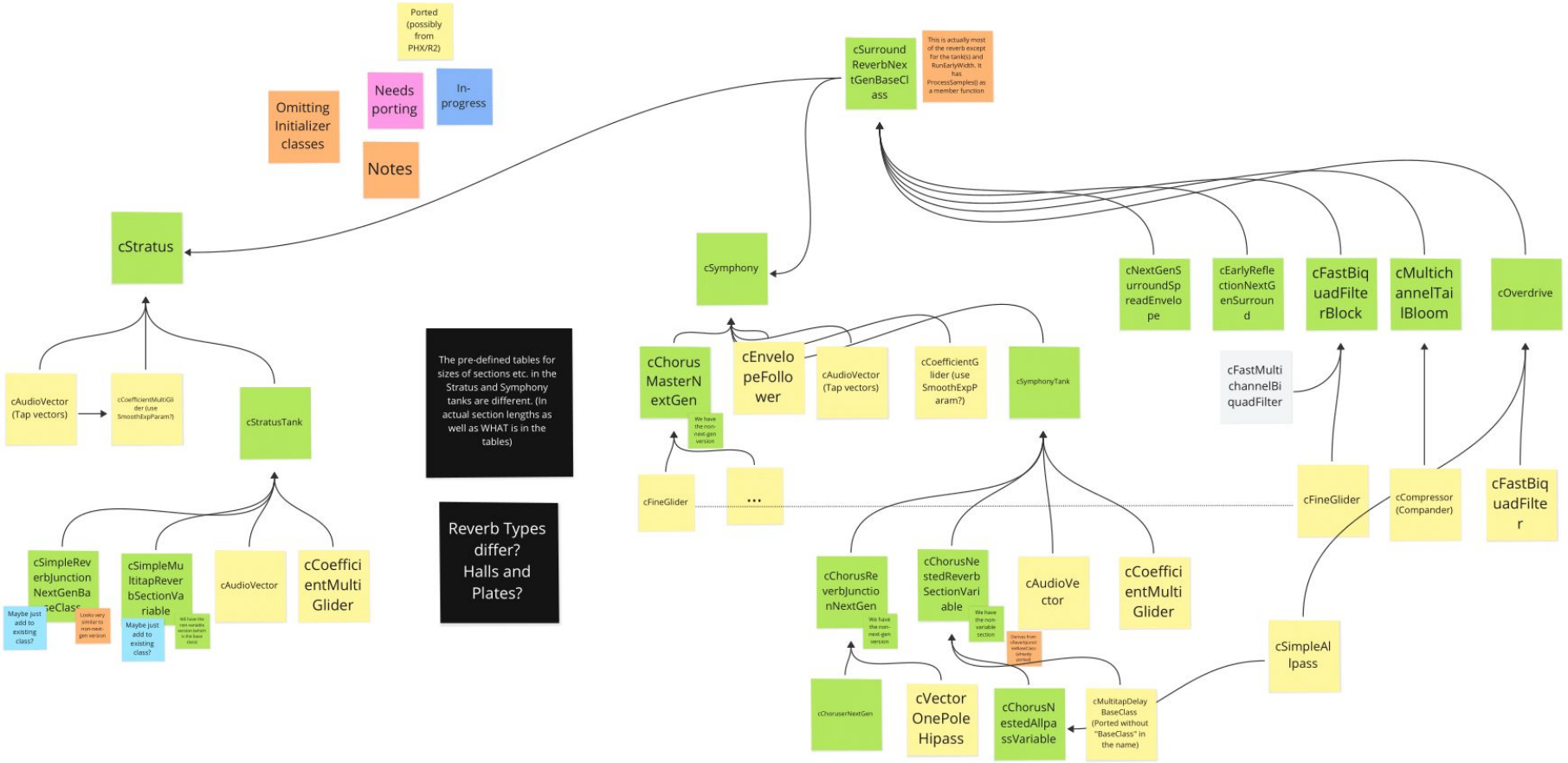
Monolithic Repo of all Exponential Audio Code

Pre C++17

Harriet Drury 3:51 PM

Happy Monday! I guess today is day 0 of the izo-expo work.

Stratus/ Symphony Next Gen Code Porting - TODOs



Stratus/ Symphony Next Gen Code Porting - The Game Plan

- Get Stratus Ported First
- Divide Work by Class
- Discuss Shared Code as a Team & Decide on Basic Class Layouts
- Add TDSP/ test Coverage
- Sit Back, Relax, Enjoy no Bugs or Misfortunes

General Class Porting TODOs

Commit `d7acb70`

harriet-drury_ntvinst committed on Jan 2

```
Initial Commit of MC's Code
```

- Initial Commit
- Namespace the Class
- Rename Members/ Variables/ Class
- Modernise Loop Unrolling (Vectorize Block Processors)
- Collapse Multi Stage Init

Commit `d7acb70`

harriet-drury_ntvinst committed on Jan 2

```
Initial Commit of MC's Code
```

NOTE: We are not trying to change the class structure of Stratus & Symphony, the goal is to keep all the specific quirks & features of the code.

Commit `d7acb70`

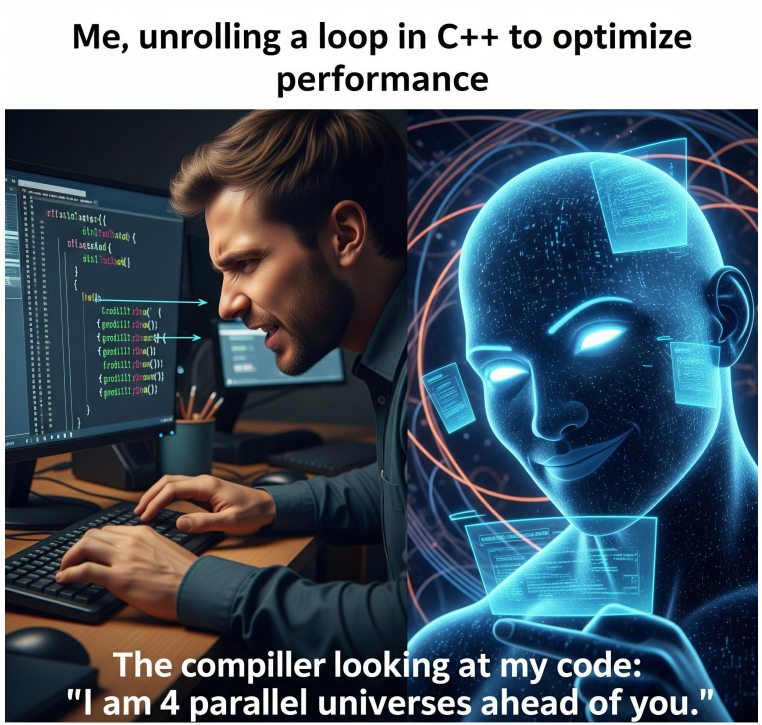
harriet-drury_ntvinst committed on Jan 2

```
Initial Commit of MC's Code
```

NOTE: We are not trying to change the class structure of Stratus & Symphony, the goal is to keep all the specific quirks & features of the code.

In fact acted as a engine to discussion around design patterns & code design

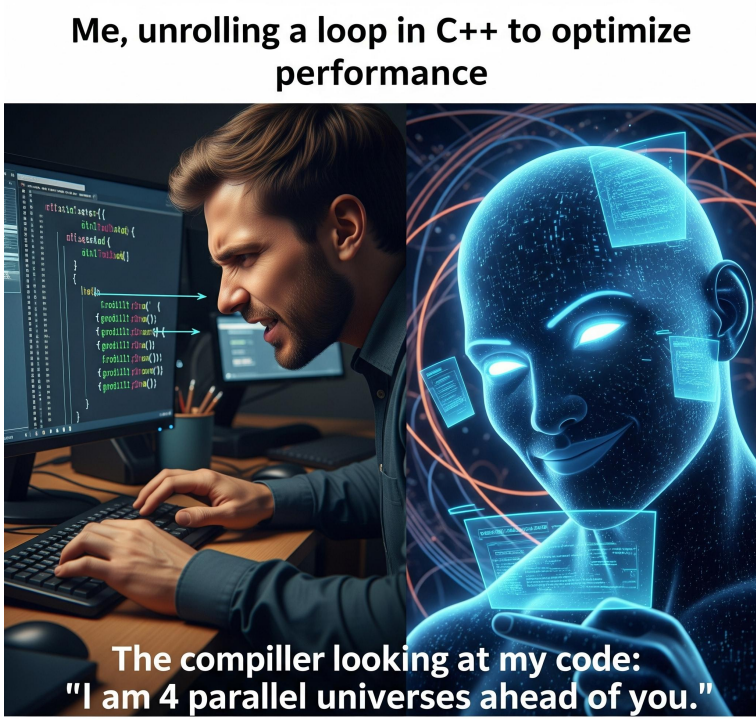
Code Adjustment - Loop Unrolling



Code Adjustment - Loop Unrolling/ Auto Vectorization

Loop unrolling is an optimization technique where the body of a loop is duplicated multiple times within a single iteration, reducing the number of loop control operations (like incrementing the index and checking the loop condition). This can improve performance by:

- Decreasing the overhead of loop control instructions.
- Allowing the compiler to better optimize code, especially for SIMD/vector instructions.
- Improving instruction-level parallelism and cache usage.



Code Adjustment - Loop Unrolling, MC's Code (Example, Not Actual Code)

```
switch (GlobalPluginEnvironment.VectorSize) {
case SMALL_VECTOR_SIZE:
    for (int i = 0; i < SMALL_VECTOR_SIZE; ++i)
        process(i);
    break;
case DOUBLE_VECTOR_SIZE:
    for (int i = 0; i < DOUBLE_VECTOR_SIZE; ++i)
        process(i);
    break;
}
```

Code Adjustment - Loop Unrolling, Templated

```
template <int VectorSize>
void RunAllpass(float* input, float* output) {
    for (int i = 0; i < VectorSize; ++i)
        process(i);
}
```

Code Adjustment - Loop Unrolling

Aspect	Original Code	Updated Vectorized Code
Loop Structure	Switch/case for each vector size	Template-based, block processing
Code Duplication	High (repeated loops)	Low (single loop, reused via templates)
Buffer Size Handling	Tied to DAW/host buffer size	Fixed internal block size, DAW-agnostic
Maintenance	Harder (add/change in multiple places)	Easier (change in one place)
Compiler Optimization	Fixed-size loops, but repetitive code	Fixed-size, template, unrolled

```
// Unrolls & Vectorises when using (MSVC) /O2 & (GCC/
Clang) -O3

#include <vector>
void xyz(std::vector<float> a, std::vector<float> b)
{
    for (size_t i = 0; i < 3; i++) {
        a[i] += b[i];
    }
};
```

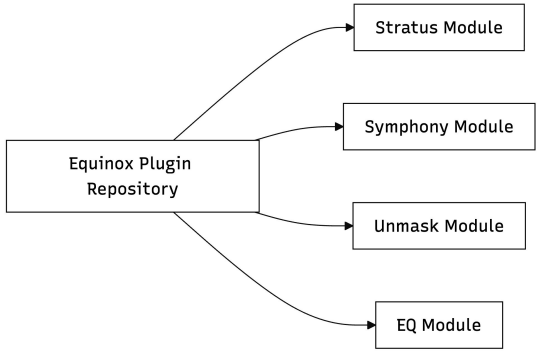
https://godbolt.org/z/8hxzx88T6

August - March - Porting & Equinox Plugin Bringup

[🎨 ART] Expo-232 create stratus plug in with suites bp #1

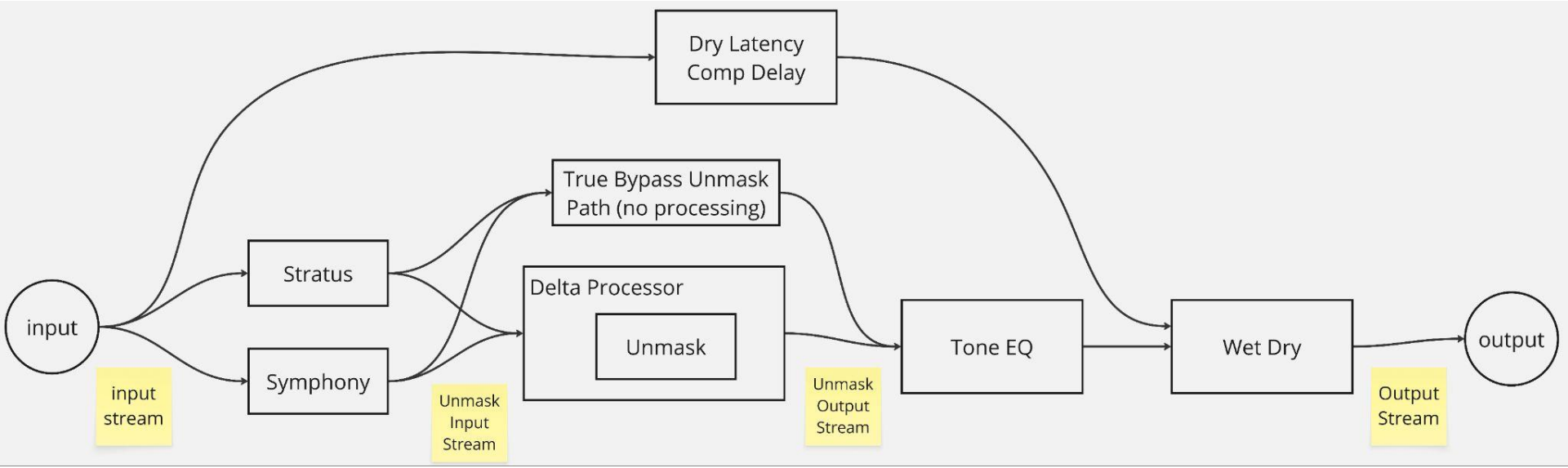
Merged harriet-drury_nt... merged 97 commits into `develop` from `EXPO-232-Create-Stratus-Plug-in-with-SuitesBP` on Aug 23, 2024

March - We Ship a Beta with Adaptive Unmasking & EQ Added!



- We have Stratus & Symphony DSP in one plugin
- We have started adding our DSP to the Plugin (Adaptive Unmasking & EQ)
- A Placeholder UI Appears!
- The bugs start……

March - We Ship a Beta with Adaptive Unmasking & EQ Added!



- We have Stratus & Symphony DSP in one plugin
- We have started adding our DSP to the Plugin (Adaptive Unmasking & EQ)
- A Placeholder UI Appears!
- The bugs start……

Bug Report! Our Tail is Crackling...

Projects / Exponential Audio / Add parent / EXPO-491

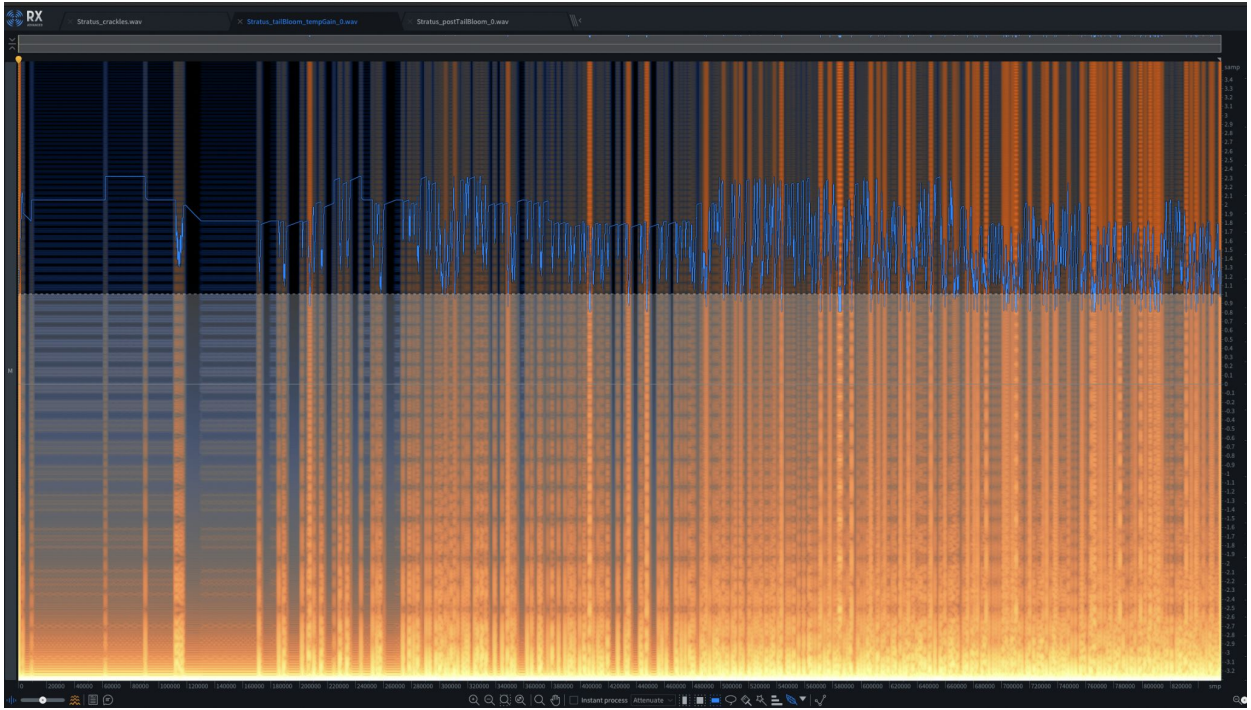
Natural algorithms crackling at high reverb times with large or small size

What did we do to change this behaviour?

What we knew:

- Stratus Specific
- Occurs Reliably with Reverb Size set to max (or min), & Reverb Time set to max
- Turning down the Early Reflections level makes crackling in the tail more obvious

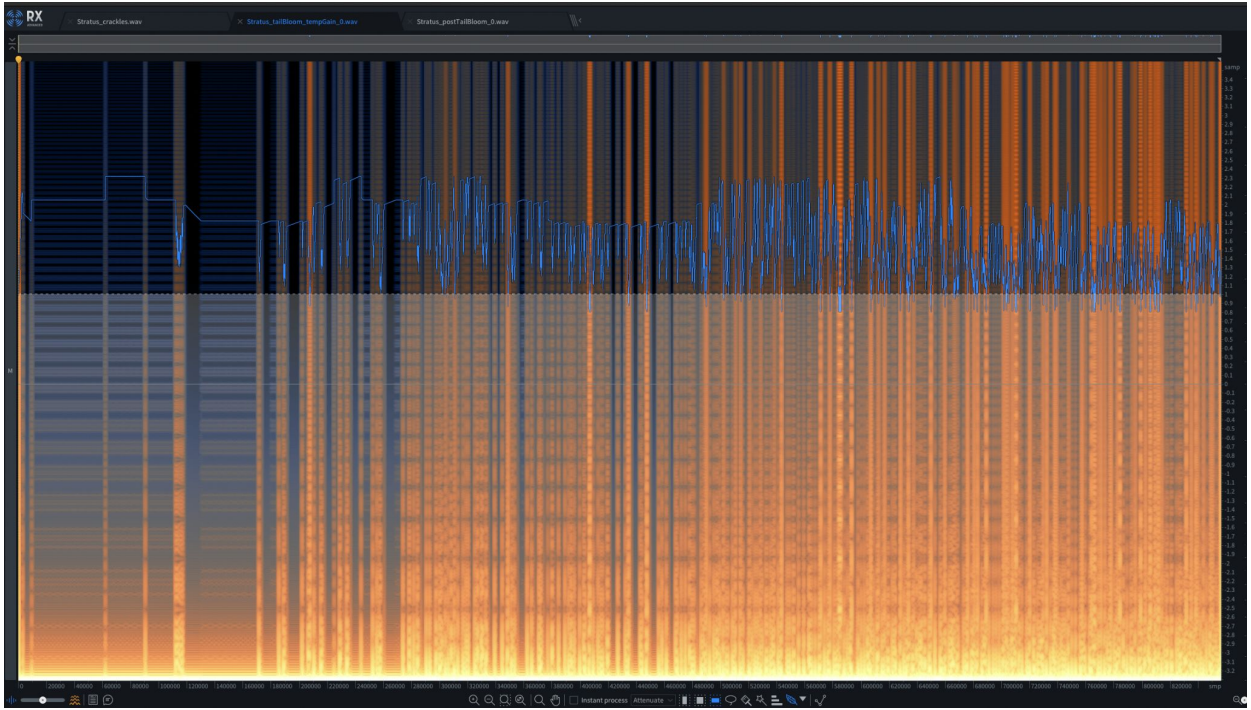
Bug Report! Our Tail is Crackling...



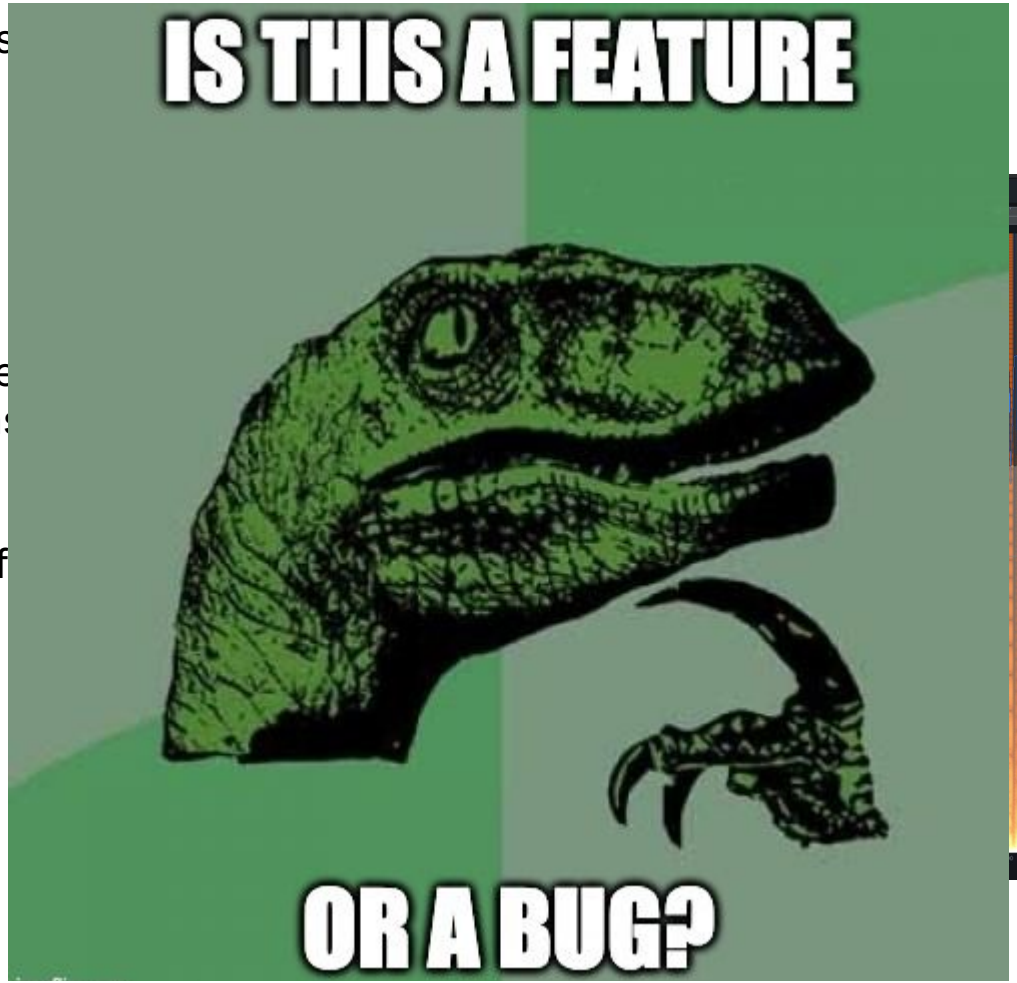
Bug Report! Our Tail is Crackling...

In some reverb modes, the gain could surpass `1.0f`

This does not differ from the original code

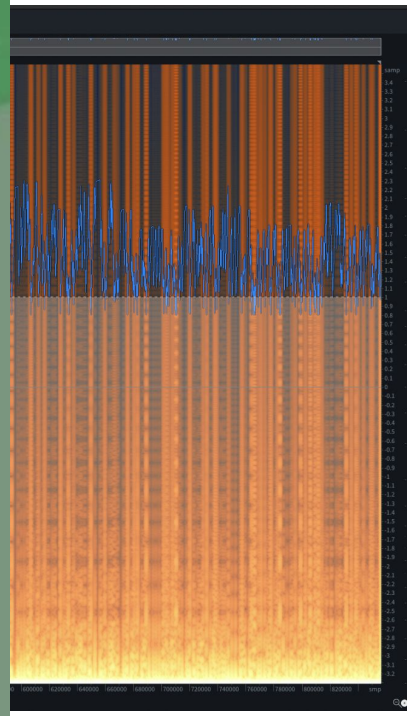


Bug Report! Our Tail is



In some reverb mode
the gain could surpas
`1.0f`

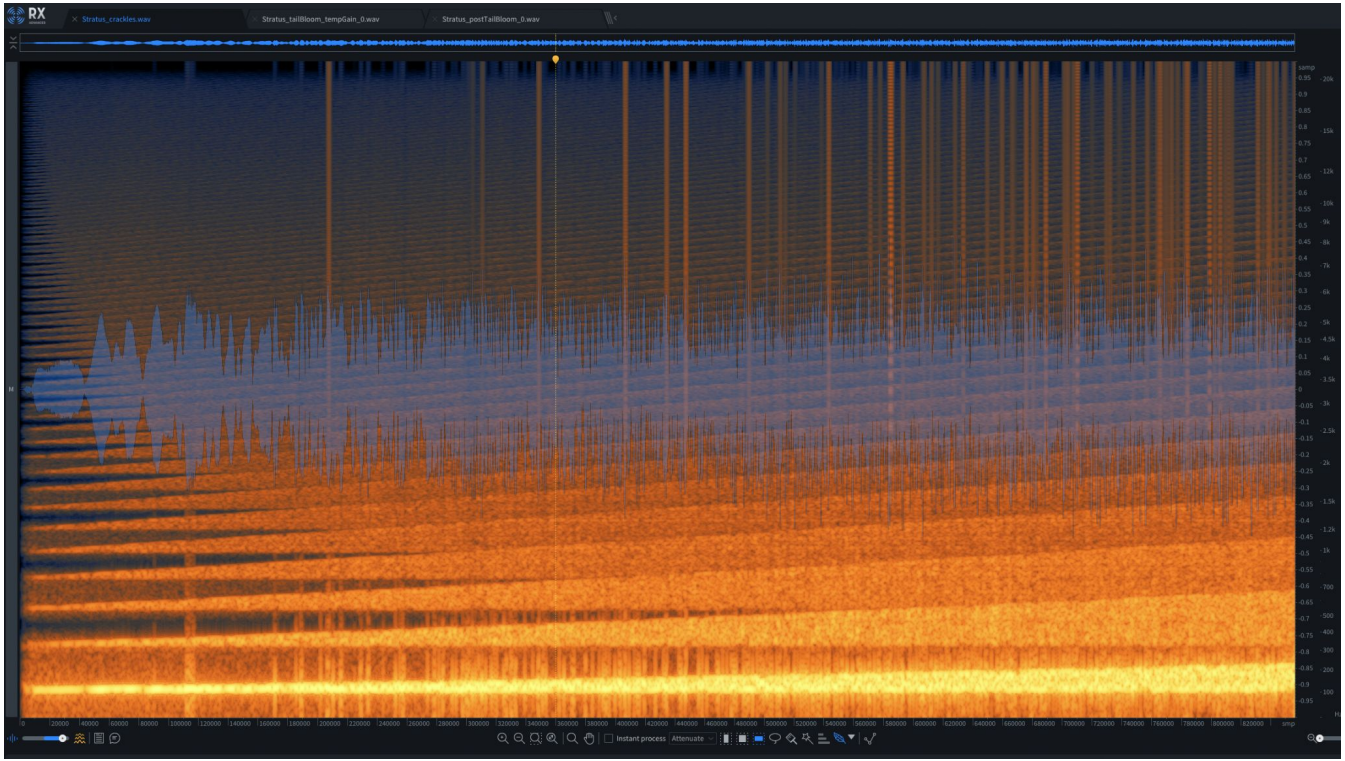
This does not differ f
the original code



Solution, `std::max`

For this `TailBloom` class, we clamped gains. The high energy remains in places

...as a feature, not a bug



April 2025 - Equinox Launches

Conclusion - We Built a Reverb Plugin!

Between August 2024 - April 2025, a team of 6 developers and 3 QAs:

- Ported Legacy Reverb Engines and implemented a seamless UI for switching between the two.
- Created a new preset system for thousands of presets.
- Extended Multi-Channel Config Support (7.x.2, 7.x.6, 9.x.4, etc.).
- Included EQ & Adaptive Unmasking tech.
- Numerous other improvements (Seeded Deterministic Randomness, Stabilised Exploding Filters, Legacy Session Conversion, etc.).
- Retired Stratus & Symphony, keeping the DSP alive in a future proofed DSP repository

Thank you!

Thanks to my Talk Collaborators:
Roth Michaels
Alex Fink
Jefferson Hobbs
All of the ART Team & Audio Production Team! 
Emma Fitzmaurice 

Harriet Drury
harriet.drury@native-instruments.com