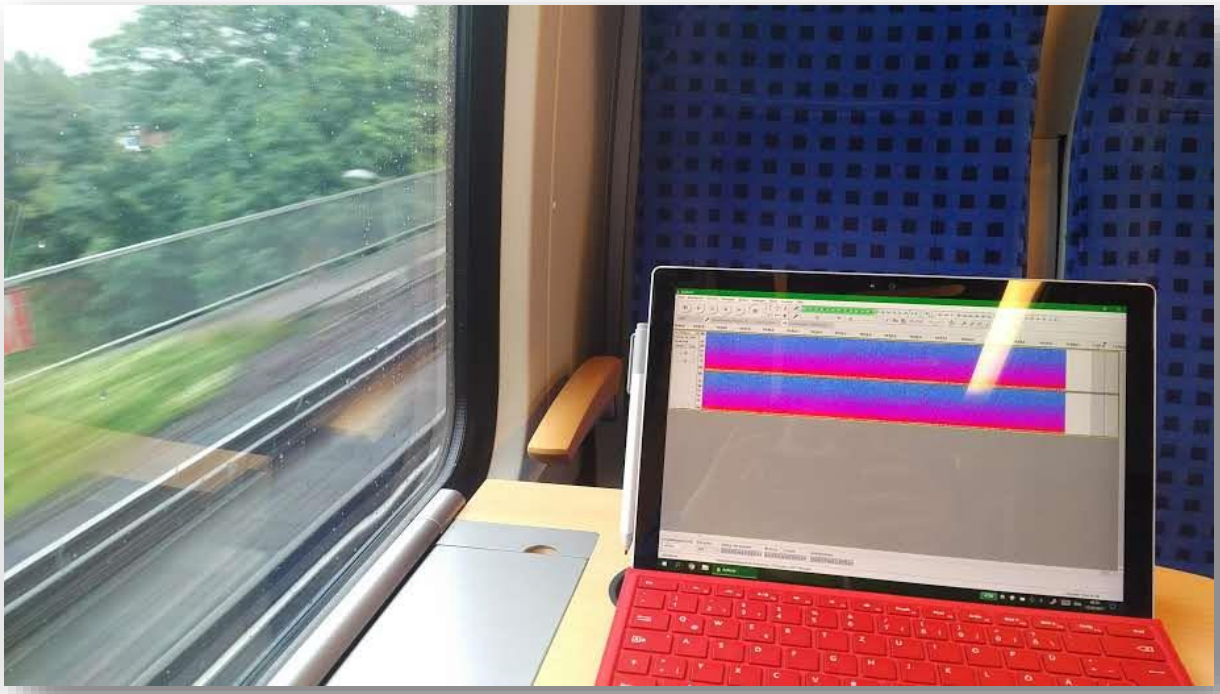
An Introduction to Soundmodding in Train Simulator

By Linus Follert



There are lots of different ways to modify the sound of a train in Train Simulator. This is my way.

This work is licensed under Creative Commons CC BY-SA 2.0. More information on the last page.

Contents

1 **The Basics of Train Simulator** 4
2 **Recording and editing Sounds** 9
2.1 Equipment 9
2.2 Recording 9
2.3 Software 10
2.4 Editing Oneshots 10
2.5 Editing Loops 11
3 **The proxyxml File** 14
<Audio> 14
<Curve> 17
<ModifierChain> 19
<InstanceGroup> 20
<Loop> and <Oneshot> 21
<Loop> 21
<Oneshot> 22
How to find your Controller Name 25
4 **How to use the blueprint editor (for starting a sound from scratch)** 27
5 **Some useful Tips and Links** 28
6 **Epilogue** 29

1 The Basics of Train Simulator

This Chapter is going to deal with the basic folder structure of the simulator so you know where everything is and what is actually necessary to edit the sound of a train.

Let's first have a look in the main directory of railworks

ocal Disk (C:) › Program Files (x86) › Steam › steamapps › common › RailWorks Search RailWorks

Name	Date modified	Type	Size
AddIns	06/05/2017 00:29	File folder	
Assets	25/05/2017 19:07	File folder	
Content	17/05/2017 17:52	File folder	
data	06/05/2017 00:29	File folder	
dev	06/05/2017 00:29	File folder	
Dialogs	06/05/2017 00:29	File folder	
Flyouts	06/05/2017 00:29	File folder	
Install	06/05/2017 00:29	File folder	
Languages	06/05/2017 00:29	File folder	
Manuals	06/05/2017 00:29	File folder	
Options	06/05/2017 00:29	File folder	
PackageInfo	06/05/2017 00:29	File folder	
plugins	06/05/2017 00:29	File folder	
Screens	06/05/2017 00:29	File folder	
Temp	22/05/2017 20:13	File folder	
ApplyXSL.exe	06/05/2017 00:14	Application	21 KB
BlueprintEditor2.exe	06/05/2017 00:15	Application	797 KB
config.xml	06/05/2017 00:15	XML File	2 KB
config_fullscreen.xml	06/05/2017 00:15	XML File	1 KB
ConvertDDSToTG.bat	06/05/2017 00:15	Windows Batch File	1 KB
ConvertToDav.exe	06/05/2017 00:15	Application	41 KB
ConvertToGEO.exe	06/05/2017 00:15	Application	837 KB
ConvertToTG.exe	06/05/2017 00:15	Application	84 KB
ConvertToTGLib.dll	06/05/2017 00:15	Application extens...	539 KB
cudart32_30_9.dll	06/05/2017 00:15	Application extens...	284 KB
d3dx9_26.dll	06/05/2017 00:15	Application extens...	2,244 KB
dbghelp.dll	06/05/2017 00:16	Application extens...	1,185 KB
IAOptimiser.exe	06/05/2017 00:16	Application	31 KB
ICSharpCode.SharpZipLib.dll	06/05/2017 00:16	Application extens...	188 KB
installscript.vdf	06/05/2017 00:16	VDF File	2 KB
libmmd.dll	06/05/2017 00:16	Application extens...	3,972 KB
LocalisedStrings.dll	06/05/2017 00:16	Application extens...	28 KB
LogMate.exe	06/05/2017 00:16	Application	63 KB
luac.bat	06/05/2017 00:16	Windows Batch File	1 KB

This is how the main directory looks like after a fresh install of railworks. You can find it by going into your steam installation folder then *steamapps/common/RailWorks* .

Most of these folders are unimportant for us. You may already have looked into the *Manuals* folder. This is quite obviously where all the manuals are stored. The other three folders that are of interest for users are the *Install*, *Contents* and *Assets* Folder.

The *Install* Folder holds the install files for third party programs that the simulator uses (e.g.: DirectX and Microsoft Redistributable). They may come in handy when the simulator encounters a problem.

The *Contents* Folder holds the information and metadata for Routes and Scenarios. It for example tells a route where a signal stands and in addition points to the *Assets* folder where the textures, etc. can be found.

Now for the most important folder. The *Assets* Folder holds all Textures, Scripts and everything that you can visually see in the simulator. In Addition, all the sound files and scripts are kept here. This will be the folder we will be taking a closer look at next.

Let us open it and see what's there.

Name	Date modified	Type	Size
3DTrains	06/05/2017 01:46	File folder	
3DZUG	06/05/2017 01:48	File folder	
103Tf	06/05/2017 01:48	File folder	
143er	06/05/2017 01:48	File folder	
AbsolutesChaoz	06/05/2017 01:48	File folder	
Acorn	06/05/2017 01:48	File folder	
Aerosoft	06/05/2017 01:51	File folder	
Albert	05/06/2017 09:56	File folder	
Allaboard	06/05/2017 01:53	File folder	
ALn668 GTT	06/05/2017 01:53	File folder	
Amisia	28/05/2017 17:30	File folder	
Andi_M	06/05/2017 01:54	File folder	
AndiS	06/05/2017 01:54	File folder	
AndreasHoff	06/05/2017 01:54	File folder	
AP	06/05/2017 01:58	File folder	
AP_Waggonz	06/05/2017 02:03	File folder	
ArtOfMotion	06/05/2017 02:04	File folder	
astauder	06/05/2017 02:04	File folder	
Attix	06/05/2017 02:04	File folder	
Bab	06/05/2017 02:04	File folder	
Barnez	06/05/2017 02:04	File folder	
Barrett	06/05/2017 02:05	File folder	
BeeKay	06/05/2017 02:05	File folder	
Belette65	06/05/2017 02:06	File folder	
BerlinTrains	06/05/2017 02:06	File folder	
BernardDeschamp	06/05/2017 02:06	File folder	
BlackChart	06/05/2017 02:06	File folder	
Brendan	06/05/2017 02:06	File folder	
Carthago	06/05/2017 02:06	File folder	

This is how my Asset Folder looks like. Don't be scared. Yours may look different. As Mentioned above you can find your trains and Route Assets here. Depending on how much DLC and freeware you have installed you may have more or less folders here.

Every Developer can make their own folder and point to the individual files in their script. But we will get to that in Chapter 3.

A quick example will show you where you can find the sound files. Let's say we want to alter the sound of the Class 378. The first step is to find the folder the developer has created for the use of his train. In our case that would be DTG, so go ahead open that folder and search for the 378 Pack

Class143Pack03	06/05/2017 01:28	File folder
Class158Pack01	06/05/2017 01:28	File folder
Class180Pack01	06/05/2017 01:28	File folder
Class375Pack01	06/05/2017 01:29	File folder
Class378Pack01	06/05/2017 01:29	File folder
Class442Pack01	05/06/2017 12:26	File folder
Class455Pack02	06/05/2017 01:29	File folder
Class801Pack01	06/05/2017 01:30	File folder
CologneKoblenz	08/06/2017 19:37	File folder
ExeterKingswear	06/05/2017 01:33	File folder

And would you look at that in the *DTG* Folder we found our train.

The Next step is to find a folder called “Audio” or “Sound” which is where we can find our sound files.

Name	Date modified	Type	Size
Class378Pack01Assets.ap	06/01/2016 21:49	AP File	115,129 KB

If you find an *.ap file follow these steps to gain access. This is necessary when you want to edit the trains sounds as this makes it easier.

1) Get yourself a program that can open zip files (WinRAR, 7Zip, etc.)

2) Now you should be able to open the *.ap file with the previously installed program

3) Copy all the contents out of the ap file and into the 378Pack Directory

The end result should look like this:

Name	Date modified	Type	Size
Audio	12/06/2017 19:44	File folder	
InputMappers	12/06/2017 19:44	File folder	
PreLoad	12/06/2017 19:44	File folder	
RailVehicles	12/06/2017 19:44	File folder	
_.sbn	06/11/2014 17:42	SBN File	1 KB
_.sbn.MD5	06/11/2014 17:42	MD5 File	1 KB
Class378Pack01Assets.ap	06/01/2016 21:49	AP File	115,129 KB
de.lan	22/10/2015 15:42	LAN File	53 KB
en.lan	22/10/2015 15:42	LAN File	49 KB
fr.lan	22/10/2015 15:42	LAN File	56 KB

Now you can open the *Audio* Folder which should hold all the relevant sound files.

Markers	12/06/2017 19:44	File folder
RailVehicles	12/06/2017 19:44	File folder

The Markers Folder is unimportant to us. It shows the sim where the sounds are emitted from (e.g. a fan sound should come from the fans and not the doors, etc.)

After you open The *RailVehicles* Folder you will find these

Bogies	12/06/2017 19:44	File folder
Coupling	12/06/2017 19:44	File folder
Derailing	12/06/2017 19:44	File folder
Electric	12/06/2017 19:44	File folder

Remember this is just an example. Each developer can set their own folder names and can structure the Audio files differently. There are even changes between DTG products.

The *Bogies* Folder will hold the sounds for the Bogies the *Coupling* Folder for the Couplings and so on. Pretty self-explanatory.

The *Electric* Folder in this case holds the important files including engine sounds and Cab sounds including button, engine, and horn sounds.

Go ahead and open it then open the *Class 378* Folder that is somehow there.

Name	Date modified	Type	Size
Cab	12/06/2017 19:44	File folder	
Class 378 Bogie Sound.bin	15/01/2015 13:00	BIN File	3 KB
Class 378 Motor Sound.bin	15/01/2015 13:01	BIN File	3 KB
Class378AirCon.proxybin	13/11/2014 15:01	PROXYBIN File	3 KB
Class378AirCon.proxyxml	13/11/2014 15:00	PROXYXML File	11 KB
Class378AirConSound.bin	11/11/2014 15:32	BIN File	3 KB
Class378Couple.proxybin	13/11/2014 15:01	PROXYBIN File	2 KB
Class378Couple.proxyxml	13/11/2014 15:00	PROXYXML File	4 KB
Class378CoupleSound.bin	11/11/2014 15:32	BIN File	3 KB
Class378DMSO.proxybin	13/10/2015 12:16	PROXYBIN File	6 KB
Class378DMSO.proxyxml	13/10/2015 12:16	PROXYXML File	25 KB
Class378DMSOSound.bin	11/11/2014 15:32	BIN File	3 KB
Class378MOS.proxybin	13/11/2014 15:01	PROXYBIN File	3 KB
Class378MOS.proxyxml	13/11/2014 15:00	PROXYXML File	10 KB
Class378MOSSound.bin	11/11/2014 15:32	BIN File	3 KB
Class378Motors.proxybin	13/11/2014 15:01	PROXYBIN File	4 KB
Class378Motors.proxyxml	13/11/2014 15:00	PROXYXML File	20 KB
Class378MotorsSound.bin	11/11/2014 15:32	BIN File	3 KB
Class378PTSO.proxybin	13/11/2014 15:01	PROXYBIN File	4 KB
Class378PTSO.proxyxml	13/11/2014 15:00	PROXYXML File	13 KB
Class378PTSOSound.bin	11/11/2014 15:32	BIN File	3 KB
Class378TSO.proxybin	13/11/2014 15:01	PROXYBIN File	3 KB
Class378TSO.proxyxml	13/11/2014 15:00	PROXYXML File	9 KB
Class378TSOSound.bin	11/11/2014 15:32	BIN File	3 KB
DTGClass378AirCon.dav	11/11/2014 15:32	DAV File	218 KB
DTGClass378AirConOFF.dav	11/11/2014 15:32	DAV File	89 KB
DTGClass378BrakeApply.dav	13/10/2015 12:11	DAV File	51 KB
DTGClass378BrakeReleaseLong.dav	13/10/2015 12:11	DAV File	422 KB
DTGClass378BrakeReleaseMedium.dav	13/10/2015 12:11	DAV File	191 KB
DTGClass378BrakeReleaseShort.dav	13/10/2015 12:11	DAV File	121 KB
DTGClass378BrakeSqueal.dav	11/11/2014 15:32	DAV File	141 KB

Now we can see the important bits. You will see script files and sound files. The script files are the files that end with *.bin / *.proxybin / *.proxyxml . Important for us are the *.proxyxml files. We will use those later in chapter three to detailed change the audio of a train.

The sound files are encoded in a *.dav format. The sim has an internal “player” and will play the *.dav files over that. Having a *.wav file works just as well except the sim will use an external source to play the file. I have never had issues with that but you may choose what you prefer. You will see later that it may come in handy to have a *.wav file instead of a *.dav. If you want you can go ahead and replace a sound to try it out. If you have a *.wav file you need to convert that into *.dav first. There are a few tools available that do that. Google (or whatever) is your friend.

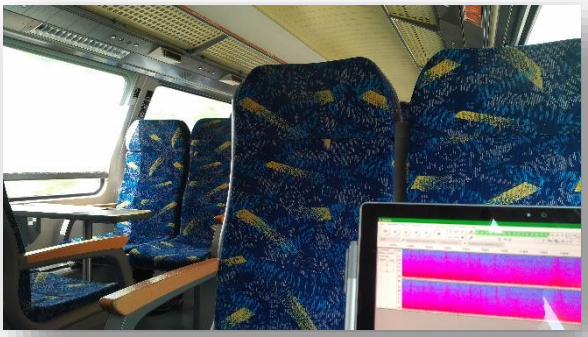
That covers the basics of audio editing. Skip to Chapter three to find out how to alter the *.proxyxml files.

2 Recording and editing Sounds

This Chapter will deal with the recording of sounds, what Equipment to use and how best to record. It will also tell you what software is best and what to look out for when editing. If you already got some sound files lying around, lucky you, you can skip this chapter.

2.1 Equipment

Start off by asking yourself a simple question. What kind of equipment do you have (laptop, mobile phone, dedicated microphone)? If you have an actual expensive microphone you want to be using that since it can give you the best results. Personally, I use my laptop (Surface Pro 4) which for me is quite enough. The quality is passable and not too bad. If you also have a modern laptop it should be fine. Try it out. Preferably try it out at home by putting a YouTube video of some sound recording on and see how your device performs. That can already tell you quite a lot.



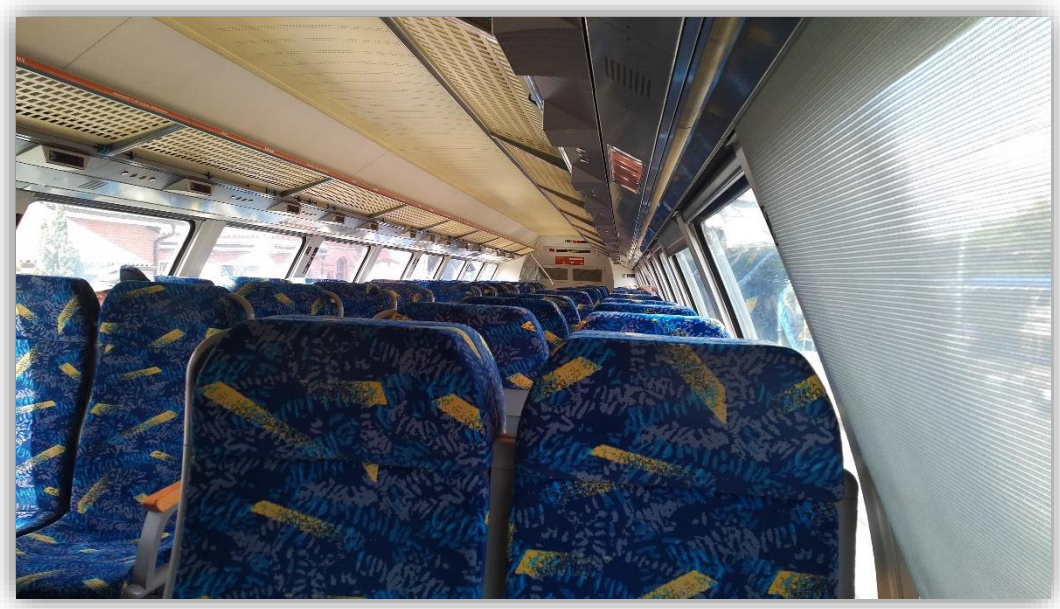
Now to the question of mobiles phones. They can sometimes be used for sounds that play once and don't loop (e.g. the sound of a train door) but if you need looping sounds for example the engine or just simple driving sounds from the bogies you should use something more professional than your mobile phone. This is really up to you. Try and see what works best.

2.2 Recording

So, you now have your device of choice and are ready to head on a train to record some bits. First choose what you want to record. If you want some fan sounds you should probably sit next to a fan. If you want to record the engine noise you should sit in the motor carriage. If you feel like recording the Bogie Sounds it is not advisable to sit in the middle of the carriage. Whatever your needs are you should look where to sit.

Next look out that there are not many people around you that would talk into your recording. A few things to look out for:

1) Don’t record in the middle of rush hour. Leaving aside the weird stares you are going to get (yes that’s a thing. Trust me) this should be obvious advice. Try to catch a train between 9AM and 11AM or just drive really late on a Sunday.
2) If you don’t happen to have an incredible amount of time on your hands maybe try and sit in the front or back of the train depending on the stations your train is calling at. Here in Germany it is quite common that people don’t like to walk very far on the platform. You might want to know your stations. Know where entrances and exits are and then sit on the opposite site.
3) Sometimes it can be beneficial to get into an all stations stopping service (Regionalbahn) rather than a fast service (Regionalexpress). Who wants to sit in a train that stops every 2-3 minutes right? Right. Plus, this gives you a bigger ‘harvest’ in your noise endeavour. Maybe you want to record the noise of the brakes!? Can’t do that if the train never stops.



As an example. I sat in the very front on a Saturday at 10 AM

2.3 Software

The type of Software you use depends on your device. If you have an Android phone just use the camera app. If you use your laptop and don’t have some expensive software you should use Audacity. It’s a free Audio Editing software and we are also going to use it for editing out the sounds later on. I will not go into using the program. There are enough tutorials out there. But it’s not very difficult.

Once you recorded everything you can go home and move to the editing part.

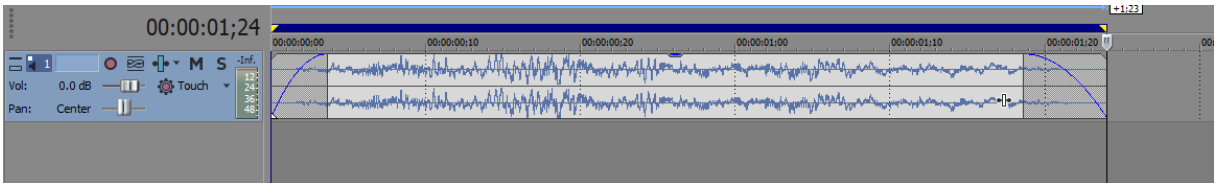
2.4 Editing Oneshots

Oneshots are sounds that the sim will play once and not in a loop. For example, the door sounds, the horn or the noise of the bogie driving over a junction.

Again, this is not going to be a tutorial on how to use a specific program. I am just going to give you some tips on what to look out for. I myself not only use Audacity but also Sony Vegas. They both have about the same capabilities when it comes to audio editing.

When you have your thirty minutes or so recording edit out the sound you want. A Oneshot is not very difficult and you do not need to look out for much stuff. Here are some tips:

First thing you want to do is fade your sounds in and out to prevent weird klicks and bumps later in the sim. You'll see what I mean. Its already enough to fade the sounds a few fractions of a second to remove these artefacts.

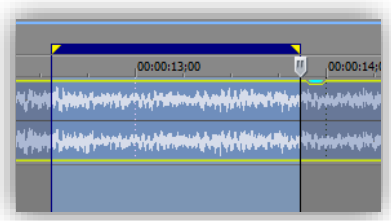


As you can see hear the sound is only one second long. Fading just a little will do quite a lot. But it's best if you try that yourself later.

When rendering the file. Render it as *.wav so that the sim can play it and so that you can convert it to *.dav later if you wish. This is important for both Oneshots and Loops.

2.5 Editing Loops

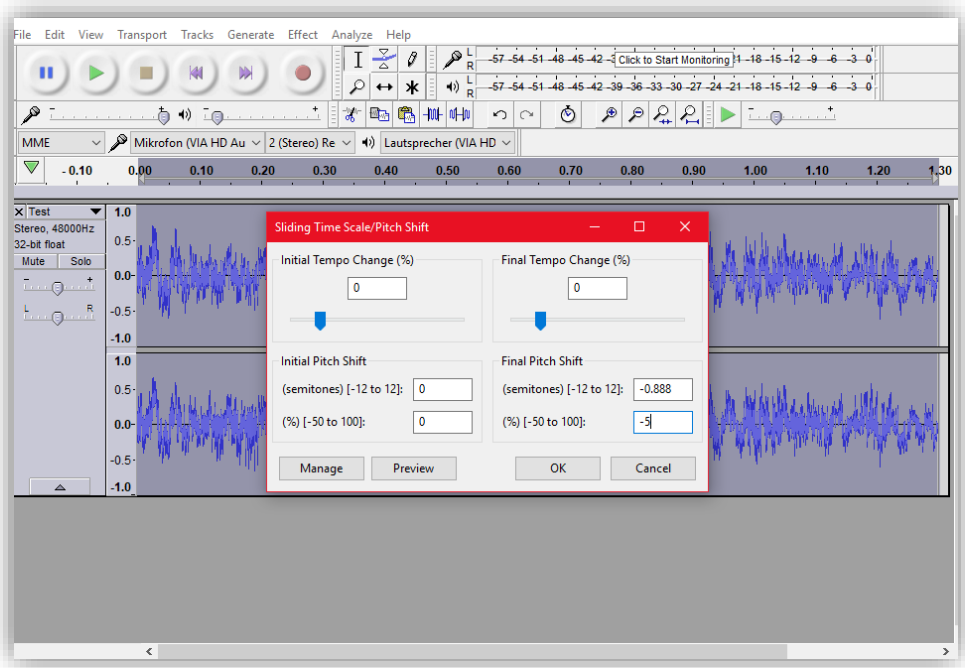
Loops are sounds that are being played repeatedly over and over again. Loops can be engine, bogie, horn sounds and many more. You can choose what you want to loop in the script. But we'll get there. I am going to show you a step by step guide on how to make a good loop. For this example, I will take a recording of the acceleration of an electric train. What we want to achieve is a loop that has no artefacts and is not pitched up or down during the loop so that we can pitch it up or down dynamically later in the sim. The Recording itself has pitched sounds obviously since it was accelerating. I am using Sony Vegas Pro and Audacity for this. Using just Audacity works just as well. Audacity, a free program, is required though.



First Step is to select an area of the recording you want to loop. Don't fade the loops.

Second. After rendering take the file you exported and drag and drop it to Audacity.

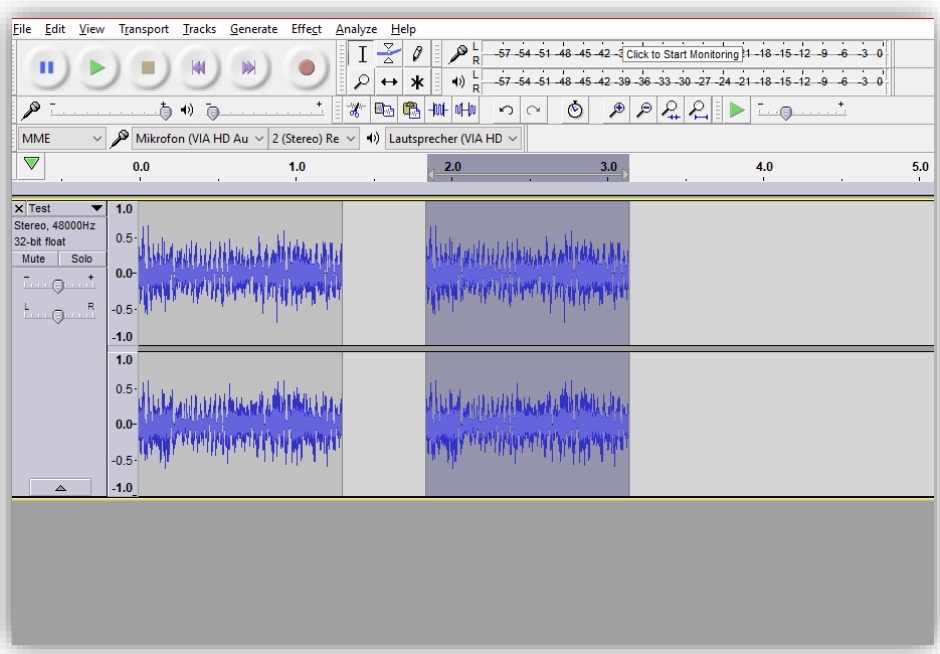
Third. Go to Effect>Sliding Time Scale/Pitch Shift. Now a window pops up. This is the step we are going to use to eradicate the pitch from the recording.



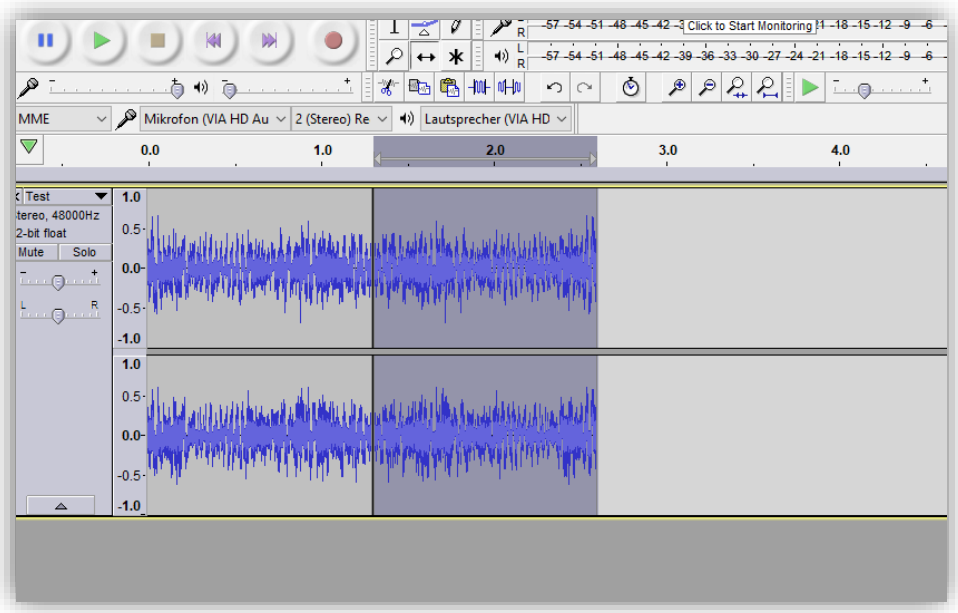
If you don't have a recording that pitches you can of course skip this step or just try to make a loop right in Sony Vegas/Audacity. Maybe you don't need these other steps.

Fourth. You now need to put a number under (%) [-50 to 100]. This will gradually pitch the recording up or down depending if your value is negative or positive. This will vary from recording to recording. You can preview it by clicking Preview (No Joke). When you are satisfied and the pitch doesn't seem to change, click OK.

Fifth Step. After doing so select the entire Audio sample, copy it, select some different time on the time line and paste it so that you now have two of the same samples.



Sixth. Your pasted sound sample should still be selected. If not select it and go to Effect>Reverse. Click F5 and move the selected same to the left so that it joins up with the previous one.



Seventh. You should now have a created a nice loop. If so, you can export the file and go to chapter 3. If not, there are three ways of fixing that.

First Way. This

Second Way. Go to Effect>Paulstretch. This will stretch your audio. This will only sometimes work and more often will make the audio sound artificial and not very good but it's worth a try.

Third Way. Export the audio file and put it back into Sony Vegas. Try to work with the file a little. Split the sound sample in half put the end in front and the start in the back. The file should overlap a little in the middle.

3 The proxyxml File

In this chapter, we will go through each section of the proxyxml file and I will tell you exactly what the mean and what they do

The proxyxml File is made up of a few sections that all interfere with each other. These sections are

The <Audio> section

This section tells the sim what sound file to use and how this sound file should be played. For example, at what volume it should be played.

The <Curve> section

This section is for the volume and pitch curves. These tell the sim at what e.g. speeds a sound will a have a specific volume or pitch.

The <Modifier Chain> section

This section tells the sim what controller value (e.g. RPM, Speed, etc.) the Curves should be played by and how they interact with each other.

The <InstanceGroup> section

This section tells the sim when to stop a specific Oneshot from being played. Useful for example with Joint sounds.

The <Loop> and <Oneshot> section

This is where it all comes together. Here you will tell the sim what sound and what Modifier Chain to use.

We are now going to go into detail with these sections:

<Audio>

As mentioned, the audio section tells the sim what sound to use and some other basics. Let's look at an example:

```
<kLoud-cSingleSampleSound d:id="1500">
    <Name d:type="cDeltaString">Motorstart</Name>
    <IsLooped d:type="bool">0</IsLooped>
    <Priority d:type="sUInt32">50</Priority>
    <BaseVolume d:type="sFloat32">0.8</BaseVolume>
    <VolumeVariation d:type="sFloat32">0</VolumeVariation>
    <BasePitchShift d:type="sFloat32">1</BasePitchShift>
    <PitchShiftVariation d:type="sFloat32">0</PitchShiftVariation>
    <AttenuationStartDist d:type="sFloat32">50</AttenuationStartDist>
    <NoFutherAttenuationDist d:type="sFloat32">300</NoFutherAttenuationDist>
    <InstanceGroup d:type="ref">0</InstanceGroup>
    <Sample>
        <kLoud-cSampleID>
            <Filename d:type="cDeltaString">RSSLO\VT642\RailVehicles\Sound\Inside\Sounds\Motorstart.wav</Filename>
        </kLoud-cSampleID>
    </Sample>
</kLoud-cSingleSampleSound>
```

<kLoud-cSingleSampleSound d:id=…..>
Give the Sound file a specific number to use later on in the <Loop> and <Oneshot> section. There's also a **<kLoud-cRandomSampleSound d:id=…..>** Tag. That means you can use several different audio files that will be played randomly. I will tell you how that works later.

<Name>
You can assign it any name you want. This has no effect on anything and is just going to help you keep an overview

<Is Looped>
Value can be 0 or 1. 0 means it will not Loop. 1 means it will.

<Priority>
Value in Percent which tells the sim the priority of the sound. I have never seen this have an effect on anything but I read somewhere that with it helps prevent bugs with old soundcards.

<Base Volume>
Value can be between 0 and 1. 0 means the Sound is muted. 1 means the sound will be played at full volume.

<VolumeVariation>
Value can be between 0 and 1. 0 means the sound is not varied in volume each time it is played. 1 means that the volume variation each time the sound is played is biggest and the volume in the sim can be anything between 0 and 1.

<BasePitchShift>
Value can be between 0 and infinity. This will tell the sim at what pitch the file shall be played. 1 is normal pitched. 1.4 would mean it will be played a bit higher. 0.7 would mean a bit lower.

<PitchShiftVariation>
Value can be between 0 and infinity. Same with the Volume Variation. If this Value is anything but 0 the sound will be pitched differently each time it is played.

<AttenuationStartDist>
Value in meters that tells the sim at what distance to start making the file quieter. I have not heard this in effect yet.

<NoFurtherAttenuationDist>
Value in meters that tells the sim at what distance to stop making the file quieter.

<InstanceGropup>
Leave this at 0 and set the Instance Group in the <Oneshot> section

<Filename>
Put in here the path for your sound file. The sim will look inside the Assets Folder. You can start there.

<kLoud-cRandomSampleSound d:id=…..>

Let's look at another example for this. You can use this Tag to play a random sound file from a pair of different files.

```
<kLoud-cRandomSampleSound d:id="3400">
    <Name d:type="cDeltaString">Bremsquitsch_30</Name>
    <IsLooped d:type="bool">0</IsLooped>
    <Priority d:type="sUInt32">50</Priority>
    <BaseVolume d:type="sFloat32">1</BaseVolume>
    <VolumeVariation d:type="sFloat32">0</VolumeVariation>
    <BasePitchShift d:type="sFloat32">1</BasePitchShift>
    <PitchShiftVariation d:type="sFloat32">0</PitchShiftVariation>
    <AttenuationStartDist d:type="sFloat32">50</AttenuationStartDist>
    <NoFutherAttenuationDist d:type="sFloat32">300</NoFutherAttenuationDist>
    <InstanceGroup d:type="ref">0</InstanceGroup>
    <Sample>
        <kLoud-cSampleWithLikelihood d:id="3415">
            <Likelihood d:type="sFloat32">0.5</Likelihood>
            <Sample>
                <kLoud-cSampleID>
                    <Filename d:type="cDeltaString">RSSLO\VT642\RailVehicles\Sound\Inside\Sounds\Bremsquitsch_30_1.wav</Filename>
                </kLoud-cSampleID>
            </Sample>
        </kLoud-cSampleWithLikelihood>
        <kLoud-cSampleWithLikelihood d:id="3416">
            <Likelihood d:type="sFloat32">0.5</Likelihood>
            <Sample>
                <kLoud-cSampleID>
                    <Filename d:type="cDeltaString">RSSLO\VT642\RailVehicles\Sound\Inside\Sounds\Bremsquitsch_30_2.wav</Filename>
                </kLoud-cSampleID>
            </Sample>
        </kLoud-cSampleWithLikelihood>
    </Sample>
</kLoud-cRandomSampleSound>
```

You can see that most of it is the same, except for the part with the filenames.

<kLoud-cSampleWithLikelihood d:id=…..>

Put here a random number. This number doesn't matter but I do suggest making it close to your *RandomSampleSound* Number as to prevent confusion.

<Likelihood>

If this value is for example at 0.5 the sound will have an equal chance to be played at random. If this value is for example 0.25 it has less chance to be played. These values don't need to add up to 1 as they are relative within their specific Random Sample Sound.[1]

<Filename>

Put in here the path for your sound file. The sim will look inside the Assets Folder. You can start there.

[1] http://railworkswiki.com/tiki-index.php?page=Sound&structure=Reference%20Manual

<Curve>

The Curve sections holds the information for the Volume and Pitch relative to a specific controller value. That controller value is specified in the <ModifierChain> section and can be absolutely anything (e.g. speed, RPM, ListenerDistanceSquared, Horn, etc.)

Let's also look at an example for this:

```
<kLoud-cCurve d:id="1403">
    <Name d:type="cDeltaString">Motor_Aus</Name>
    <CurvePoint>
        <kLoud-cRVector2 d:id="1407">
            <X d:type="sFloat32">1</X>
            <Y d:type="sFloat32">1</Y>
        </kLoud-cRVector2>
        <kLoud-cRVector2 d:id="1408">
            <X d:type="sFloat32">500</X>
            <Y d:type="sFloat32">0</Y>
        </kLoud-cRVector2>
    </CurvePoint>
</kLoud-cCurve>
<kLoud-cCurve d:id="2453">
    <Name d:type="cDeltaString">Rollen_60</Name>
    <CurvePoint>
        <kLoud-cRVector2 d:id="2454">
            <X d:type="sFloat32">2</X>
            <Y d:type="sFloat32">0</Y>
        </kLoud-cRVector2>
        <kLoud-cRVector2 d:id="2455">
            <X d:type="sFloat32">17</X>
            <Y d:type="sFloat32">1</Y>
        </kLoud-cRVector2>
        <kLoud-cRVector2 d:id="2455">
            <X d:type="sFloat32">30</X>
            <Y d:type="sFloat32">0</Y>
        </kLoud-cRVector2>
    </CurvePoint>
</kLoud-cCurve>
```

<kLoud-cCurve d:id=…..>

This is the number for your curve. Give it a specific one. We need it later in the <ModifierChain> section.

<Name>

You can assign it any name you want. This has no effect on anything and is just going to help you keep an overview.

<kLoud-cVector d:id=…..>

Put here a random number. This number doesn't matter but I do suggest making it close to your *kLoud-cCurve* Number as to prevent confusion.

As you can see we always have an X and a Y value. It's a system of coordinates. Finally, your math lessons paid off.
On the X axis, you can always see the controller value that you want your curve to be. *Motor_Aus* for example hast the RPM on the X axis and *Rollen_60* has the speed.
The Y axis will always give you the Pitch or volume. The Pitch can be anything you want it to be. 1 is always going to be the normal pitch. Volume can only be between 0 and 1. 1 being the Loudest. You can set your curve as a pitch or volume curve in the Modifier Chains further down.

Here’s another visualization from ChrisTrains Website:

The four points are (X=0, Y=0), (X=11.11, Y=1), (X=16.66, Y=1), (X=22.22, Y=0.8). In the

Screenshot from: http://www.christrains.com/ts_faq_sounds.html

That about covers the <Curve> section.

<ModifierChain>

The ModifierChain is putting the information from the Curves into context. Here is an example:

```
<kLoud-cModifierChain d:id="4202">
    <Name d:type="cDeltaString">2_1</Name>
    <Modifier>
        <kLoud-cVolumeCurveModifier d:id="1">
            <ControllerIsGlobal d:type="bool">0</ControllerIsGlobal>
            <ControllerName d:type="cDeltaString">ListenerDistanceSquared</ControllerName>
            <Curve d:type="ref">645302987</Curve>
        </kLoud-cVolumeCurveModifier>
        <kLoud-cPitchCurveModifier d:id="4206">
            <ControllerIsGlobal d:type="bool">0</ControllerIsGlobal>
            <ControllerName d:type="cDeltaString">VirtualThrottle</ControllerName>
            <Curve d:type="ref">234</Curve>
        </kLoud-cPitchCurveModifier>
        <kLoud-cVolumeCurveModifier d:id="4205">
            <ControllerIsGlobal d:type="bool">0</ControllerIsGlobal>
            <ControllerName d:type="cDeltaString">AbsoluteSpeed</ControllerName>
            <Curve d:type="ref">4203</Curve>
        </kLoud-cVolumeCurveModifier>
        <kLoud-cPitchCurveModifier d:id="4215">
            <ControllerIsGlobal d:type="bool">0</ControllerIsGlobal>
            <ControllerName d:type="cDeltaString">AbsoluteSpeed</ControllerName>
            <Curve d:type="ref">4213</Curve>
        </kLoud-cPitchCurveModifier>
    </Modifier>
</kLoud-cModifierChain>
<kLoud-cModifierChain d:id="4312">
    <Name d:type="cDeltaString">RetarderLow Brake</Name>
    <Modifier>
        <kLoud-cVolumeCurveModifier d:id="4316">
            <ControllerIsGlobal d:type="bool">0</ControllerIsGlobal>
            <ControllerName d:type="cDeltaString">VirtualThrottle</ControllerName>
            <Curve d:type="ref">345</Curve>
        </kLoud-cVolumeCurveModifier>
        <kLoud-cVolumeCurveModifier d:id="4315">
            <ControllerIsGlobal d:type="bool">0</ControllerIsGlobal>
            <ControllerName d:type="cDeltaString">AbsoluteSpeed</ControllerName>
            <Curve d:type="ref">4333</Curve>
        </kLoud-cVolumeCurveModifier>
        <kLoud-cPitchCurveModifier d:id="4325">
            <ControllerIsGlobal d:type="bool">0</ControllerIsGlobal>
            <ControllerName d:type="cDeltaString">AbsoluteSpeed</ControllerName>
            <Curve d:type="ref">4323</Curve>
        </kLoud-cPitchCurveModifier>
    </Modifier>
</kLoud-cModifierChain>
```

<kLoud-cModifierChain d:id=…..>

This is the number for your ModifierChain. Give it a specific one. We need it later in the <Loops> and <Oneshot> section.

<Name>

You can assign it any name you want. This has no effect on anything and is just going to help you keep an overview.

<Modifier>

This is the section where you are defining your curves. Here is where you tell the sim which volume or pitch curve is dependent on which ControllerName.

<ControllerIsGlobal>

No Idea. Doesn’t seem to change anything.

<Controller Name>
A ControllerName is a parameter which defines when a sound is played. I found this quite hard to explain as it can be so many things like speed, a switch, the motor starting or a change in weather as long as these things are implemented by the developer. Further down you will find a guide that will tell you where to find these controller names. I myself fully understood what a Controller Name is once I went into the sim and looked at what the Controller Values, when You press a button or change the speed. A guide on how to find these is further down.

<Curve>
This is the actual number of the curve. You should have set this number under *<kLoud-cCurve d:id=…..>* in the <Curve> section.

<InstanceGroup>
With the InstanceGroup you can tell the sim to stop certain sounds from being triggered too many times. I use this mainly for Junction Sounds

```
    </kLoud-cModifierChain>
</ModifierChain>
<InstanceGroup>
    <kLoud-cInstanceGroup d:id="55555">
        <Name d:type="cDeltaString">Joint</Name>
        <MaximumNumberOfInstances d:type="sUInt32">1</MaximumNumberOfInstances>
        <InstanceExceededAction d:type="cDeltaString">RejectNew</InstanceExceededAction>
    </kLoud-cInstanceGroup>
</InstanceGroup>
<Loop>
    <kLoud-cControlledLoop d:id="8962">
        <PlayState d:type="cDeltaString">OUTSIDE</PlayState>
```

<kLoud-cInstanceGroup d:id=…..>
This is the number for your InstanceGroup. Give it a specific one. We need it later in the <Loops> and <Oneshot> section.

<Name>
You can assign it any name you want. This has no effect on anything and is just going to help you keep an overview.

<MaximumNumberOfInstances>
Put in here the maximum number of sounds that are allowed to be triggered at the same time.

<InstanceExceedAction>
This is where you can specify by which behaviour the sim should stop or reject a certain sound to be played. There are three things you can enter here:

Instance Exceed Action	**Reject New** - No further sounds from this instance group will play until the current sound has finished or reached the end of the file; i.e. all new trigger requests are rejected. **Stop Oldest** – If a new trigger request is made then a new sound will play immediately and the oldest playing sound in the instance group will stop immediately. (Becomes more useful with maximum number of instances larger than 1). **Stop Newest** - If a new trigger request is made then a new sound will play immediately and the newest playing sound in the instance group will stop immediately. (Becomes more useful with maximum number of instances larger than 1).

[2]

[2] http://railworkswiki.com/tiki-index.php?page=Instance%20Group&structure=Reference%20Manual

<Loop> and <Oneshot>

This is where everything will come together. In these sections you will specify which sound to be played and which Modifier Chain influences that sound

<Loop>

As I've mentioned Loops are Sounds that play over and over again. They loop like the word suggests.

```
<kLoud-cControlledLoop d:id="2500">
    <PlayState d:type="cDeltaString">INSIDE</PlayState>
    <Name d:type="cDeltaString">Engine 1000 Inside</Name>
    <Sound d:type="ref">2000</Sound>
    <ModifierChain d:type="ref">2400</ModifierChain>
    <ControllerName d:type="cDeltaString">AbsoluteSpeed</ControllerName>
    <LoopCondition d:type="cDeltaString">ValueInRange</LoopCondition>
    <ActiveRangeStartValue d:type="sFloat32">0</ActiveRangeStartValue>
    <ActiveRangeEndValue d:type="sFloat32">2100</ActiveRangeEndValue>
    <ValueIsChangingGateTime d:type="sFloat32">0</ValueIsChangingGateTime>
    <InstanceGroup d:type="ref">0</InstanceGroup>
</kLoud-cControlledLoop>
```

<kLoud-cControlledLoop d:id=…..>

This is the number for your Loop. This doesn't necessarily have to be a specific one but watch out that this number hasn't already been used somewhere else. That causes problems and crashes.

<PlayState>

This can either be INSIDE , OUTSIDE or BOTH.

INSIDE: The sound will only play on the inside of the train.

OUTSIDE: The sound will only play on the outside of the train.

BOTH: The sound will be played on the outside of the train but will be hearable on the inside by attenuation. EFX must be enabled for this to work.

<Name>

You can assign it any name you want. This has no effect on anything and is just going to help you keep an overview.

<Sound>

Specify here which Sound file is supposed to be played. The Number is referenced in the <Sound> section under <kLoud-cSingleSampleSound d:id=…..> or <kLoud-cRandomSampleSound d:id=…..>

<ModifierChain>

Specify here which ModifierChain is supposed to influence the sound Loop. This should be referenced in the <ModifierChain> section under <kLoud-cModifierChain d:id=…..>. If this value is at 0, no Modifier Cahin will be used and the sound will be played by the following parameters.

<ControllerName>
The ControllerName the entire sound is being played by.

<LoopCondition>
There are a few possibilities here:

ValueInRange: Loop will play whenever the Controller Name is within the active range of start and end values.

ValueIsChangingUp: Loop will play whenever the Controller Name is changing upwards. Active range will be ignored

ValueIsChangingDown: Loop will play whenever the Controller Name is changing downwards. Active range will be ignored

ValueIsChanging: Loop will play whenever the Controller Name is changing upwards or downwards. Active range will be ignored

<ActiveRangeStartValue>
Start Value of the Controller Value when the loop will be played.

<ActiveRangeEndValue>
End Value of the Controller Value when the loop will stop being played.

<ValueIsChangingGateTime>
It is vital to use this setting when triggering a sound from a value which often fluctuates or changes unpredictably (i.e. some simulation parameters). If this is set to '10' then no matter how much the controller value changes and is valid to trigger a sound, nothing will happen until 10 seconds has passed from the last trigger. This can be used to stop loops juddering on and off when based on a cab lever moving or can be used to stop hundreds of Oneshots being triggered when triggering from a simulation controller value that changes every frame.[3]

<InstanceGroup>
Specify here which InstanceGroup is supposed to influence the loop. When Value is 0 it will be ignored.

<Oneshot>

Oneshots only play once in the sim. They don't loop.

```
<kLoud-cTriggeredOneshot d:id="57690184">
    <PlayState d:type="cDeltaString">BOTH</PlayState>
    <Name d:type="cDeltaString">PAX Door Close</Name>
    <Sound d:type="ref">58330920</Sound>
    <ModifierChain d:type="ref">0</ModifierChain>
    <ControllerName d:type="cDeltaString">DoorsOpenCloseRight</ControllerName>
    <TriggerType d:type="cDeltaString">ValueDecreasePast</TriggerType>
    <TriggerValue d:type="sFloat32">0.990000</TriggerValue>
    <ValueIsChangingGateTime d:type="sFloat32">0.000000</ValueIsChangingGateTime>
    <InstanceGroup d:type="ref">0</InstanceGroup>
</kLoud-cTriggeredOneshot>
```

[3] http://railworkswiki.com/tiki-index.php?page=Loops%20and%20Oneshots&structure=Reference%20Manual – Slightly edited in places

<kLoud-cTriggeredOneshot d:id=…..>
This is the number for your Oneshot. This doesn't necessarily have to be a specific one but watch out that this number hasn't already been used somewhere else. That causes problems and crashes.

<PlayState>
This can either be INSIDE , OUTSIDE or BOTH.

INSIDE: The sound will only play on the inside of the train.

OUTSIDE: The sound will only play on the outside of the train.

BOTH: The sound will be played on the outside of the train but will be hearable on the inside by attenuation. EFX must be enabled for this to work.

<Name>
You can assign it any name you want. This has no effect on anything and is just going to help you keep an overview.

<Sound>
Specify here which Sound file is supposed to be played. The Number is referenced in the <Sound> section under <kLoud-cSingleSampleSound d:id=…..> or <kLoud-cRandomSampleSound d:id=…..>

<ModifierChain>
Specify here which ModifierChain is supposed to influence the sound Oneshot. This should be referenced in the <ModifierChain> section under <kLoud-cModifierChain d:id=…..>. If this value is at 0, no Modifier Cahin will be used and the sound will be played by the following parameters.

<ControllerName>
The ControllerName the entire sound is being played by.

<TriggerType>
There are a few possibilities here:

ValueIncreasePast: Oneshot will play whenever the Controller Name increases past the Trigger Value

ValueDecreasePast: Oneshot will play whenever the Controller Name decreases past the Trigger Value

ValueMovesPast: Oneshot will play whenever the Controller Name decreases or increases the Trigger Value

ValueIsChangingUp: Oneshot will play whenever the Controller Name changes upwards by the specified Trigger Value (If set to e.g. 0.5 the sound will always be played when the Controller moves up by 0.5)

ValueIsChangingDown: Oneshot will play whenever the Controller Name changes downwards by the Trigger Value

ValueIsChanging: Oneshot will play whenever the Controller Name changes upwards or downwards by the Trigger Value

EveryTriggerValueIncreasing: Basically, the same as Value is Changing Up

EveryTriggerValueDecreasing: Basically, the same as Value is Changing Down

EveryTriggerValueMovesPast: Basically, the same as Value is Changing

<TriggerValue>

Explained above

<ValueIsChangingGateTime>
It is vital to use this setting when triggering a sound from a value which often fluctuates or changes unpredictably (i.e. some simulation parameters). If this is set to '10' then no matter how much the controller value changes and is valid to trigger a sound, nothing will happen until 10 seconds has passed from the last trigger. This can be used to stop loops juddering on and off when based on a cab lever moving or can be used to stop hundreds of Oneshots being triggered when triggering from a simulation controller value that changes every frame.[4]

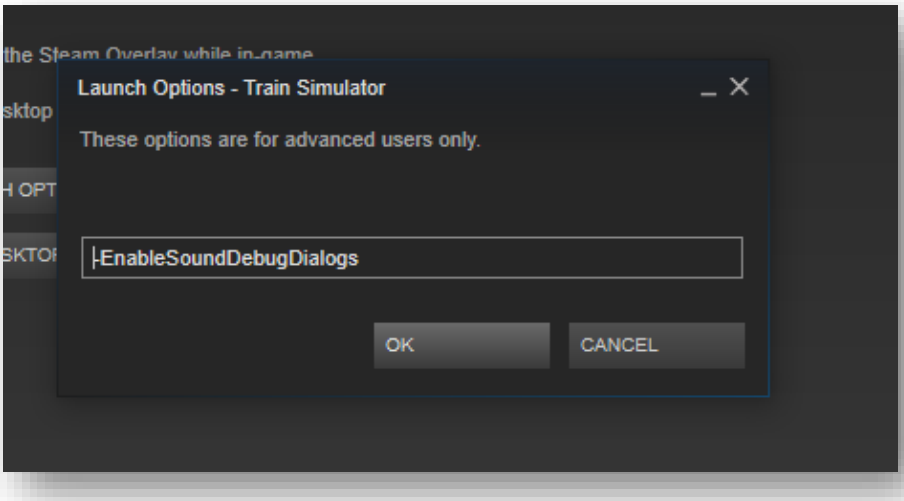
<InstanceGroup>
Specify here which InstanceGroup is supposed to influence the loop. When Value is 0 it will be ignored.

[4] http://railworkswiki.com/tiki-index.php?page=Loops%20and%20Oneshots&structure=Reference%20Manual – Slightly edited in places

How to find your Controller Name

As mentioned above each developer can and will set his own Controller Names in the engine script of the train. There are some basic Controller Names (e.g. AbsoluteSpeed for the speed in m/s) you should be able to find in all kinds of trains but especially developers which do more advanced stuff with their trains (e.g. vR, AP, RWA, etc.) make their own custom Names to be able to do what they actually do. So how do you find these controller Names? I will tell you here.

1) Go to steam, right click on Train Simulator, click Properties then on “SET LAUNCH OPTIONS…”
2) Put in “-EnableSoundDebugDialogs” with the dash.

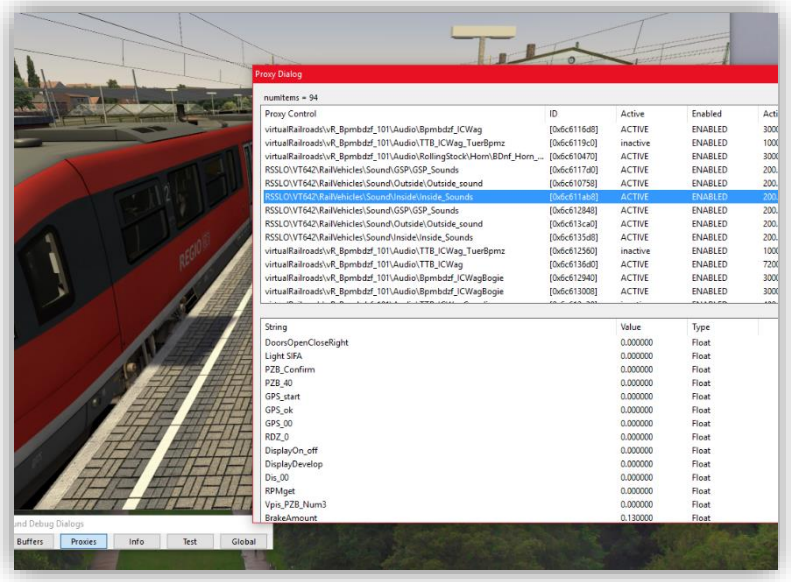


3) Start Train Simulator go to Settings>Graphics and under “Full Screen” select the windowed mode

4) Click save and restart the simulator

5) Once back in the menu a small window should have appeared in the bottom left.

6) Now start a scenario with the train you'd like to find the controller names of.
7) Click on "Proxies". Now you can select the actual proxyxml files and see ALL the controller names plus their values in real-time inside the sim.



Pretty neat and helpful.

Here are some basic ControllerNames you should find in every train to get you started.

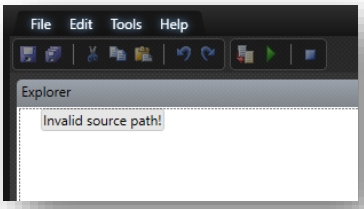
- AbsoluteSpeed: Gives you the speed in meters per second.
- DistanceTravelled: Gives you the Distance you have travelled with you train meters. Resets every time you change trains or restart the scenario.
- Joints or JointCount: This Value will add up in steps of 1 each time a Joint in the Track is being driven over.
- VirtualThrottle: Value from (mostly) 0 to 1. Will give you the position of the throttle.
- BrakeAmount: Gives you the brake amount where 0 is no brake and 1 is full brake.
- Ammeter (only on electric or diesel-electric trains): Will give you the current drawn from overhead lines in Ampere. This value is sometimes unrealistically high due to wrong implementation or a workaround for a particular problem on the developer side.
- RPM (only on diesel-hydraulic trains): Gives you the RPM Value in RPM.

4 How to use the blueprint editor (for starting a sound from scratch)

As mentioned above the Blueprint Editor is really helpful when starting a sound from scratch. It is also really easy to use.

1) Start by opening it. Go to you railworks main folder (steam/steamapps/common/railworks) and look for the *BlueprintEditor2.exe*

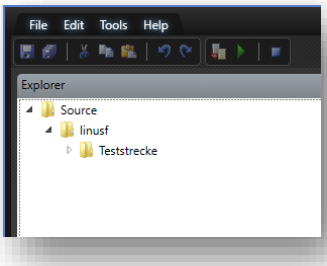
2) Open it up. If you get an error message like this



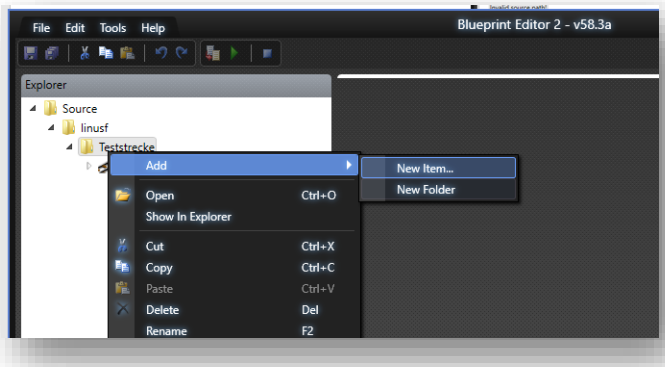
simply go back to the railworks main directory and create a folder called Source and restart the program

3) After restarting you should have a folder called Source up in the top. Right click on it and create another folder. The Folder Name will be the Name under which the folder will pop up in the Assets Folder. Next create another folder and call it what you want. Preferably Audio or RailNetwork or something like that. Depends on your use later.

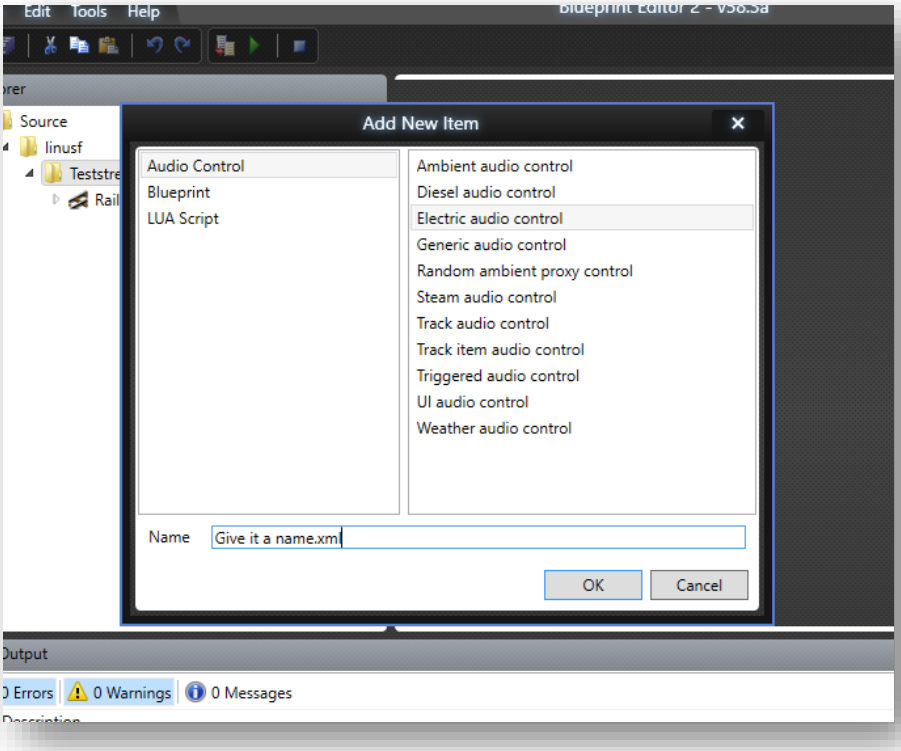
4) Everything should now look like this (except for the names)



5) In the last subfolder you should now be able to right click and create a new Blueprint by going to Add and then clicking New Item



6) A window will pop up with a bunch of different blueprints to create. For trains we will use either the Electric/Diesel or Steam audio control. I am going to create an electric blueprint. You can give it a name in the text box at the bottom.



7) Now you will see a bunch of sort of dropdown menus which should already be quite familiar to you. In each of these sections you can click the little arrow to add a new Element.

It's best if you now click and try a few things. Chapter 3 shall help you with the rest. When you finished implementing your sound you can got to File>Export *Name*.xml>Export with references. If you now go to the Assets you will find the audio .xml plus your .wav sound files converted into .dav and the sounds ready to use.

5 Some useful Tips and Links

Here are some useful tips that should help you.

1) Don't try anything too complicated. I see a lot of Developers do this. Trying to put together a lot of different Sound which in the end it just don't sound good. For example, having more than one or two rolling sounds playing at the same time for the same speed. Just use one sound for a few different speeds and fade them into each other while braking or accelerating. It'll do wonders. For example, I use rolling sounds for about 60 km/h, 120 km/h and 160 km/h (and 200 or 300 if the train can go faster).
2) Train Simulator isn't good with sound. That means that your sound files don't have to be perfect too.

Some Links:

- https://www.rail-sim.de/wiki/index.php/Startparameter (German, gives you all the different launch options)
- https://www.christrains.com/ts_faq_sounds.html (ChrisTrains gives some introduction on how audio is processed in Train Simulator)
- http://railworkswiki.com/tiki-index.php?page_ref_id=499 (Probably the most important website as it contains even more explanations on audio blueprints)

6 Epilogue

Alright. That's it. I should have explained to you the tools given by the simulator. Now is your turn. Soundmodding requires some creativity and critical thinking. I wish you the best of luck and lots of fun.

COPYRIGHT:

There is no Copyright on this work. You are free to translate, reupload, transform and build upon this work but you must always credit the original author Linus Follert. More information here.