

Introducing



New Digital Audio Workstation inspired multi-track music and event editing interface with hardware control surface support

FMOD Studio V1.08.08

Introduction

This document covers FMOD Studio, a software tool that allows sound designers to create interactive audio content for use in games, simulators, and other applications.

FMOD Studio is designed to be used in conjunction with the FMOD Studio Programmer's API. Using the Programmer's API an audio programmer can implement content made in FMOD Studio into a software project and control its playback through code.

A Note for FMOD Designer Users

If you've used FMOD Studio's predecessor, FMOD Designer, you may notice that FMOD Studio uses many of the same concepts. It is, however, an entirely new product with new features, workflows, and capabilities, and so many common Designer features and tasks need to be approached differently. Here's a short list of major changes to look out for:

- The Interactive Music System has been merged with the Multi Track Editor to create a new 'Event Editor' with the best features of both.
- Sound Defs no longer exist. Instead, each sound instance (or 'sound module') has its own playlist and settings. You can re-use a sound module in multiple locations by using Event Reference Sound Modules.
- Reverb Defs are now handled through the new 'snapshot' system.
- Parameter velocity is now exclusive to the 'Timeline' Parameter automatically included in every event.
- The concept of 'Workspace' no longer exists, and it is no longer necessary to split a project up into multiple project files to keep it manageable. Studio projects consist of a folder containing multiple automatically-managed files, and revision control integration allows multiple sound designers to work on a single project simultaneously without interfering with each other's work.
- By default, any audio file you add to a project is automatically copied into a subdirectory of the project, instead of being externally referenced.
- It is no longer necessary to assign individual audio files to banks. Instead, assigning an event to a bank assigns all its contents to that bank automatically.

Important

FMOD Studio differs from many traditional audio content creation tools in that it allows only non-destructive editing. This means that no matter what changes you make to a sound in FMOD Studio, the sound file used to create the sound is never directly altered in any way. Instead, all processes that change or manipulate audio output are performed in real time.

Under normal circumstances, importing a sound file into an FMOD Studio project automatically copies that file into the project's audio asset subdirectory. Studio then uses this copy of the file for all purposes. If you need to alter a sound file used in your project, you need to edit the version in the audio bin folder.

Before You Start

FMOD Studio is continually being improved, and updates containing new features are released frequently. Unfortunately, updating this manual to reflect these changes takes time, and so there is typically a small delay between a new feature being released and that feature being included in this manual.

In the event that you cannot find exactly what you are looking for in this manual, we recommend you visit the [FMOD Support Page](#) for help and documentation.

Table of Contents

Introduction.....	1
A Note for FMOD Designer Users.....	1
Core Concepts	17
The Sound File is not the Sound	17
Implementation and Creation: The blurry line	17
FMOD Versions.....	18
Minor Releases.....	18
Major Releases.....	18
Platform Support	18
Key Controls.....	19
Context-sensitive Menus	19
Tooltips	19
Manual Value Input.....	19
The Menu Bar	20
File Menu	20
The Edit Menu.....	21
<i>Set Color</i>	23
The View Menu	23
The Window Menu	24
The Help Menu	26
Common Control Types.....	26
Dials	26
Sliders	27
Buttons	28
Manual Value Input.....	29
The Preferences Window.....	29
Audio.....	30
Metering channel order	30
Assets	30
Assets folder (optional)	30
External Sample Editor (optional)	30
Plug-ins folder (optional)	30
Build	30
Built banks output directory (optional)	30
Control Surfaces.....	30
Source Control.....	31

The Event Editor Window.....	31
The Events Browser	31
The Banks Browser	32
The Assets Browser	32
Context Menu.....	32
The Editor	33
Breadcrumbs.....	34
Playheads and Cursors.....	34
Missing References	34
Transport Bar	34
Birdseye View	36
The Deck	36
3D Preview	36
Properties	37
The Event Browser Window.....	37
The Events Browser	37
The Tags Browser	37
The Mixer Window	38
Mixing Desk	38
Mixer View	39
The Mixer Browser	39
Snapshots Tab	39
Selected Buses Tab	39
The Deck	39
The Audio Bin Window.....	40
The Assets Browser	40
FMOD.io	40
Tracks View and Strips View.....	40
Tracks View	40
Strips View.....	42
Navigating the Strips View.....	43
Getting Started: Saving the Project.....	44
Project Files	44
Saving a Project.....	45
Basic Control Concepts.....	45
Multiple Windows.....	45
Editor Zoom	46
Resizing and Repositioning Modules	46
Event Drag and Drop.....	46
Common Properties	46

Volume.....	47
Components with Volume Controls	47
Pitch.....	47
Components with Pitch Controls.....	48
Project Element Data.....	48
Command Line Interface.....	49
Modulators.....	49
Creating a Modulator	50
Accessing the Pitch or Volume Controls.....	50
Creating a New Modulator	50
Random Modulation	50
Random Modulation Example.....	51
AHDSR Modulation	51
Autopitch.....	52
Creating an Autopitch Modulator	53
Accessing or creating a Game Parameter.....	53
Accessing or Creating a Sound Module	54
Side Chain Modulator.....	55
Creating a Sidechain Modulator	56
Sidechain Properties.....	56
Automation	58
Creating Automation	58
Accessing an Event	59
Creating Automation	59
Automation Track Names	60
The Assets Browser	61
Adding Sound Files to the Audio Bin.....	61
Method 1	61
Method 2	61
Audition Mode.....	62
The Assets Browser Context-sensitive Menu.....	62
New Folder.....	62
Move into New Folder	62
Move to Top Level	62
Move to	62
Refresh	62
Delete.....	62
Open in Explorer/Open in Finder	62
Open in External Sample Editor	62
Unused Sound Files	63

Creating Your First Event.....	63
Introduction	63
Creating a new Event.....	63
Naming the New Event	63
Adding Sounds to an Event.....	64
Option 1	64
Option 2	64
Modules.....	66
Sound Modules	66
Modules that Produce Sound	66
Creating a Sound Module	67
Multi Sound Module	67
Playlist Properties	68
Playlist Context Menu	68
Plug-in sound modules	69
Bulk Editing	70
Editing Properties.....	71
Range Extremes	72
Controlling Property	72
Bulk Editing Parameters	72
Bulk Editing the 3D Previewer	73
Module Playback Behavior.....	73
Module Looping Behavior.....	74
Looping a Module's individual Playlist Entries	74
Module set to single playback, all entries set to loop	74
Sound Module set to single playback, some entries set to loop.....	74
Sound Module set to looping playback, all entries set to not loop.....	74
Sound Module set to loop, some entries set to loop.....	75
Sound Module set to loop with all playlist entries set to loop	75
Single vs. Multi Sound Modules.....	75
Event Module	75
Event Reference Modules	75
Creating Event Reference Modules	76
Copying an Event Module.....	77
Scatterer Sound Modules.....	77
Trigger Behavior	77
A word on Generative Audio	79
Scatterer Sound Modules in 3D Events	79
Audio Weather Module.....	79
Single Sound Module	79

Events within Events	80
Parameters in Nested Events.....	80
Event Instances.....	81
Trigger Regions.....	81
Trigger Region Names.....	81
Renaming Trigger Regions	82
Trigger Region Colors.....	82
Trigger Region Instances.....	83
Precise positioning for Trigger Regions.....	83
Move To.....	83
Move to Cursor	83
Bring to Front	83
Send to Back.....	84
The Logic Track	84
Loop Regions	84
Creating a Loop Region on the Logic Track	85
Creating a Loop Region on a Trigger Region.....	85
Sustain Points	85
Creating a Sustain Point	86
Working with Sustain Points	87
Timelocked and Non-Timelocked Sounds	88
Timelocked Sound Modules	88
Non-Timelocked Sound Modules	88
Loop Region Logic.....	89
Tempo Markers.....	89
Markers	90
Creating a Marker	91
Transition Timelines	91
Programmer Sound Modules.....	91
Multiple Projects.....	91
FMOD Studio Music System.....	92
The Logic Track.....	92
Markers	92
Creating Markers	93
Working with Markers	93
Tempo Markers.....	93
Creating Tempo Markers	94
Defining Tempo.....	94
Working with Tempo Markers	95
Transitions	95

Creating Transitions	95
Working with Transitions	96
Transition Functionality	96
Transition Regions	97
Creating Transition Regions	97
Working with Transition Regions	98
Transition Region Quantization	98
Transition Timelines	98
Loop Regions and Sustain Points	100
Trigger Behavior	101
Defining Trigger Behavior	101
Time	101
Tempo	102
Seek	103
Polyphony	103
Parameters	103
How Parameters Can Affect Events	104
Parameter Properties	104
Value	105
Smoothing	105
Parameter Modulation	105
Distance Parameter	106
Direction Parameter	106
Elevation Parameter	107
Event Cone Angle Parameter	107
Event Orientation Parameter	107
The Timeline Parameter	108
Aspects of Time-based Events	108
Timelocked sounds	108
Logic Markers and Time	109
The Distance Parameter	109
Step 1 Creating a Distance Parameter	110
The Event Cone Angle Parameter	110
Step 1 Creating an Event Cone Angle Parameter	111
Controlling Events External to an Event	112
Automating Parameters of Nested Events	112
Linking Parameters of Nested Events	115
Tags	115
Searching for Tags	116
The Tag Browser	116

Creating and Assigning Tags	117
Creating a new Tag.....	117
Method 1: Adding Tag information to an Event	117
Method 2: Assigning an Event to a Tag.....	117
Creating a Tag in the Tag Browser.....	118
The Browser Search Bar	119
GUID Searches.....	119
Tag Searches.....	119
Routing and Assigning.....	120
The Mixer	121
In Depth: The Mixer Functionality.....	121
The Mixer Tool Set	122
Multiple Mixer view.....	122
Group Solo.....	122
Basic Signal Paths.....	122
A Word on Effects	124
Group Buses.....	125
Why Group Buses?.....	125
Keep Things Simple.....	125
Creating a Group Bus	126
Step 1 Display the Mixer View	126
Step 2 Create a Group Bus	126
Routing Events into Group Buses	126
Step 1 Open a Project with existing Events and Group Buses	127
Step 2 Route an Event into an existing Group Bus	127
Step 3 Rerouting an event to a New Group Bus	128
Step 3 Route an event to the Master Bus.....	128
Signal Routing	129
Working with Group Buses	129
A new Channel Strip	129
The Deck.....	130
Buses within Buses.....	130
Creating Group Buses.....	131
Step 1 Open a Project with existing Events and Group Buses	131
Step 2 Create a Group Bus within an existing bus	131
Sends	132
Basic Send routing in Events.....	133
Basic Send routing in Mixer	133
Adding Sends to Events	135
Step 1 Creating a Send.....	135

Send/Return Functionality	135
Routing Audio Tracks to Return Tracks	136
Step 1 Creating a Send.....	136
Adding Sends in the Mixer	137
Adding a Send	137
Step 1 Creating a Send.....	137
Routing Buses to existing Return Buses.....	138
Step 1 Creating a Send.....	138
Routing at the Event Level	139
Routing to tracks.....	139
Routing to Audio Tracks.....	139
Step 1 Create an new Event.....	139
Step 2 Assigning Audio Tracks to Audio Tracks.....	140
Signal Routing within an Event	140
VCAs	141
Implementation	141
VCA Controls	141
Creating VCAs.....	142
Step 1 Create a new VCA.....	142
Assigning to VCAs.....	142
Step 1 Assigning to VCAs.....	143
Multiple Mixer Views	143
Using Mixer Views	143
Administering Mixer Views.....	144
Bus Naming	144
Assigning to Mixer Views.....	144
Step 1 Assigning to Mixer views	145
Snapshots	145
Mixer Snapshots	146
Overriding Snapshots	146
Blending Snapshots	146
Creating a Snapshot.....	147
Working with Snapshots	148
Modulation	149
Automation.....	149
Copying Snapshots.....	150
Blank Snapshots	150
Snapshots and Stacking	151
The Snapshot Priority List.....	151
Multiple Instances of the same Snapshot.....	151

Multiple Snapshots with no shared Properties	151
Multiple Snapshots with Shared Properties	151
Priorities	151
Snapshot Automation	152
Snapshot Timeline and Parameters	152
Mixer Routing Window	152
Assigning Mixer Buses to VCAs	153
Add Sends: To Mixing Desk	153
Channels and Panning.....	154
Setting Channel modes.....	154
Step 1 Switching channel output modes.....	154
Using the Panner Controls	156
Concept	156
Stereo Panner.....	156
Surround Panner.....	156
Pan Automation	157
Switching Speakers On and Off.....	158
Mixer Strip Source Control Lock	159
The 3D Panner	160
Min and Max Distance	160
3D Sound Properties.....	160
Sound Size.....	161
Min Extent	162
Pan Override.....	163
3D Preview.....	163
Distance Attenuation	163
Linear Squared	164
Linear	164
Inverse	164
Inverse Tapered.....	165
Turn Off Distance-based Attenuation	165
Event Macros	165
Pitch.....	166
Doppler	166
Scale.....	166
Max Instances	166
Stealing.....	166
Oldest.....	166
Quietest.....	166
Virtualize	166

Off	167
Cooldown.....	167
Priority.....	167
Effect Modules	167
Third-Party and User Created Plugins	168
Effect Automation	168
3-EQ	169
3 EQ Properties and Default Values.....	169
Properties.....	169
Compressor	171
Compressor Properties and Default Values	171
Properties.....	171
Chorus	173
Flanger Properties and Default Values.....	173
Properties.....	173
Delay.....	175
Delay Properties and Default Values.....	175
Properties.....	175
Flanger.....	176
Flanger Properties and Default Values.....	176
Properties.....	176
High Pass	177
High Pass Properties and Default Values.....	177
Properties.....	177
Limiter	178
Limiter Properties and Default Values	178
Properties.....	179
Low Pass	179
Low Pass Properties and Default Values	179
Properties.....	180
Parametric EQ	181
Parametric EQ Properties and Default Values	181
Properties.....	182
Reverb	182
Reverb Properties and Default Values	183
Properties.....	183
Convolution Reverb	184
Distortion.....	185
Highpass Simple.....	185
Lowpass Simple	186

Pitch Shifter	186
Tremolo.....	187
Settings for creating basic LFO shapes:.....	187
Plugin Modules	188
Gain Effect Module	188
Distance Filter Effect Module.....	188
Additional Modules Requiring Licenses.....	189
Sidechaining.....	189
Channel Ducking.....	189
Creating a Sidechain	189
Creating a new Compressor.....	190
Creating the Sidechain.....	190
Localization	191
Overview.....	191
Audio Tables	192
Programmer Sounds	193
Localization API.....	194
Building and Implementing Projects.....	195
Introduction	195
Live Update	196
Enabling Live Updates	196
Banks	196
The Master Bank	197
Assigning Events to a Bank.....	197
Accessing the Events within a Project.....	197
Assigning to a Bank	197
Assigning Events within the Banks Browser	198
Assigning Events within the Banks Browser.....	198
Removing Events from Banks	199
Method 1	199
Method 2	199
Creating New Banks.....	200
Creating a New Bank	200
Bank Audio Tables	200
The Build Process	201
Setting up Control Surfaces	201
Configuring the Device	201
MCU Pro	202
Adding the Device.....	202
SSL Nucleus	203

Software Set-up	203
ipMIDI.....	203
Adding the FMOD Studio Profile to the Nucleus	204
Nucleus Control with Studio	205
Nucleus Profiles	206
Common Controls.....	206
Transport Controls	206
Selection buttons	207
Instrument	207
Plug In	207
EQ	207
Pan.....	208
Send.....	208
Track	208
Send E	208
Pan.....	208
Track 9 Selection Button.....	208
Track 10 Selection Button.....	208
Tracks 11-16 Selection Buttons.....	208
The Shuttle Wheel	209
Global Controls	209
DAW 2 and DAW 3	209
Enter.....	209
Save	210
ESC.....	210
Undo.....	210
F2	210
Miscellaneous Controls.....	210
Flip	210
User 1 & User 2	210
Channel Strips	211
Default Channel Strip Behavior	211
Profiler	212
The Session Browser	213
Adding a Bus to a Session.....	213
Recording a Session	213
Master Bus	214
Playback	215
Graphs	215
Event Instances	217

Loudness Meter	220
Loudness Meter Module	220
Integrated Loudness	221
Loudness Range	221
Max Peak	222
Target	222
Pause	222
Reset	222
Bar Meters	222
Meter Signal Peak and Range Indicator	223
Loudness Histogram	223
Scripting	223
The Shell Interface	223
Scripting Reference	224
Global objects	224
Studio objects	224
Project objects	234
Scripting Examples	237
Creating and deleting an event	237
Checking the tool version	237
Modifying the global context	238
Custom Menus	238
Custom Save & Build menu item	238
Setting master volume	239
Selecting an event and setting timeline position	239
Single Sound Modules vs. Multi Sound Modules	239
Introduction	239
The Basics	239
The Example	240
Taking Things Further	241
A Variable Duration Cycle	241
The Event	241
The Mechanics	242
Going Further	242
Pitch Shifting Magic	243
The Event	243
Going Further	244
Appendix A	245
Keyboard Shortcuts	245
Support	246

Website.....	246
Technical support	246
Sales / licensing enquiries	246
Training and tutorials	246
Credits	246

Core Concepts

To use FMOD Studio, it is first necessary to understand the concepts it is built to support. These concepts are in many ways fundamental to game sound design and how it differs from sound design for media such as film and music.

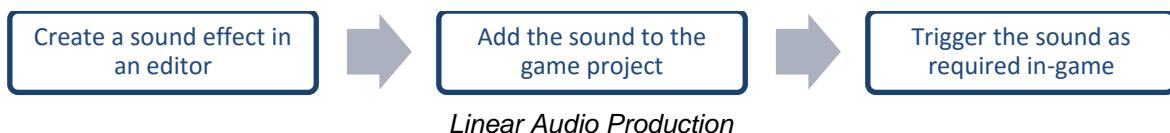
The Sound File is not the Sound

In any game, the circumstances in which a sound plays can vary. The position of the sound's source relative to the player's avatar, its speed of movement, the surrounding environment, and countless other details can change and alter the sound that is played by the game; the audible end product of an in-game event can be very different to the recorded sound file that that event used. Understanding this is crucial to producing good quality game audio.

FMOD Studio can import and play back a single sound file with little or no alteration, but it also includes a variety of tools and features for creating dynamic audio environments. These environments can be as complex or simple as your game needs.

Implementation and Creation: The blurry line

A significant number of game studios still use linear media methods of audio production. That is; create a sound effect in an external editor program, add that sound to a game project through a tool such as FMOD, then trigger that sound as required in game.



While this process is a perfectly functional method of getting sounds to trigger in a game, it misses the opportunity to produce audio that sounds better, uses less resources and requires less programmer time to implement.

High-level game production requires outstanding quality audio, but quality audio is more than just a spectacular sound: It's a sound that is spectacular each time it is triggered, and even unique where appropriate; It is a sound that is as efficient as possible in resource usage; And it is a sound that does not require hours of programmer time to set up complex playback behavior. FMOD Studio combines creation and implementation, and puts both into the hands of sound designers.

A complex sound can be constructed from smaller sound components, and a sound can be modified differently depending on the current states of the game. When done correctly, this results in each sound being unique when triggered because it was assembled in real-time. The product of this is referred to as **dynamic audio**.





Dynamic audio is not an all or nothing solution. There are always times when a sound needs to be consistent, and under those circumstances a well-crafted sound file with no variation can be the best solution. Conversely, there are many situations in which a perfectly consistent sound would be incongruous or odd.

FMOD Versions

FMOD Version numbers have three components: The Product Version number, the Major Version number, and the Minor (or Patch) Version number. In FMOD Studio version 1.09.02, for example, the product version number is “1,” the major version number is “09,” and the minor version number is “02.”

Some FMOD Studio licenses and support contracts only cover specific versions or ranges of versions of FMOD Studio. Consult your license before updating to a new FMOD Studio version.

Minor Releases

In a minor release (also known as a patch release or a point release), the minor version number is incremented, but the other version numbers remain unchanged. Minor releases typically include low-impact efficiency improvements and bug fixes. Minor releases never include significant changes to behaviour, project format, bank format, or features, and are designed to be compatible with all other minor versions within the same major version.

As a result, updating an ongoing game project to use a new minor version of FMOD Studio is typically a safe and easy process.

Major Releases

In a major release, the major version number is incremented and the minor version number is set to 00. Major releases may include significant changes such as new features, changes to existing features and behaviour, changes to project and bank format, and changes to the FMOD Studio programmer’s API.

Because these changes may potentially alter the behaviour of your project or require you to modify your project to work with the new version, it is safest and easiest to adopt new major versions of FMOD Studio near the beginning of your project’s development.

To avoid problems when upgrading to a new major release, we recommend creating a backup of your project before upgrading, and allocate time to thoroughly test your project’s behaviour for unexpected changes after each upgrade.

Platform Support

FMOD Studio currently supports developing content for the following platforms:

- Windows
 - Windows Desktop
 - Windows Store
 - Universal Windows Platform (UWP)
- Linux
- Mac OS X
- iOS
- Apple TV

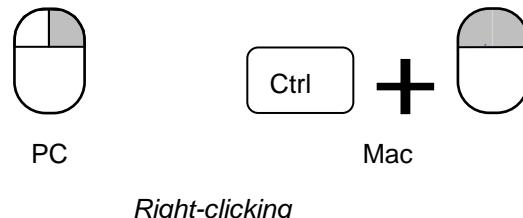
- Android
- Windows Phone
- Xbox360
- Xbox One
- PlayStation 3
- PlayStation 4
- PlayStation Vita
- Wii U

Key Controls

There are some basic and common control and interface conventions that are used frequently throughout FMOD studio. Understanding how these controls work makes exploring the interface and discovering functionality far easier.

Context-sensitive Menus

Throughout FMOD Studio, you can right-click (in Windows) or ctrl-click (on a Mac) on various parts of the interface to open a menu containing commands specific to that context. In some cases, these context menus are the only way to access certain advanced features.



Sometimes, even if multiple items are selected, only the specific one that was right-clicked or Ctrl-clicked is used as the context for the menu that is opened.

Right clicking or Ctrl-clicking various parts of the interface is a good way of discovering FMOD Studio's features and capabilities.

Tooltips

Tooltips are small text windows that explain or describe what is at the current cursor position. They appear when the mouse cursor is held still over part of the interface for a short period of time.

Some tooltips describe the name or function of a tool, icon or other interface element, while others provide the full name of a project element whose displayed name is truncated or abbreviated.

Manual Value Input

When a property's value is displayed, it is usually possible to edit the value of that property by double-clicking on the displayed value and typing in a new one.

*Direct Value Input*

The Menu Bar

The Menu Bar provides access to a number of useful commands and settings. When using the Windows version of FMOD Studio, this menu bar is found at the top of most FMOD Studio windows. On Mac OS X, the menu bar is found at the top of the screen, as usual.

File Menu

The File menu contains commands for managing files. Each of its commands is described below.

New

'New' creates a new empty FMOD Studio project in a new Event Editor Window.

Open...

'Open...' opens a file selector window. If you specify an existing FMOD Studio project in this window, it is opened in a new Event Editor Window.

Open Recent

'Open Recent' opens a submenu containing a list of projects that were recently open in Studio. Selecting any of these submenu items opens the corresponding project in a new Event Editor Window.

Close

'Close' causes the project associated with the current window to be closed, first prompting you to save if there are unsaved changes in the project.

This automatically closes any FMOD Studio windows associated with the closed project; If this results in the closure of all remaining FMOD Studio windows, FMOD Studio automatically exits.

Save

'Save' causes the project to be saved. If the project has been saved before, this menu item overwrites the existing save; if not, FMOD prompts you to specify a name and location for where to save the project.

Save As...

'Save As...' prompts you to specify a name and location for the project. The project is then saved with that location and name.

Revert to Saved...

'Revert to Saved...' returns the project to the state it was in when it was last saved, effectively undoing all changes made since then. Only use this command if you are sure you want to abandon all work done since the last save.

Note that 'Revert to Saved...' has no effect on your project's audio assets. If you have changed your project's audio assets since you last saved, they will not be reverted by this command.

Import Audio Files...

'Import Audio Files...' prompts you to specify a file. If the specified file is in the project's Shared Audio Source directory, it is added to the project's Audio Bin as a reference. Otherwise, it is automatically copied into the project's Audio Asset subfolder.

Once a file is in the Audio Asset subfolder, it can be found in the project's Assets browser and dragged into modules and events as needed.

Build...

'Build...' converts all events and audio assets in the project into bank files that can be loaded and used in a game. This menu item builds banks only for the currently selected platform.

Build all Platforms...

'Build all Platforms' converts all events and audio assets in the project into bank files that can be loaded and used in a game. This menu item builds separate banks for every platform the project supports.

Export GUIDs...

'Export GUIDs...' generates a text file containing the GUIDs (Globally Unique IDentifiers) of the events, snapshots, buses and banks that make up an FMOD Studio project, and informs you of that file's name and location.

Exported GUIDs can optionally be used in the Programmer's API in place of the paths and names of events, snapshots, buses and banks. Most game projects do not require the use of this feature.

Validate Project

This command checks for certain inconsistencies and omissions in the project, most commonly caused by revision control merge errors. If such errors are detected, FMOD provides information about where they might be found and prompts you to fix them.

Connect to Game...

'Connect to Game...' prompts you to specify the IP address of a networked device running a version of your game with live update enabled. Once connected, you can mix and edit your game's audio in real time as the game runs.

Source Control

'Source Control' opens a submenu containing commands for using revision control in conjunction with your project.

Exit

'Exit' prompts you to save any currently open projects with unsaved changes, then exits FMOD Studio.

The Edit Menu

The Edit menu contains commands for making changes to a project's content.

Undo

'Undo' reverses the last change made to the project. It is possible to select this command repeatedly to undo multiple changes. By doing this repeatedly, you can return the project to the state it was in when most recently loaded.

Note that certain rare actions can 'clear your undo history,' preventing you from undoing any actions taken up to that point.

Note also that certain actions are ‘not undoable.’ These actions do not appear in the Edit menu as undo commands, and cannot be undone, but do not prevent you from undoing other changes. Most but not all ‘not undoable’ actions do not meaningfully change the behavior of your project. Changes to a project’s audio assets are never undoable.

Redo

‘Redo’ reverses the effects of ‘Undo,’ reversing the reversal of a change. This command can be selected repeatedly to redo multiple changes.

Note that making a change to the project by means other than selecting Undo or Redo renders the redo action unavailable, as if no actions had been recently undone.

Copy

‘Copy’ puts a copy of whatever is currently selected in the event editor on the clipboard.

Paste

‘Paste’ places the contents of the clipboard in the current window. If pasting an event module, the pasted module is pasted to the current track, on the currently-visible Parameter, and in the parameter cursor’s position.

Delete

‘Delete’ removes from the project whichever elements are selected in the currently active area of the current window.

Duplicate

‘Duplicate’ creates an exact copy of the current selection in the browser, exactly as if ‘Copy’ and ‘Paste’ were used in conjunction. Only events, event folders, profiler sessions, profiler folders and snapshots can be duplicated.

Move To...

‘Move To...’ prompts you to specify a new location and length for the selected item. Note that this command is only available if the selected item is a trigger region.

In the ‘Move To’ window, the length or duration of a trigger region is usually locked. This means that if the Start or End position of the trigger region is changed, the other value is automatically updated such that the difference between the values remains constant. Clicking on the padlock button toggles the locked state of the trigger region’s length on or off.

Move to Cursor

‘Move to Cursor’ repositions the currently selected trigger region to the location of the Parameter cursor. This does not change which Parameter or track the trigger region is on. This menu item is only available when the current selection is a trigger region.

Bring to Front

‘Bring to Front’ rearranges the ordering of trigger regions in your event such that the selected trigger region is in front of any other trigger regions it overlaps. When you click on an area where multiple trigger regions overlap, the trigger region closest to the front is the one that becomes active. This menu item is only available when the current focus is a trigger region.

Send to Back

‘Send to Back’ rearranges the ordering of trigger regions in your event such that the selected trigger region is behind any other trigger regions it overlaps. When you click on an area where multiple trigger regions overlap, the trigger region closest to the front is the one that becomes active. This menu item is only available when the current focus is a trigger region.

Set Color

'Set color' opens a submenu of commands that change the color of the currently selected trigger region. This menu item is only available when the current focus is a trigger region. The color of a trigger region has no impact on your project's behavior.

Open in Explorer

Use this option when an audio file is selected (either in the Assets browser or in a sound module) to display the file in a Windows Explorer or Finder window.

Note that the the file used by your FMOD Studio project may be a copy of a file that also exists elsewhere on your drive. If this is the case, this menu item targets the copy used by your project.

Open in External Sample Editor

'Open in External Sample Editor' opens the selected sound file in the application associated with its file type. This command is only available in an Audio Bin window when an audio file is selected in that window.

Note

Studio does not discriminate between audio file editor applications and other applications. If the application associated with your source audio's file type is a media player, the 'Open in External Sample Editor' command will open your source audio in that media player.

If you want to specify a different audio editor without changing your operating system's file type associations, you can do so in the 'Assets' tab of FMOD Studio's 'Preferences' window.

Preferences...

'Preferences...' opens the 'Preferences' window.

The View Menu

The View Menu contains commands that change what is displayed and auditioned.

Browser

'Browser' hides or displays the Browser in the current window. This command is marked by a checkmark if the Browser is currently visible. This command is only available in windows with a Browser that can be hidden.

Deck

'Deck' hides or displays the Deck in the current window. This command is marked by a checkmark if the Deck is currently visible. This command is only available in windows with a Deck.

Properties

'Properties' hides or displays the Overview in the current window. This command is marked with a checkmark if the Overview is currently visible. This command is only available in windows with an Overview that can be hidden.

Loop Playback

'Loop Playback' toggles Studio's loop playback audition mode on or off. This command is marked with a checkmark if loop playback audition mode is currently on. Events that are auditioned while this mode is on are automatically restarted when they reach their natural ends.

Note that Loop Playback mode affects auditioning only, and has no effect on the behavior of events in your game.

Follow Cursor

'Follow Cursor' toggles Studio's Follow Cursor audition mode on or off. This command is marked with a checkmark if Follow Cursor audition mode is currently on. When auditioning while viewing a Timeline Parameter in this mode, the editor automatically scrolls to keep the Timeline cursor visible.

Snap to Items

'Snap to Items' toggles Studio's Snap to Items mode on or off. This command is marked with a checkmark if Snap to Items mode is currently on. While in this mode, dragged trigger regions and markers are automatically aligned to nearby logic markers and trigger regions.

Snap to Ruler

'Snap to Ruler' toggles Studio's Snap to Ruler mode on or off. This command is marked with a checkmark if Snap to Ruler mode is currently on. While in this mode, dragged trigger regions are automatically aligned to the nearest notch in the Parameter or timeline ruler. When unchecked, dragged trigger regions can be dragged to any possible position.

Note that Snap to Ruler mode only functions on the Timeline if there is a tempo marker to the left of the item being dragged.

Note

Sometimes you might want to override the Snap to Ruler and Snap to Items modes only briefly. Rather than having to open the View menu, you can simply hold down the Ctrl key. This modifier key allows objects to be moved as if the Snap to Ruler and Snap to Items modes are disabled.

Zoom In

'Zoom In' redraws displayed content such that it appears larger, allowing the finer details of that content to be seen more easily.

Zoom Out

'Zoom Out' redraws displayed content such that it appears smaller, allowing more of that content to be viewed.

Zoom to Fit

'Zoom to Fit' redraws the content of the currently displayed Parameter or timeline at a scale small enough for all of that content to be visible at once.

The Window Menu

The Window menu contains commands that open, manipulate, and navigate between windows.

Minimize

'Minimize' minimizes the current window, exactly as if you had clicked on its minimize button.

Zoom

'Zoom,' on Mac OS X, zooms the current window exactly as if you had clicked on its zoom button. On Windows, it restores or maximizes the current window, exactly as if you had clicked its restore button.

Cycle Through Windows

'Cycle Through Windows' takes you to another open FMOD Studio window. Selecting this menu item repeatedly cycles through all your open windows according to the order in which they were opened.

Bring All to Front

'Bring All to Front' brings all FMOD Studio windows in front of any other windows that are open, unless those windows always appear on top of other windows.

Event Editor

'Event Editor' always causes a new Event Editor window to be opened if it is selected from an existing Event Editor window. Otherwise, it presents you with an existing Event Editor window, or opens a new Event Editor window if one is not already present.

An event editor window allows you to organize, audition and edit events. Having multiple event editor windows open allows you to easily audition more than one event at a time.

Mixer

'Mixer' always causes a new Mixer window to be opened if it is given from an existing mixer window. Otherwise, it presents you with an existing Mixer window, or opens a new Mixer window if one is not already present.

A Mixer window contains tools for mixing a project.

Audio Bin

'Audio Bin' presents you with an existing Audio Bin window, or opens a new Audio Bin window if one is not already present.

The Audio Bin window allows you to view, audition and organize a project's audio files.

Event Browser

'Event Browser' presents you with an existing Event Browser window, or opens a new Event Browser window if one is not already present.

An Event Browser window allows you to create, assign and organize a project's Events and Tags.

Mixer Routing

'Mixer Routing' presents you with an existing Mixer Routing window, or opens a new Mixer Routing window if one is not already present.

A Mixer Routing window allows you to modify your project's bus routing.

Profiler

'Profiler' always causes a new Profiler window to be opened if it is given from an existing Profiler window. Otherwise, it presents you with an existing Profiler window, or opens a new Profiler window if one is not already present.

A Profiler window allows you to monitor and record your project's performance.

Contextual List of Windows

Near the bottom of the Windows menu is a list of FMOD Studio windows that are currently open. Any of these commands presents you with the corresponding window.

Close Window

'Close Window' closes the current window, first prompting you to save if there are no other windows associated with the project and the project has unsaved changes. If this results in the closure of the last remaining FMOD Studio window, FMOD Studio automatically exits.

Note that Audio Bin, Event Browser, and Mixer Routing windows do not count for the purposes of determining whether there are any remaining windows.

The Help Menu

The Help menu contains commands that give you more information about FMOD Studio.



The Help Menu

Manual

'Manual' opens the FMOD Studio user manual that you are reading currently.

Visit the Questions Page...

'Visit the Questions Page...' opens our Q&A page in your web browser. There you can browse a wealth of questions, answers and discussions of FMOD products, or ask new questions of your own.

Contact support@fmod.org...

'Contact support@fmod.org...' allows you to contact Firelight Technologies by e-mail. Under most circumstances, we respond to any request or question within 24 hours, and all support e-mails are answered directly by the developers who make FMOD Studio.

To ensure speedy resolution of a problem, we advise you to include as much information about it as possible, preferably including:

- Product and product version (e.g.: FMOD Studio, v1.08.01).
- Platform and platform version (e.g.: Windows 10, iOS 4.0).
- Steps that allow us to reproduce the problem.
- Any output from error messages and logging tools.

About FMOD Studio

'About FMOD Studio' displays the version and changelist numbers of FMOD Studio, as well as release notes and some legal information.

This window also allows you to open our examples project, view this user manual, and enable or disable the logging of anonymous usage statistics. (We use these statistics to learn which features are most commonly used.)

Common Control Types

Dials

Various properties are represented by dials. These dials resemble knobs (or 'pots') of the kind that appear on many traditional hardware devices.



A Volume Dial

The notch on a dial indicates the specific value to which it is set. A dial also displays the range of values between its current value and its default value as a colored band, and may display the presence and amount of randomization modulation it is subject to in a similar way.

To adjust the value of a property controlled by a dial, click on the dial and hold down the mouse button while dragging the mouse up and down.

Fine Tuning Dials

Holding down the ‘Ctrl’ key (or, on a Mac, the ‘Command’ key) while adjusting a dial causes the dial’s property to change by smaller increments.

It is also possible to precisely specify a new value for most dial properties by double-clicking on the existing value display and typing a new one, as described in the Manual Value Input section.

Randomization Modulators on Dials

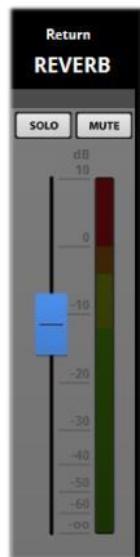
If a property controlled by a dial is affected by a randomization modulator, holding down the ‘Alt’ key while adjusting the dial alters the value of the randomization modulator. If this is done to a dial that is not currently subject to a randomization modulator but could be, a randomization modulator is automatically added to it.

Sliders

Many properties of effects, buses and event tracks can be controlled by sliders. Sliders resemble the faders found on some hardware control surfaces.

Sliders can be horizontal or vertical, but in either case the handle of a fader represents the property’s current value. Some properties can be assigned both a maximum and a minimum value, and sliders represent this with two handles connected by a ribbon, known as a “ribbon slider.”

To adjust the value of a slider, click and drag the handle to a new position. In the case of a ribbon slider, you can also drag the ribbon to reposition both handles at once.



A Mixer Strip, Featuring a Slider

Fine Tuning Sliders

Holding down the 'Ctrl' key (or, on a Mac, the 'Command' key) while dragging a slider handle causes the slider's property to change by smaller increments.

It is also possible to precisely specify a new value for some slider properties by double-clicking on the existing value display and typing a new one, as described in the Manual Value Input section.

Note

All Mixer Strip sliders have an associated dial that can be viewed in the deck, and any property represented by a dial in the deck can be represented by a slider in the Strips view by using the 'Flip to Faders' context menu item. In either case, both the dial and the slider represent the same underlying property value. It is not possible to adjust one without adjusting the other, as they both represent the same property.

Bulk Editing Sliders

If you multi-select a number of strips while in the Strips view, dragging one of the selected strips' slider handles causes the handles of all selected strips to move. Holding down the 'Shift' key while dragging a slider handle causes all handles in the selection to be set to the same value as the dragged handle.

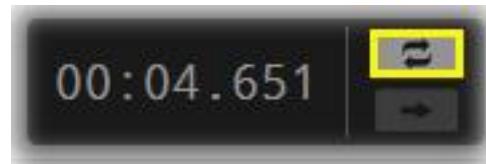
Buttons

Buttons appear throughout FMOD Studio. Clicking on a button causes a specific action or change associated with the button to occur.

Several types of button are used within FMOD Studio. Some of the more complex ones are described here.

Toggle Buttons

Some properties and settings have exactly two possible states, and can only be in one of them at a time. A toggle button is a single button whose appearance indicates the current state, and that causes the current state to change when clicked.



The Loop Playback Mode Toggle Button

Examples of toggle buttons include the Solo and Mute buttons found on track heads and the Loop Playback and Follow Cursor playback mode buttons in the transport bar.

Radio Buttons

Radio Buttons exist where a property or setting has multiple discrete possible states of which only one can be active at a time. Radio buttons are always arranged in tight clusters of buttons where each button represents a single possible state of the property or setting, and the button representing the current state is highlighted.

Changing states is a simple matter of clicking on a different button.



Radio Buttons

Manual Value Input

Many properties and settings allow you to type in a new value for the property using a computer keyboard. To do this, double-click on the existing value of the property to make a value input field appear, type the new value, then press the 'Enter' key.

If a value input field requires input in a specific format or range, it automatically ignores and discards any inappropriate values.



Keyboard Value Input

The Preferences Window

The Preferences window contains a number of settings that affect the project, as well as FMOD Studio as a whole.

Audio

Metering channel order

FMOD Studio provides three different options for the order in which surround meters are displayed.

- ‘Standard’ displays meters ordered from left to right by how likely it is that the corresponding speaker is present in your speaker setup. This mimics the default order used in many Digital Audio Workstation applications.
- ‘Separate LFE’ is similar to Standard, but positions the LFE meter to the right of all others.
- ‘Positional’ orders the meters based on the relative locations of the corresponding speakers as perceived by a listener.

The default setting is Positional.

Assets

Assets folder (optional)

In order to ensure that projects are easily portable, FMOD Studio’s default behavior is to copy each file imported into a project into a subfolder of that project, and to use that copy in the project instead of the original file. Setting an ‘Assets folder (optional)’ for the project allows you to override this behavior: If you import a file that exists in the specified directory, Studio uses the file in that location, and does not copy it into the project subfolder.

External Sample Editor (optional)

‘External Sample Editor (optional)’ specifies the third-party application that is used to open an audio asset when you select the ‘Open in External Sample Editor’ command. If no directory is specified, FMOD Studio defaults to using whichever application your operating system associates with the file’s type.

Plug-ins folder (optional)

FMOD Studio allows you to use third-party Plugins. The ‘Plugins folder (optional)’ setting dictates where FMOD Studio looks for Plugins.

Note that some plugins may require additional libraries if they are to be used in-game as well as in FMOD Studio.

Build

Built banks output directory (optional)

‘Built banks output directory (optional)’ specifies the directory in which FMOD Studio deposits built banks when the project is built. If no path is specified, Studio defaults to outputting the built banks into the ‘Build’ subfolder of the project folder.

Control Surfaces

The Control Surfaces tab provides a drop-down menu of the available supported control surfaces that can be used to control FMOD studio. Once a control surface is selected, further settings specific to that device become available.



The Control Surfaces Tab

Source Control

The ‘Source Control’ Tab provides a drop-down menu of available supported source control methods. Once a revision control method is selected, further settings specific to that method become available.

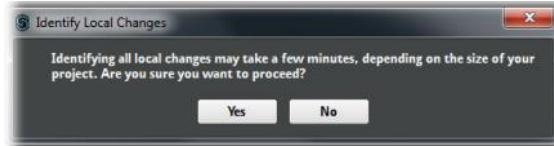
The Event Editor Window

The Event Editor window is the first window that opens when FMOD Studio is run. This window contains a variety of features and tools for creating, editing and organizing ‘events,’ the primary units of sound design in FMOD Studio.

There are three browsers that can appear in an Event Editor window: the Event browser, the Banks browser, and the Assets browser. Only one of these can be displayed at a time. You can change which browser is displayed by clicking on the tabs at the top of the browser.

Identify Local Changes

The ‘Identify Local Changes’ command is located in the Events Browser context sensitive menu, but only appears when using revision control. It compares the current version of your project with the last logged version on your source control server. Studio displays any differences in a log. This allows you to identify and check in any changes made locally while not connected to your source repository.



The Identify Local Changes window

When merging versions of a project, you are prompted to choose between accepting your (local) changes and “their” (serverside) version.

The Events Browser

The events browser displays your project’s events and event folders. It allows you to create, assign and organize events.

You can filter which items are displayed by clicking on the search bar at the top of the browser and typing part of the name of the item you want to see, or by clicking on the loupe icon and selecting an option from the loupe menu.

Clicking on an item selects it and makes it the ‘active item’ displayed in the Editor and Properties panes. Hold down the ‘Shift’ or ‘Ctrl’ keys when clicking on items to select multiple items, allowing you to edit or manipulate them as a group.

If you drag an item from any browser over another browser’s tab, the browser associated with that tab displays.

The Banks Browser

The banks browser displays your project’s banks and bank folders, as well as the events and audio tables assigned to each bank. It allows you to create, edit and organize your project’s banks, as well as to assign events to banks.

As in the events browser, you can filter which items are displayed by clicking on the search bar at the top of the browser and typing part of the name of the item you want to see.

Clicking on an item selects it and makes it the ‘active item’ displayed in the Editor and Properties panes. Hold down the ‘Shift’ or ‘Ctrl’ keys when clicking on items to select multiple items, allowing you to edit or manipulate them as a group.

If you drag an item from any browser over another browser’s tab, the browser associated with that tab displays.

The Assets Browser

The assets browser displays the audio files and folders in your project’s assets directory. It allows you to audition, edit and organize your project’s audio files, as well as to add those files to sound modules and events.

As in the events browser, you can filter which items are displayed by clicking on the search bar at the top of the browser and typing part of the name of the item you want to see.

Clicking on an item selects it and displays its properties at the bottom of the assets browser. Hold down the ‘Shift’ or ‘Ctrl’ keys when clicking on items to select multiple items, allowing you to manipulate them as a group.

If you drag an item from any browser over another browser’s tab, the browser associated with that tab displays.

To add an audio file to an audio track or playlist, click and drag the audio file from the assets browser onto the target event or playlist.

Note

Assets Browser represents the actual files and folders in your project’s Assets directory. Any change to the items displayed in the Assets Browser represents a change to those files and folders, and cannot be undone via the ‘Edit > Undo’ menu item.

Context Menu

The assets browser’s context menu contains a number of commands for managing your project’s assets.

Open in Explorer/Open in Finder

This command displays the selected file or folder in a new Finder or Windows Explorer window, allowing you to locate it without having to navigate to it manually.

Open in External Sample Editor

Open in External Sample Editor opens the selected file in the application associated with the file's filename extension in your OS. You can specify a different application in the Assets tab of the Preferences window.

Move into New Folder

Move into New Folder creates a new folder in the current location, and puts the currently selected items inside it.

Move to Top Level

Move to Top Level removes the currently selected items from their current locations and puts them in your project's assets folder.

Move To

Move To opens a submenu of subdirectories within your project's assets directory. Selecting any of these subdirectories moves the selected files and folders to the chosen directory.

New Folder

New Folder creates a new folder in the selected location.

Refresh

Refresh checks the selected file for changes, and updates all events that use the file to use the new version. If the new version is a different length, this causes the trigger regions of timelocked single sound modules that use the file to be rescaled to match the new file length.

Consolidate...

This command only appears if multiple files are selected. It replaces all references to files in the selection with references to the file on which you right-clicked, and removes the other files from the assets directory.

This allows you to replace a single file with a new version throughout your project, as well as to consolidate multiple different versions of a file with a single replacement.

Delete

Delete removes the selected items from your assets directory. You will be prompted to confirm your decision.

The Editor

The Editor allows you to create and change the content of events. This includes the game Parameters and trigger regions that define when and how the audio content of an event generates a signal, as well as the internal buses, routing, effects and automation that manipulate that signal.

By default, a new event is created with one Audio track, a Master track and the Timeline for the event. In the (default) Tracks view you can add Parameters and place sound modules on the Audio tracks as required. The logic track for your event is located above the audio tracks in the Tracks view and you can add loop regions, transitions and other logic markers to it as required.

Context sensitive menus within the editor allow you to set up routing within your event.

When auditioning an event (by pressing Play), you can manipulate the current values of its Parameters either manually or by using the transport controls. This means that you can listen to how

the event sounds in different states without making any lasting changes to the way the event behaves in-game. Changing any part of an event other than its current parameter values alters the in-game behavior of that event.

Breadcrumbs

Breadcrumbs appear when viewing the content of an Event Sound Module or Event Reference Sound Module. From left to right, the breadcrumbs indicate the path you took to navigate to the current event, starting with the event selected in the Events Browser. Clicking on a breadcrumb takes you back to the corresponding event.



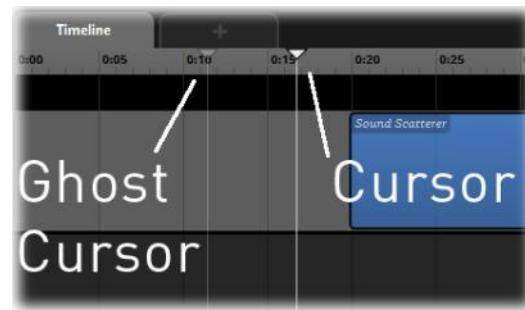
Breadcrumbs

Playheads and Cursors

The cursor, also known as the playhead, is a white vertical line found on all timelines and Parameters. The cursor indicates the current value of the Parameter.

Each timeline and Parameter also features a ghosted cursor that indicates the value to which you most recently set the Parameter value. This is the value to which the cursor position is returned if the event is reset.

Clicking on a track or the ruler while viewing a Parameter moves that Parameter's cursor and ghosted cursor to the location clicked. Note that if the Parameter has a seek speed and the event is currently being auditioned, the cursor does not move instantaneously.



The cursor and ghosted cursor

Missing References

Occasionally, when using revision control, the components of a project may come to be out of synchronicity with each other. A send effect module, for example, may point to a return bus that has been deleted, or a sound module to an audio file that has been removed. This phenomenon is known as a 'missing reference,' and is indicated by an exclamation mark icon or red text.

These 'missing reference indicators' indicate that the corresponding project element no longer serves any function, and should be fixed, replaced or deleted as best suits your project's requirements.

Transport Bar

The Transport bar contains tools for auditioning the currently displayed event.

Stop Button

The stop button can have multiple possible effects, depending on the circumstances.

If the event is currently auditioning, holding down the stop button for more than half a second triggers any stopping behavior that the event has, and stops the event. Note that if the event has no stopping behavior, this is functionally identical to clicking the stop button twice.

If the event is currently auditioning, clicking the stop button pauses the audition. This does not audition any stopping behavior that the event has. A paused event can be resumed by clicking the Play button, or reset by clicking the Stop button again.

If the event is already paused, clicking the stop button immediately stops the event and resets the timeline cursor to the position of the ghost cursor. This does not audition any stopping behavior that the event has.

If the event is already stopped, clicking the stop button immediately moves the timeline cursor and ghost timeline cursor to the beginning of the timeline.

Play Button

The play button has multiple possible effects, depending on the circumstances.

If the event is paused or stopped, clicking the play button auditions the event. If an audition was previously paused and not stopped, this effectively causes it to resume.

If the event is currently auditioning, clicking the Play button resets the audition, causing the event to play from its beginning.

Timecode Display

The Timecode Display shows the current value of the Timeline parameter.

Clicking on the Time Display allows you to type new time values.

Note that the value of the timeline Parameter is not necessarily the amount of time that has elapsed during the current audition. Besides the passage of time, the value of the timeline Parameter can also be altered by game code, logic markers, and event pitch adjustment.

Loop Playback Button

The Loop Playback button toggles the Loop Playback audition mode on or off. While the mode is on, auditioned events automatically reset and restart whenever they reach a natural end. While it is off, auditioned events that reach natural ends continue to play until manually stopped.

Note that loop playback mode has no effect on the behavior of events in your game. It is an auditioning tool only.



Loop Playback

Follow Cursor Button

The Follow Cursor button toggles Follow Cursor mode on and off. While this mode is on, the Editor automatically scrolls to keep the cursor visible when an event is auditioning. Manually scrolling horizontally automatically turns off Follow Cursor mode.

Birdseye View

The Birdseye view, located directly below the editor, is a scale representation of the view currently displayed in the editor. The reticule in the Birdseye view represents the part of that the Strips or Tracks view that is currently visible in the editor.

Repositioning the reticule changes which part of the view is displayed in the editor. The reticule may be repositioned by clicking and dragging it to a new location or by clicking on a part of the Birdseye view it does not cover.

While using the Tacks view, the reticule may be resized by clicking and dragging its left or right edge. This horizontally zooms the editor.

The Deck

The deck displays the module associated with the trigger region that's active in the editor, the signal chain of the active track, or the properties of a recently-selected parameter. Otherwise, the event's macro controls are displayed instead.

The collection of modules and properties displayed in the deck when a particular element is selected is frequently called the deck of that element.

Almost any property represented by a dial in the deck can be affected by modulators and automation. In addition, when the deck displays the signal chain of a track, it is possible to add and remove effects in that signal chain by using the context-sensitive menu, and effects modules can be rearranged by clicking and dragging them to new positions.

By default, the master track of a newly created event contains a 3D Panner effect module and a send effect module that sends to the 'Return' bus, in the mixer.

Note

There is a Deck in the Mixer window as well. There are many similarities between the Event Editor Deck and the Mixer Deck, but they are not identical. The most important differences are that the effects modules and modulators available in each window differ, the Mixer Window's deck displays the signal chains of project buses instead of event tracks, and editing automation in a Mixer Window is possible only inside snapshots.

Refer to the sections on each window's version of the deck for more details.

3D Preview

The 3D Preview appears in the top of the overview pane when viewing a 3D event in the Editor. It allows you to audition the position and orientation of the event relative to the listener. The center of the circle represents the listener, and the circle represents the space around it, with the upper half of the circle representing the area in front of the listener, and the lower half representing the area behind. The edge of the circle is the maximum distance of the event.

The arrow-shaped icon represents the event emitter. The location of the emitter icon indicates the emitter's position in the space surrounding the listener, and the direction of the emitter icon's arrow indicates the direction in which the emitter faces.

Clicking and dragging in the 3D Preview moves the emitter, allowing you to audition it at different positions relative to that of the listener. Holding down the control key while dragging away from the center of the circle increases the altitude of the emitter, and holding the key while dragging towards the center lowers it. Spinning the mouse wheel rotates the emitter.

Note that repositioning or rotating an event emitter has no audible effect unless the event has some feature that changes based on its position or orientation. Therefore, changing the position of the emitter only results in audible change if the event has automation curves or trigger regions on its Distance or Elevation Parameters or has a 3D Panner effect module, and rotating an emitter only results in an audible change if the event has automation or trigger regions on its Direction, Event Cone Angle or Event Orientation Parameters.

Double-clicking on the 3D Preview resets the emitter to its default position and orientation.

Properties

The Properties section of the Overview pane displays the active event's tags, user properties and notes.

Tags are keywords used to navigate to and search for events in FMOD Studio. Tags are not included in built projects. For more information about tags, see the Tags section.

User properties are customizable variables that are included in built projects and can be accessed by your game's code. The names and values of these Properties are specified by users in Studio and that can be inspected and used in game code via the Programmer's API. The value on the left side of each pair is the name of the property and the value on the right is its value. More information can be found by searching for `getUserProperty` in the Programmer's API documentation.



User properties can be named and assigned values as required

The most common use of user properties is to indicate to your game that certain events should be treated differently.

The notes field of an event can contain any text you choose. This is useful for communicating the details and purpose of an event to other people working with the same FMOD Studio project. Notes are not included in built projects.

The Event Browser Window

The Events Browser

The Event browser displays your project's events and event folders. It allows you to create, assign and organize events.



Event Browser Options

The Tags Browser

The Tags browser is where you can sort and organize all of the tags in your project. Here you can sort by name (alphabetically) and organize your events and their associated tags. Expanding each tag highlights all the events associated with that tag. You can also drag tagged events into other tags to assign them to those tags.

Actions available in the Tags Browser

- Create a Tag: Select ‘New Tag’ from the context-sensitive menu to create a new tag.
- Delete a Tag: Select ‘Delete [tag name]’ from the context menu to delete a tag.
- Open an Event in a new window: Right click on an Event in the Tags browser and select ‘Open in New Window’ from the context-sensitive menu. This opens a new Event Editor window with that Event open.
- Remove a Tag from an Event: Select ‘Remove “[tag name]” from “[event name]”’ to remove a tag from an event.
- If you drag an item in either browser over the other browser’s tab, the browser associated with that tab displays.

The Mixer Window

The mixer window contains tools and features that allow you to set up routing and mix your project, to apply effects to its signal chains, and to define how those effects and that mix can change in response to game events.

In addition, all of these features can be used while connected in real time to your game as it runs.

By default, all Events created in your project are routed directly to your project’s Master bus. The Mixer window allows you to supplement your events with group and return buses. These return and group buses can then be routed into group buses, and their signal chains can be populated with sends to return buses and effects modules. These features resemble those of hardware mixers.

You can also create snapshots. When active, a snapshot alters the value of any effect property or bus volume scoped into it. Volumes and properties not scoped in to a snapshot are left unchanged, meaning that multiple snapshots can be active at once without overriding each other if they apply to different parts of the project.

Mixing Desk

The Mixing Desk View is a simple visual representation of a selection of buses. Buses are displayed as vertical strips to the left of the Master Bus in the Mixing Desk.



Mixer View

You can create a custom view of mixer objects for your convenience. Although Mixer Views do not affect the in-game behavior of their contents, they can be a handy tool when working in larger projects to easily find and view specific objects or groups of objects.

Any Event, Group bus, Return bus or VCA can be added to a Mixer view. This means you can, for example, set up a specific Mixer view that contains all of a project's character sounds, another for enemy sounds and a third one for environmental sounds. Right clicking on the tab lets you create or remove a Mixer View. You can also open a view in a new window. Double click on a Mixer View tab to rename it.

The Mixer Browser

The Mixer Browser is used for routing signal path information in the Mixer. Each Event in your project appears in the routing browser. These are all by default routed directly to the Master Bus for playback output.

You can create more buses or change their routing.

If you drag an item in either the Routing or Snapshots browsers over the other tab, the browser associated with that tab displays.

Snapshots Tab

Snapshots are a method of selecting a list of properties that you want to affect and specify new values for the properties when the snapshot is playing. You can freely add as many or as few properties to the snapshot as you need by 'scoping in' and 'scoping out' individual properties until you have a list that suits your purpose.

Snapshots only apply their property values when currently playing; Snapshots not being played have no effect on their scoped properties.

This means that two snapshots might affect completely different sets of properties. For example, one of the most common uses of snapshots is to reduce the volume of non-dialog channels during in-game conversations. For this purpose, you'd make a snapshot that includes only the volumes of those buses that needed to be ducked - often just a handful of group buses.

Snapshots can be used individually or in conjunction. It is possible to set up a transition between Snapshots.

When used individually, FMOD Studio transitions the values of properties scoped into a snapshot from the current values specified in the mixer to the Snapshot's values.

Selected Buses Tab

In a larger project with many buses, you may want to view only a specific selection of Buses. You can do this by selecting all of the desired objects in the Mixer Brower and then clicking the Selected Buses Tab. You can select and deselect different Buses as desired

The Deck

The Deck displays the controls of the selected Channel strip.

The Audio Bin Window

The Audio Bin window contains tools and features for managing and organizing the source audio files used by your project.

The Assets Browser

This is the same assets browser that appears in the Event Editor window. See page 32 for more details.

FMOD.io

FMOD.io is an online service that allows you to purchase and download audio files for use in your projects.

The FMOD.io tab of the Audio Bin window takes you straight to the FMOD.io page, where you can browse, preview, and purchase audio files. You can also download purchased files directly into your project's assets directory.

Tracks View and Strips View

The tracks view and strips view are different ways in which Studio can display content. In the Mixer Window, these views display snapshots. In the Event Editor Window, these views display events. When a snapshot or event is displayed, you can switch between views by clicking on the two buttons found to the right of the Loop Playback and Follow Cursor toggle buttons.



Tracks and Strips View Options

Tracks View

The Tracks view is the default view in the Event Editor. This view displays the parameter-based contents of the active event or snapshot. This allows you to create and edit automation, as well as to change the routing of event tracks and buses. In the Event Editor window, the tracks view additionally allows you to add sound modules to the active event's tracks.

Tracks

In the Tracks view you can see the tracks associated with your event or snapshot. For an event, these tracks include the event's audio tracks, return tracks, and automation tracks, as well as its master track. For a snapshot, the tracks represent the return and group buses scoped in to the snapshot, as well as automation tracks.

Each track displays the trigger regions that can be triggered by the currently-selected parameter.

By default, the displayed parameter is the timeline. You can display different Parameters (or add new ones) by clicking on the Parameter tab(s) at the top of the Editor.

Arranging Tracks in an Event

In an event, the tracks are displayed in the order they were created, with the Master track at the bottom. Return tracks are always located below the audio tracks and above the Master track. The order of audio and return tracks can be changed by dragging them into new locations.

Color Coding Audio Tracks

Complex events and snapshots may have so many tracks that they become difficult to manage. A quick way to visually sort your audio tracks is to set a different color for each track or group of tracks. Right click on the track header and select Set Color to assign a color to a track.

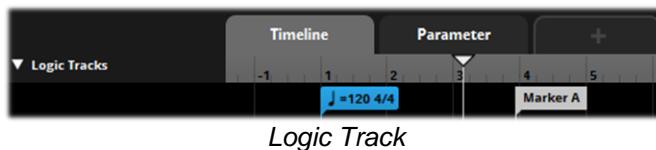
Audio Track Output

By default, all Audio and Return tracks of an event output to that event's Master Track.

You can change the output of an audio track to create a sub mix for the event. This is how you would mix each instance of the event prior to outputting its signal to the project mixer from the Master track.

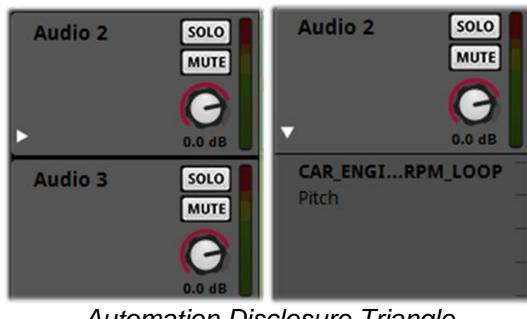
Logic Track

Click on the white disclosure triangle to display or hide your event's Logic Tracks.



Automation

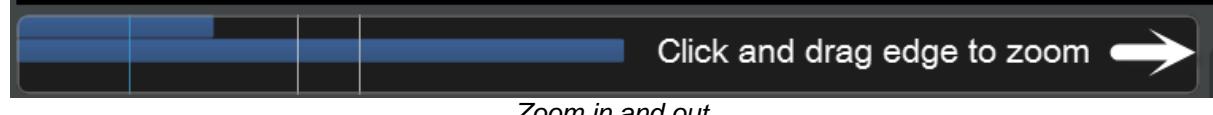
If an event contains automation, additional disclosure triangles are present on the head of each track on which automation is present. Use these to hide or reveal the automation tracks associated with these tracks.



Automation Disclosure Triangle

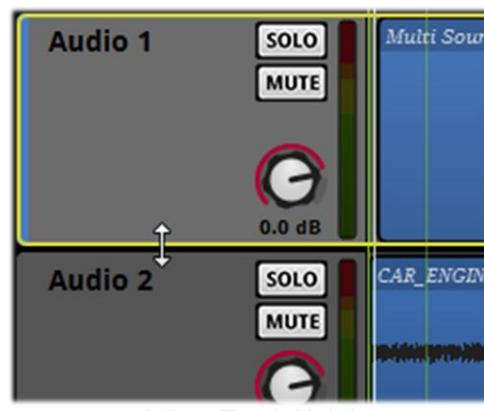
Navigating the Tracks View

You can zoom in and out using the Birdseye View. Left click on the right edge of the reticule and drag left to zoom in and right to zoom out.



If your event contains more audio tracks than can be displayed at once, Studio automatically creates a white scroll bar at the right side of the window. Click and drag this bar up and down to display the desired tracks.

If you want to reduce or increase the size of a track, simply move your mouse to the bottom of the head of that track and drag it up and down.

*Adjust Track Height*

You can also use the reticule in the Birdseye view to quickly navigate larger projects by dragging it around the view.

*Navigating via the Birdseye View*

Strips View

The Strips View displays the mixer strips of the selected event's tracks, or of the project's buses. This lets you easily compare buses for mixing within your event. The Editor displays the tracks view by default, and the Mixing Desk the strips view.

In the case of the Mixing Desk, the Tracks view is not available unless you have set it to display a Snapshot.

*The Strips View*

Color Coding Strips

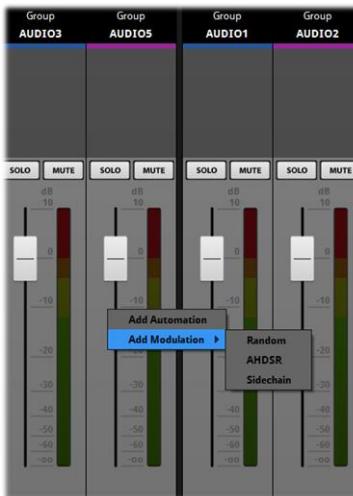
If you have color coded your Audio tracks, the color code is also displayed in the Strips view. You can change or set the colors in this view by right clicking on a track header and selecting "Set Color."



Setting Color in the Strips View

Adding Modulation

Right clicking on sliders in the Strips view allows you to add automation and modulation to properties.



Adding Automation Modulation in the Strips View

Navigating the Strips View

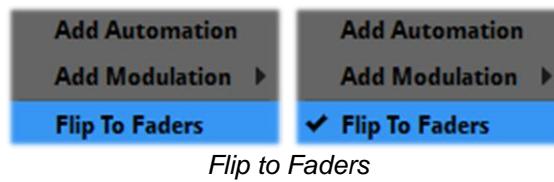
Unlike the Tracks view, which has both horizontal and vertical navigation, the Strips view is laid out entirely horizontally. Use the bottom scroll bar or the Birdseye view to scroll left and right through the strips. The Birdseye view in the Strips view has no zooming function.

The sliders that appear on mixer strips usually represent the volume of the corresponding Bus or track.

Flip to Faders

Sliders in the Strips view can represent properties other than volume.

Right click on a property dial in the deck and select the 'Flip to Faders' menu item to view and adjust other properties using the Strips view. To revert to viewing the volume property, right click on the property dial and select the Flip to Faders option again.



Flip to Faders

Bulk Editing Mixer Strips

If you want to edit more than one Mixer Strip at the same time, hold shift or ctrl to select multiple strips before adjusting their sliders. Sliders can be edited relatively (by dragging one of the selected strips' sliders) or absolutely (by holding the 'Shift' key while dragging).

Compact Strips Display

If there are a large number of strips to display and you want to view more strips at once, go to View > Compact Strips. This reduces the size of each individual strip. The shortcut key "C" toggles this option on and off.



Compact Strips

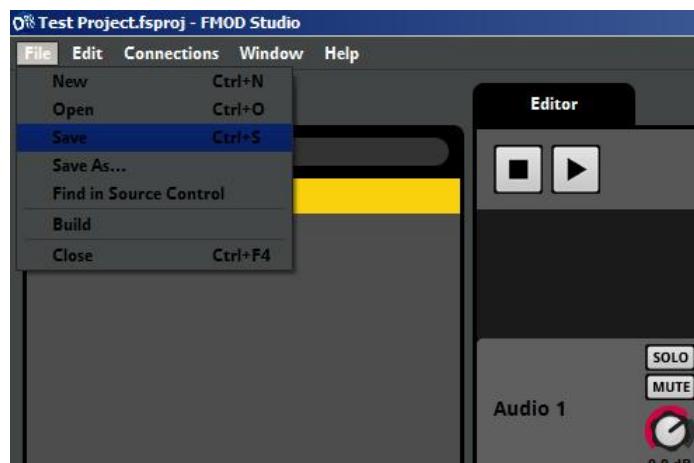
Getting Started: Saving the Project

Project Files

An FMOD Studio project consists of a folder containing a number of subfolders and files. In general, any attempt to manipulate or edit individual files and folders of a project outside of FMOD Studio is likely to result in a broken and unusable project. The easiest way to avoid problems is to treat the project folder as a single indivisible file, and avoid editing its contents outside of FMOD Studio.

Saving a project for the first time automatically creates all the required files and folders on your hard drive.

Saving a Project



Saving a Project

1. From the File Menu, select Save.

If the project has been saved before, this automatically overwrites the existing project folder. If the project has not yet been saved, FMOD Studio prompts you to specify a location and name for the project.

Note

When FMOD Studio first saves a project to a location, it creates a folder with exactly the same name used for the project, and populates that folder with all the files necessary to the project. To load the project after it has been saved, you must open the .fsproj file inside that folder.

Important

If possible, avoid renaming anything inside an FMOD Studio project folder. While an FMOD Studio project can be moved without causing any problems, renaming parts of it causes significant errors. This is because the various files and folders within a project need to be able to identify each other by name and relative path.

Basic Control Concepts

Multiple Windows

There are a number of different window types available in FMOD Studio. FMOD Studio supports having more than one such window open at once, as well as having multiple windows of certain types open simultaneously.

For more information on working with multiple windows, see the “The Window Menu” section of this manual.

*Multiple Windows*

Editor Zoom

The Tracks View can display either the Timeline or a Parameter, and either may include a considerable amount of content. For this reason, Studio allows you to zoom within the Timeline and Game Parameter views. Use Alt + mouse wheel to zoom in and out on the Timeline and Game Parameter views.

Resizing and Repositioning Modules

Modules are a core element of FMOD Studio and a significant part of any project is spent in the tracks view, positioning and aligning module trigger regions on a Timeline or Parameter. Trigger regions can be moved horizontally along an Audio Track by dragging them to the desired position. If there is more than one Audio Track in the tracks view, a trigger region can also be dragged onto other tracks.

Moving the mouse over the side edges of a trigger region changes the cursor to a double ended arrow. Clicking or dragging the edges of a trigger region allows that trigger region to be resized on the Timeline or Parameter.

Event Drag and Drop

Opening an Event Browser Windows lets you assign events to Banks and Tags by dragging the event onto the tag or bank. Refer to the manual sections on Banks and Tabs for more information on creating these objects.

Dragging objects over Browser Tabs results in the browser associated with that tab being displayed. This allows you to drag events between browsers. For instance, dragging an event from the Events browser and hovering over the Banks Browser switches the display to the Banks browser, allowing the event to be dropped onto the desired bank.

Common Properties

FMOD Studio provides a range of properties and related controllers across all of its components. Common control methods provide consistency and efficiency.

Volume

Volume is usually set by a Dial controller. Volume is measured in decibels (dB). Volume for almost all components defaults to a value of 0 dB (Zero Decibels). 0 dB does not equate to an inaudible sound, it represents no alteration.



A Volume Dial

Volume dial controllers can be adjusted by clicking on the dial and dragging the mouse up and down. The displayed value changes as the dial is turned, and displays values in whole dB numbers with two decimal places. Volume properties can range from $-\infty$ dB to +10 dB. In practical terms, this is a range from silence to 10 dB above the native input level of the signal. Hold the CTRL key while moving the mouse to fine tune the property value.

Components with Volume Controls

There are a range of project elements that include Volume Controls. These volume controls usually appear in the Deck, and can be accessed by selecting the relevant project element to make it active.

- Audio Tracks Located in the deck when an audio track is active.
- Master Tracks Located in the deck when a master track is active.
- Sound modules Located in the deck when a sound modules is active.
- Mixer Master Bus Located in the deck when the Master mixer strip is active.
- Return Buses Located in the deck when a return bus mixer strip or track is active.
- Group Buses Located in the deck when a group bus mixer strip or track is active.

Note

Every Volume Controller within a signal path contributes to the overall signal output, and the values stack. This means that if in an Event's audio track the volume is set to -6dB, and that event routes into a group bus whose volume is set to +6dB, the overall output adds the two values together to a net result of 0dB.

Pitch

Pitch is usually set by a Dial controller. Pitch is measured in semitones (st). Pitch for all components defaults to a value of 0.00 st (zero semitones). This means FMOD Studio plays back a signal from the original source sound file with no alteration to the value of the Pitch.



A Pitch Dial

Pitch Dial controllers can be adjusted by clicking on the dial and dragging the mouse. The displayed value changes as the dial is turned and displays values in whole semitone numbers with two decimal places. Most pitch properties can range from -24 s/ t to +24 st. In practical terms this is a range from two octaves below to two octaves above the native pitch of the signal. Hold the CTRL key while moving the mouse to fine tune your control.

Components with Pitch Controls

There are a range of project elements that include Pitch controls. These controls usually appear in the deck when the relevant project element is active.

- Event Master Tracks Located in the deck when a Master track is active.
- Sound modules Located in the deck when a sound module is active.

Note

Just like Volume, Pitch Controllers within a signal path contribute to the overall signal output and the values stack. This means that if a sound module's pitch is set to -6 st and the event's Master track's pitch is set to +6 st, the overall output will play at the sound file's original pitch.

Project Element Data

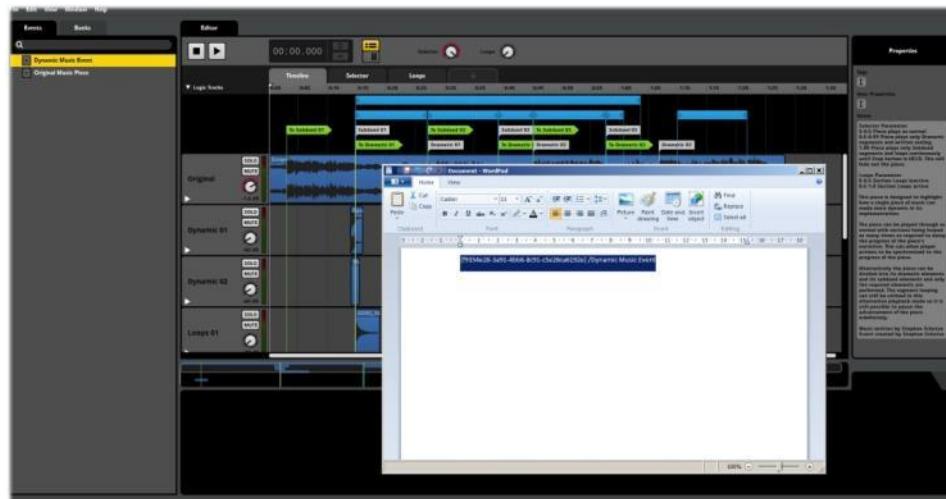
Dragging and dropping certain elements of a project into a text editor causes useful data about the dragged elements to be displayed in the text editor. The most important data that can be obtained in this manner are the GUIDs (**Globally Unique IDentifiers**) of the dragged elements. A range of elements can display their data in this way:

- Events
- Buses
- Snapshots
- Audio Assets
- Profiler Sessions

The data that is displayed varies depending on the type of object that is dropped into the text editor. It may also vary if the project has not been saved. The data for an element usually includes all of the following:

- Element Name
- Element Path
- GUID

The element name is often displayed as part of the path information. Please be aware that not all text editors support this kind of input.



Element dragged into text editor

Command Line Interface

FMOD Studio includes a command line interface that can be useful for setting up build processes to run remotely or for batch processing. The command line interface is accessed through the `fmodstudio.com` file, located in the FMOD Studio installation directory. The complete list of arguments and usage examples that are available can be accessed by running `fmodstudio.com -help`.

```
C:\Administrator:C:\Windows\system32>cmd /c fmodstudio.com
C:\Program Files (x86)\FMOD SoundSystem
C:\Program Files (x86)\FMOD SoundSystem\FMOD Studio
C:\Program Files (x86)\FMOD SoundSystem\FMOD Studio\fmodstudio.com
Usage: fmodstudio -build [-shared-audio-source-dir <directory>] [-ignore-warnings]
[-banks <banknames>] [-platforms <platformnames>] [-ignore-warnings]
[-export-guids] [-project <uproj>]
Build mode for command line tool.
shared-audio-source-dir <directory>..... Overrides the shared audio source
directory. (Optional)
banks <banknames> ..... A list of bank names to build. (Optional)
platforms <platformnames>..... A list of platforms to build. (Optional)
ignore-warnings ..... Ignores all warnings detected. (Optional)
export-guids ..... Exports asset guid and mixer guide
guides. (Optional)
help ..... See usage information for command
line tool.

Examples:
fmodstudio -build foobar.uproj
fmodstudio -build "Master Bank.FP1_Music" foobar.uproj
fmodstudio -build "banks \"Master Bank\" -platforms \"Desktop\" foobar.uproj
fmodstudio -build "platforms \"Desktop_Thru One_Platform\" \" foobar.uproj
fmodstudio -build "banks \"Master Bank\" -ignore-warnings foobar.uproj
fmodstudio -build "banks \"Master Bank_Dialog\" -export-guids foobar.uproj
fmodstudio -build -shared-audio-source-dir \"C:/audio assets\" foobar.uproj
C:\Program Files (x86)\FMOD SoundSystem\FMOD Studio>
```

Command Line Interface

The list of arguments can also be viewed by running `fmodstudio.com` with no arguments. The command line interface is intended to allow scripting options for programmers working to automate the Build process in FMOD Studio.

Modulators

Modulators are Sub Properties that provide extended functionality for various FMOD Studio Properties. Modulators allow for the set value of a Setting to be extended beyond a single specific value to either a randomized value or an envelope setting

Not all Properties within FMOD Studio can have Modulators applied to them. Those that can, have a Modulator option in the Right Click menu for the relevant Setting Controller. Volume and Pitch are the two primary Properties that include Modulator options.

Creating a Modulator

Take the following steps to create a Modulator for either Pitch or Volume Properties.

Accessing the Pitch or Volume Controls



Selecting a Track

1. Select an Event by clicking on it to make it the active Event
2. Select either an Audio Track or the Master Track to make it active

Creating a New Modulator



Adding a Modulator

1. Within the Deck Region select either the Pitch or Volume control
2. Right Click on the Control Dial to access the context-sensitive menu
3. Select Add Modulator and via the arrow select Random or AHDSR

Random Modulation

The Random Modulator provides a Value Dial that defines a percentage value. Dragging the mouse increases or decreases this value by percentage amounts. Holding CTRL while dragging allows finer control of the value changes.

The Random Modulator extends its associated Property setting beyond its single defined value by the percentage amount set in the Modulator. For Volume it would extend the single defined value set in the Volume dial and instead provide a range of volumes over which Studio randomly selects values.

As the Modulator value is altered a colored outline on the associated Controller reflects visually how the Modulator is going to affect the signal. For Volume controls the outline moves between the set value and the lowest value of -∞dB, for Pitch controls the outline expands in both directions away from the defined value towards the maximum values of -24 st and +24 st.



Random Modulation on Volume

Random Modulation Example

An Audio Track has its Volume set to -6dB. This means the sound output for that Audio Track is lowered by exactly 6dB before it continues through the signal chain. If a Modulator with the Random functionality is added to this volume controller it allows for this single value of 6dB to be extended. A Modulator set to 25% instructs Studio to randomize the value of the associated Volume Controller between 6dB and 25% of the values between 6dB and $-\infty$ dB.

Value Range “Windows”

Modulators essentially create value range “windows” for functions. Instead of a function having a set, fixed value a Modulator allows for a range of values within which Studio randomly selects a value. We define a “window” with a minimum value and a maximum value and within that window of values Studio is free to randomly select the value it applies each time the relevant Event is triggered. This is one of the principle ways in which FMOD Studio helps sound designers to avoid repetition of Event playback.

AHDSR Modulation

Attack, Hold, Sustain, Decay, Release; these are common terms used to describe a volume envelope in many audio editing setups. The ADHSR Modulation setting in FMOD Studio provides a graphical interface to create and control an AHDSR envelope for a setting in FMOD. The graphical display lets you move the nodes that relate to each of the AHDSR elements and define envelope shapes to apply to the output of a signal.

The Text display provides range values for each of the AHDSR elements as they are manipulated. In this way you can create a specific envelop shape with exact values as required. As with other Properties holding CTRL while using the mouse allows finer control of values.

AHDSR modulation can be used to define Volume and Pitch envelopes as well as other Properties within FMOD Studio. This can be very useful for getting the most out of sound file assets as combinations of Modulation controls can greatly alter the playback of a single sound file.



AHDSR Modulation on Pitch

Autopitch

The Autopitch modulator has been designed specifically to support the creation of engine sounds in game projects. Because of this it functions in a specific manner, but it can be adapted to other purposes with a little creativity.

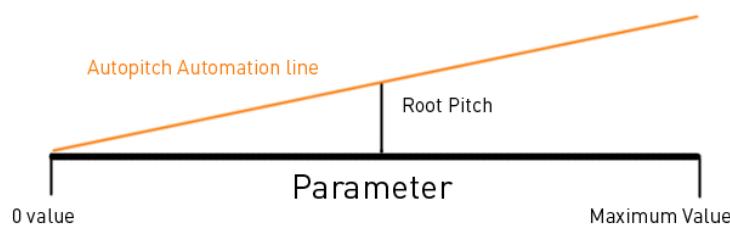


Accessing Autopitch via the Pitch Property Context Menu

When applied to a Sound Module that's triggered by a Parameter other than the Timeline, an Autopitch Modulator alters the pitch of the Module in relation to the value of that Module's controlling Game Parameter.

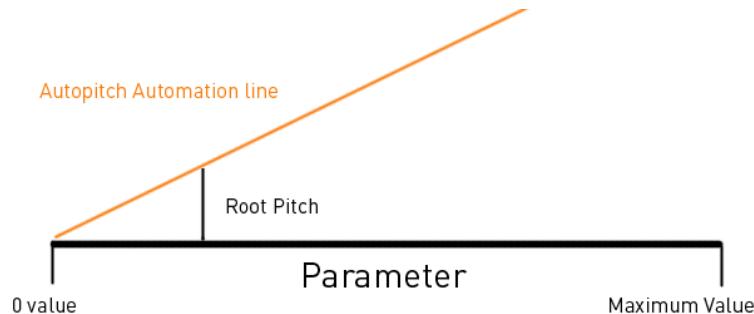
The Module for Autopitch includes a Root Pitch property that represents the point on the Parameter line where the Sound Module plays at its original pitch.

Autopitch works by defining a value of negative infinity pitch at the zero value of the Parameter. This value is used to achieve a pitch of zero hertz at the zero value of the Parameter. The Autopitch modulator references the Root Pitch and its position on the Parameter line and essentially creates a straight automation line for Pitch from $-\infty$ at 0 on the Parameter, through no pitch adjustment at the Root Pitch value, and increasing pitch further up to the maximum value of the Parameter.

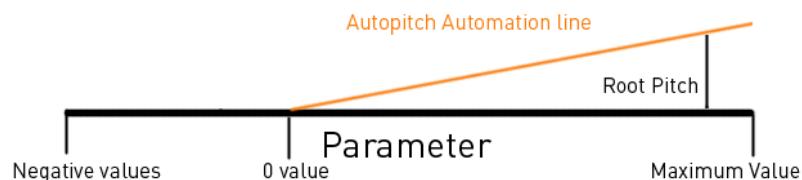


The Autopitch angle is defined by the position of the Root Pitch

As the lowest value of the Autopitch is always anchored to the zero value of the Parameter the angle of the automation line is entirely defined by the value of the Root pitch. It is also possible to create Parameters with negative values, but the Autopitch –Infinity point is always locked to the Parameter's zero value, and adjusts the pitch by –Infinity at all negative Parameter values. This is why the Root Pitch cannot display negative values.



A Root Pitch close to Parameter 0 generates a steep Pitch angle

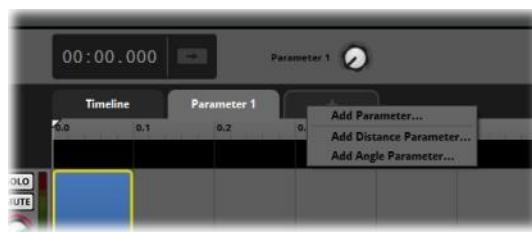


Autopitch is Anchored to Parameter value 0

Creating an Autopitch Modulator

The Autopitch Modulator can only be created under specific circumstances. The Modulator must be associated with a Sound Module that exists on a Game Parameter.

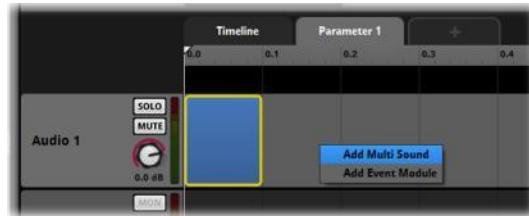
Accessing or creating a Game Parameter



Selecting and creating Parameters

1. Select an Event by clicking on it to make it the active Event.
2. Either select or create a game parameter.

Accessing or Creating a Sound Module



Selecting and creating Modules

1. Within the Audio Track select an existing sound module or create a new sound module.
2. Click on the trigger region of the target sound module.

Studio displays the deck for the selected sound module.

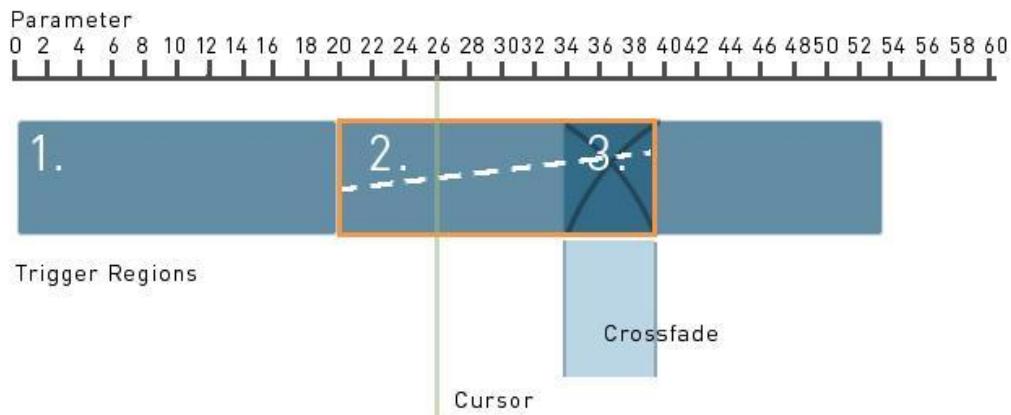
1. In the deck, right-click on the Pitch dial to access the context menu.
2. Select the Autopitch modulator.

Studio creates an Autopitch modulator in the Deck area next to the Master controls. The Autopitch can now be tuned relative to the Parameter range.



The Autopitch range matches the Game Parameter range

The above image displays the dial for the Autopitch Module. The Dial controls the Root Pitch of the Autopitch function. Root Pitch is the value at which there is no pitch alteration applied, i.e. the sound you hear is the unaltered sound of the Sound Module. You can alter the Root pitch value to “tune” your sounds, this is important if you are blending between several sounds across an Audio Track. It allows you to ensure the pitch at which one sound finishes is the same as the pitch at which the next sound begins so they blend seamlessly.



Autopitch Representation

The above diagram represents an Audio Track in FMOD Studio under a Game Parameter with a range from 0-60. The Audio Track contains three trigger regions 1, 2 and 3. The white dotted line is a visualization of the Autopitch values. Autopitch has no graphical representation within the Event Editor.

In this Diagram Trigger number 2 is highlighted, indicating it has been selected and the Root Pitch for this trigger could be defined. The Cursor is set to a value of 26 on the Parameter. The Root Pitch is also set to 26. The white dashed line reflects this because the pitch at that value is neither positive nor negative; it is at a neutral value.

The Root Pitch for both Triggers 1 and 3 are not indicated in this diagram, but they would need to be set so the perceived pitch at the beginning of Trigger two matches the perceived pitch at the end of Trigger 1. These coincide with a value of 20 on the Parameter. Trigger 2 and Trigger 3 meet with a cross fade so this makes tuning the root pitches more tricky, the pitches would need to match throughout the range of the crossfade to create a smooth blend. Matching them at 27 on the Parameter should allow for a clean blend of the two triggers.

Note

The range of values displayed on the Root pitch Dial exactly match the range of the Game Parameter in which the Sound Module has been placed. If for any reason you change the range of the Parameter, the Root pitch range changes to reflect this. An easy way to visualize this is to think “What position on the Game Parameter line does the cursor occupy when my sound has no pitch alteration?”

Side Chain Modulator

FMOD Studio includes a Sidechain Modulator that can be added to Events. This modulator allows Sidechain type control within an Event by using the signal from one Track within an Event to modulate the value of a Property in another Track of the Event. It is not possible to use a Sidechain in one event to control a Sidechain modulator in another event.



The Sidechain Modulator

Creating a Sidechain Modulator

A Sidechain Modulator can be created in the same manner as other Modulators. Simply access the context menu for any appropriate Property by right clicking on it and from the Add Modulation sub menu select Sidechain.



Creating a Sidechain Modulator

Sidechaining is most often associated with channel ducking by controlling a Compressor Effect Module, but the process of Sidechaining is essentially controlling one signal using a signal from somewhere else. The Sidechain Modulator can apply this function to any Event Property that can be modulated. The extent to which a Sidechain Modulator alters the Property is dependent on the signal it receives for the Sidechain input.

Sidechain Properties

The Sidechain Modulator has several Properties that define how it influences the Property it has been assigned to.

Attack and Release

The ‘Attack’ and ‘Release’ properties limit how rapidly a Sidechain can rise and fall, and are defined in terms of the time the Sidechained value takes for to reach its full value from its minimum, and how long it takes for it to return to its minimum from full value. These Properties function in a similar manner to the Attack and Release Properties on a Compressor Effect.

Threshold

The ‘Threshold’ ribbon fader sets the minimum and maximum values beyond which the Sidechain Modulator’s controlling signal has full or zero effect.

Amount

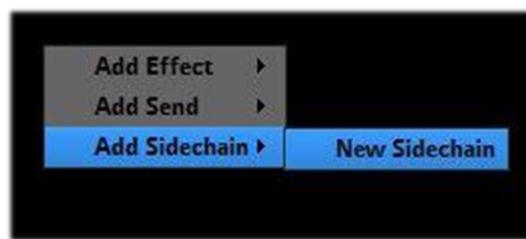
The ‘Amount’ knob sets the amount of effect the Sidechain has on the modulated Property when the Sidechain signal exceeds the Modulator’s maximum threshold. The precise units vary depending on the Property being modulated.

Note

Many of the Properties of the Sidechain Modulator are like those of a Compressor Effect Module because the manner in which they function is very similar.

A Sidechain Modulator requires a signal from elsewhere in an Event to function. A Sidechain Module can be added to the Deck region of another Audio Track within the Event to act as an input signal for the Sidechain Modulator.

To create a Sidechain Module simply right click in the Deck region of the target Audio Track and select Add Sidechain from the context menu.



Adding a Sidechain to an Audio Track

Once a Sidechain has been created somewhere within the Event you can specify it as a Sidechain Modulator's input signal. Right click on the Sidechain Modulator and from the context menu select Sidechain Input.



Assigning a Sidechain input to a Sidechain Modulator

The exact effect achieved by using a Sidechain Modulator varies considerably depending on what Property it modulates. The exact values of the Properties within the Sidechain Modulator and the Track that provides the input signal determine the end effect.

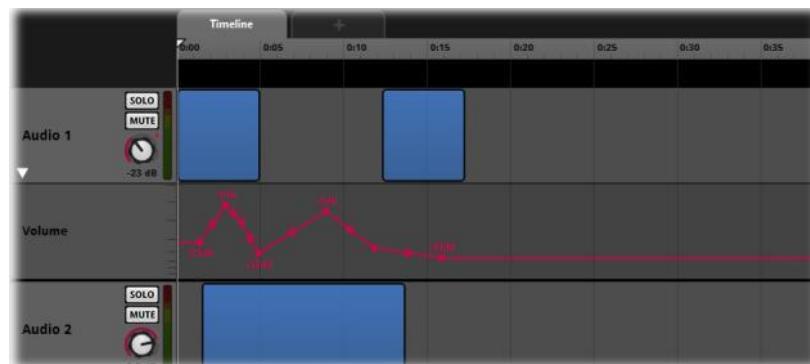
Hot Tip

Within Game project production, efficient resource management is critical to effective audio design. Modulators are an incredibly powerful tool for allowing you to maximize the potential of every single sound file that is used in a game project. Before you add a sound file to a project, consider if anything you already have in memory could be altered, combined with other sounds, blended or edited internally to Studio to achieve the sound you need. This is not only good practise in being more efficient with resources it often results in far more innovative and engaging sound design as you discover results you may have not realized were possible.

Automation

Many Properties within FMOD Studio Events can be controlled via Automation. All Event Audio Tracks and Master Tracks can have Volume, Pitch and Effects controls added as Automation Tracks.

Automation provides you with a way to define Properties that can alter throughout the course of Event playback. Once Automation has been applied to a specific setting a relevant Automation Track is added to the associated Audio or Master Track and can be edited as required.



Automation Track for Volume

Automation Tracks provide a graphic interface for you to plot out the values of the various Effects Properties over either time in the Timeline tab or relative to Game Parameter values in Parameter tabs. The Automation lines can have control nodes added to them and through manipulation of these nodes, envelope shapes can be created to achieve a wide range of dynamic effects.

Creating Automation

Automation can be added to many of the Properties within Audio Tracks and Master Tracks within Events. This section describes how to create Automation for an Event Setting.

Accessing an Event



The Event Editor

1. Select an Event by clicking on it to make it the active Event
2. Select either an Audio Track or the Master Track to make it active

Creating Automation



Adding Automation

1. Within the Deck region Right Click on the Setting Dial of an Effect module or Pitch or Volume Control
2. From the Context-sensitive menu select Add Automation

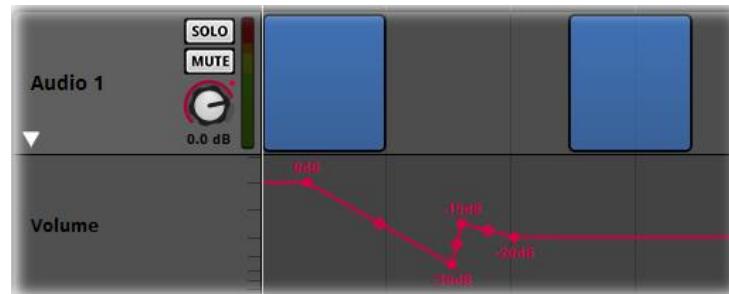
Studio adds an Automation Track to the Relevant Audio or Master Track in the Event editor. The Automation Track displays open upon creation, but can be hidden by clicking on the small white arrow on the Track Header of the associated Audio or Master Track.



Track Header and Automation Header

By default the Automation Line in the track is greyed out as it defaults to the overall value of the relevant controller. If you use the control dial to change the Properties values, the line in the automation track alters to reflect any value changes.

Clicking within the Automation Track makes it live, it changes color and creates nodes along the automation line that correspond to input clicks. This is how you create automation envelope shapes. Drag an existing node to reposition it. A text display indicates the value position of each node as it is set or manipulated.



Automation Nodes and Text Values

Between nodes there are curve markers that allow for the type of curve shape to be defined. Drag the curve marker to change the shape of the curve.



Automation Curve Types

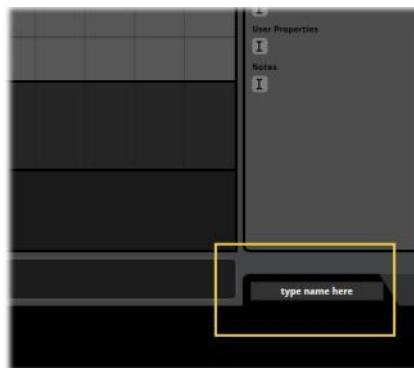
Automation Track Names

Automation Tracks adopt the name of the Sound Module or effect they are associated with, and automatically update if that name changes. This provides useful information at a glance and is particularly helpful when dealing with multiple Sound Modules and Automation Tracks.



Automation Track names match Module names

Sound Module names can be edited easily in the Deck region or event editor by double-clicking on the name and inputting a new name for the Module.



Define the Module name in the Deck area

The Assets Browser

The Assets Browser is a key part of the FMOD Studio interface. It contains every sound file added to an FMOD Studio project and is an efficient way to organize and add sounds to your project.

You can create folders or a folder structure and assign sound files to different folders in the same way the Event Browser allows you to assign Events to various folders. This is particularly useful for keeping track of potentially thousands of sound files used in larger projects.

The Assets browser also allows sound files to be added directly to Events and Modules by being dragged from the Assets browser to the desired location within Studio. Once a sound file has been added to an FMOD Studio Project it appears in the Assets browser and from there can be utilized over and over again.

Adding Sound Files to the Audio Bin

The Audio Bin is the main administrative area for Sound Files within an FMOD Studio Project. Once added to the Audio Bin sound files can be used an unlimited number of times within a Project.

There are two methods of adding sound files to the Audio Bin.

Method 1

1. Open the Audio Bin window from the Window menu, or click on the Assets tab in the Event Editor window, to display the Assets browser.
2. Open the folder on your computer that contains the sound files you wish to use in Studio
3. Drag the sound files directly onto the Assets Browser.

FMOD Studio adds the dragged sound files to the Assets Browser. You can drag and drop single sound files as well as groups of sound files. If you drag the files onto a folder in the Assets Browser, the sound files are assigned to that folder.

Method 2

1. In the File menu, select “Import Audio Files...”
2. Navigate to the location on your computer that contains the target sound files
3. Select or group select the sound files you wish to import and press Enter

FMOD Studio creates a reference for each selected file within the Assets browser. The Import Audio File method always adds the new sound files to the general root area of the Audio Bin. From there they can be reassigned to folders as required.

Audition Mode

The Assets browser includes an audition function that allows you to play sound files directly from the Assets browser and displays the wave form of the selected sound file.

Sound files can be played by either clicking the Play button or pressing the spacebar key. Pressing play again while a sound file is auditioning resets the play cursor to the beginning of the sound file. The cursor can be manually positioned by clicking on the waveform graphic to plays the sound file from that point in the file.

The audition area also includes information about the selected sound file. The sound file's name, sample rate and bit rate are displayed. During playback a timer also displays playback time.

The Assets Browser Context-sensitive Menu

There are a range of commands in the Assets browser context menu.

New Folder

This creates a new folder within the Assets browser and allows the folder to be named as required.

Move into New Folder

This creates a new folder within the Assets Browser and assigns the selected sound file or sound files to the newly created folder.

Move to Top Level

This moves the selected sound file or sound files to the root of the Assets browser.

Move to

This expands to a submenu containing all the existing folders within the Assets browser. Selecting one of these folders moves the selected sound file or sound files to the desired folder.

Refresh

This causes Studio to re-load a file so that changes made to it in an external editor are picked up. Studio has no way of detecting when you've saved your changes in an external editor so you need to tell Studio when a file has been updated.

Delete

This function deletes the currently selected sound file or sound files.

Open in Explorer/Open in Finder

This opens a Windows Explorer or Finder window with the file selected in the folder where it is located.

Open in External Sample Editor

This opens the Sample Editor defined in the Preferences window as the selected Sample editor to use with FMOD Studio and loads the selected sound file or sound files into the sample editor for editing.

Unused Sound Files

You can search for any unused sound files within the audio bin by typing in **#unused** into the search bar. This displays a list of all sound files within the project that are not currently used within any Events. This is a useful method of finding and removing unused sound files from a Project.

Creating Your First Event

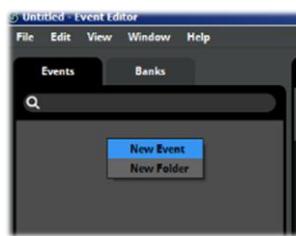
Introduction

Events are the core objects used to create and control sound output in FMOD Studio. There are a series of Module types that are added to Events to utilize various Properties depending on your needs. All Events start with an empty Event object in the Event Browser.

This section introduces Events by creating an empty Event object.

Creating a new Event

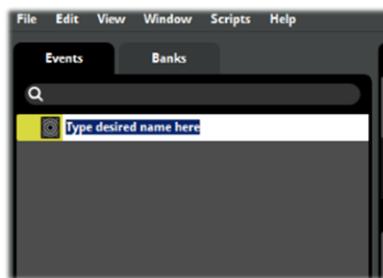
Creating Events in the Browser provides a new empty object to which sound files can be directly added. This is the first step towards creating sounds.



Creating a New Event

1. Right click anywhere in the Event Browser.
2. Select New Event from the context-sensitive menu.

Naming the New Event



Naming a New Event

1. The New Event is created with the default name *New Event*.
2. Type in the desired name for the Event you have created.
3. Press Enter or left-click away from the Event title to confirm the Event name.

Note

The context-sensitive menu contains more options if you right click on an existing Event, but you are still able to create a New Event from this menu.

Once you have created an Event in the Browser Region you can start to work with it in the Editor.

Adding Sounds to an Event

An event does not produce any sound on its own; it needs to include at least one Module Object. There are a range of Modules that can be added to Events. The simplest consists of a single sound file set to play once when triggered.

This tutorial demonstrates how to add a single sound file to an empty Event. Any WAV file would be suitable for this tutorial.

Now that there is an Event it is possible to add sound files directly into the Editor region. There are a variety of ways in which to add sounds to Studio but the simplest is to drag and drop.

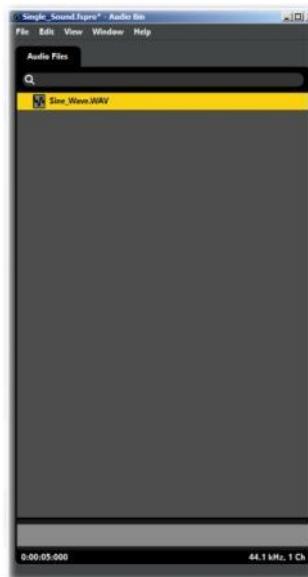
Option 1 Adding the target sound file; drag and drop



1. Open the folder on your computer with the desired sound file.
2. Drag the sound file from the folder into the Editor Region of FMOD Studio.
3. Drop the Sound File onto the Track Audio 1.

FMOD Studio creates a module in the timeline. The object occupies an area of the timeline relative to the sound files length and can be adjusted to take more or less time by drag the right or left edge of the trigger region. The module displays a silhouette of the wave in both the Timeline box and in the Deck below the Event. The Deck is also where the properties for any selected Module are displayed. In this case the Volume and Pitch properties are displayed to allow for those properties to be altered for this Module.

Option 2 Adding the target sound file; drag from Audio Bin



The Audio Bin

1. Open the Audio Bin via the Menu Bar.
2. Drag the sound file from the Audio Bin into the Editor Region of FMOD Studio.
3. Drop the Sound File onto the Track Audio 1.

FMOD Studio creates a Module object in the Event timeline. The object occupies an area of the timeline relative to the sound file's length and can be adjusted to take more or less time by click-dragging the edges of the object. The Module displays a silhouette of the wave in both the Timeline box and in the Deck below the Event. The Deck is also where the properties for any selected Module are displayed. In this case the Volume and Pitch properties are displayed for this Module.

Note

You can add a Sound File by dragging and dropping it into the Audio Bin or via the file navigation system. You can select multiple sound files and drag them into the Audio Bin as a group.

Hot Tip

It is possible to create a new Event containing a Single Sound Module by directly dragging a sound file from either the Audio Bin or your computer into the Event Browser. The newly created Event is named exactly the same as the sound file and contains a Single Sound Module that contains the dropped sound file.

If you drag the sound file into an Event Folder the Event is created inside the folder, otherwise it is created in the general browser area.

Modules

There are a variety of objects that can be added to Events within FMOD Studio. These objects are called Modules and each type of Module object has a range of Properties that allow you to achieve different results. Most Modules also include the Volume and Pitch Properties. The ability to nest Events within other Events and to link Events allows you to combine the Properties of a series of Modules to create complex playback behavior.

Sound Modules

FMOD Studio uses the term Module to describe the variety of objects that can be included in Events. Just like with hardware audio setups, modules can be inserted into the signal path in Studio to define, influence and control the ultimate audio playback that is produced by FMOD Studio.

Modules are displayed in the deck region of both the Event Editor and Mixer. There are also modules that are unique to the Event area. These are the sound producing modules.

Modules that Produce Sound

The various sound producing modules that can be added to a Sound Event function differently to the Effects modules. Most importantly, they are created initially within an Audio Track within the Event editor. When created they consist of two components that are equally important to how the module functions.

The first component is called a trigger region and is represented by a colored box that is displayed on either the Timeline or on a Game Parameter depending on where it was initially created. A trigger region defines exactly when a sound producing module produces a sound in FMOD. The region that it occupies on either the Timeline or a Game Parameter responds to the position of the cursor during playback. When the cursor intersects with the colored box it triggers playback of that sound producing Module.



Sound Module and associated trigger region

Trigger regions can be resized and repositioned in the Event Editor to define their exact playback behavior in relation to the timeline or Game Parameter values. Trigger regions can be assigned different colors and named appropriately to assist with workflow efficiency.



Colored Trigger Regions

The second component is the Module itself that is displayed in the Deck area of the relevant Audio Track. The Module interface in the Deck includes information relevant to the type of Sound Module that has been created. FMOD Studio has a number of Sound Modules, each with a unique set of properties to define its playback behavior.



Single Sound and Multi Sound Modules

Sound Modules have properties that can be modulated and automated in the same manner as Effect Modules. Their function as a sound producing object in FMOD Studio differentiates them from all other modules and means that they have both the Deck interface and the Trigger Region in the Editor.

Creating a Sound Module

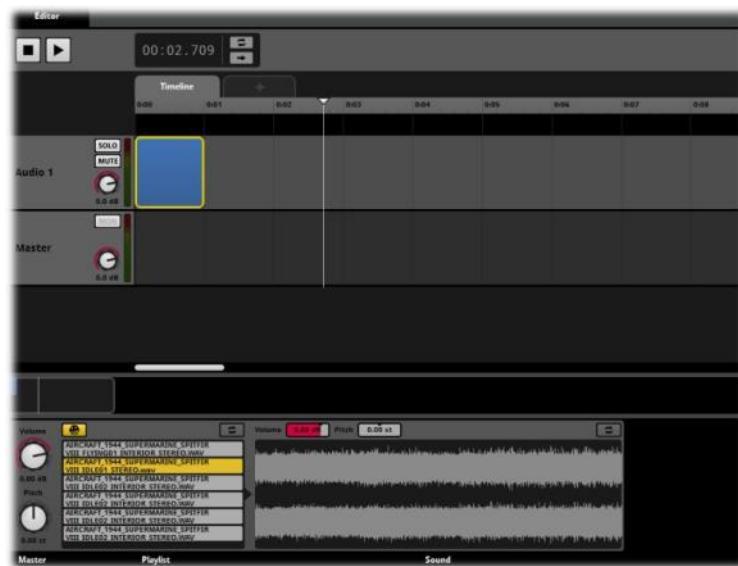


Module Menu

1. Right click anywhere within an Event Audio Track in the Editor.
2. From the context-sensitive menu select the desired Module.
3. Click to create the new Module in the Audio Track.

Multi Sound Module

A Multi Sound Module can contain more than one sound file. The properties of this object allow you to define a variety of playback behaviors.



Multi Sound Module in the Event Editor

Playlist Properties

The playlist for a Sound Module is more than just a list of sound files. Clicking on a playlist entry causes information and properties specific to that entry to be displayed, grants access to that entry's volume and pitch, and looping properties, and allows you to set its start offset behavior.

Note

A Sound File included in a Module Playlist is both a reference and an instance. Alterations made to any of its Properties in a Multi Sound Module only apply to the instance of the Sound File within that Module. Thus, altering the pitch or volume of a sound file in a Module does not affect other instances of that sound file used elsewhere within a project.

In addition to the playlist entry-specific properties, Pitch, Volume and looping controls are included in the Properties of a Multi Sound Module's Playlist.

Playlist Context Menu

Each entry in a playlist can be modified via a context menu. Right clicking on a Playlist entry displays a menu of additional options.



Playlist Entry Context Menu

Open in External Sample Editor

This opens the Sound File in an external program so it can be edited as required. If you haven't specified a sample editor in the 'Assets' tab of the 'Preferences' window, FMOD Studio uses your operating system's file type association to open the file.

Note

Editing a playlist entry in an external editor changes the associated sound file in the project's Audio Bin, or in its original location if the file is in your project's shared audio source directory. This permanently changes this asset and all instances of it so use this feature carefully.

Rename

This allows the Playlist entry to be renamed as required via direct text input. You can also press the F2 key to rename the selected item.

This function is very useful for renaming Event Sounds so that they are not all labelled "Event Sound" but it can be used on any playlist entry.

Note

Renaming an item in the Playlist does not rename its associated sound asset, so this must be done with care. Extensive renaming across a project could result in the project being hard to understand.

Set Play Percentage

You can specify the percentage chance of a playlist entry being played when the play mode is set to random. The total set play percentage for all play list entries cannot exceed 99%, and any entries that do not have a play percentage value defined are automatically assigned an equal share of whatever percentage remains.

For example, in a Module with three Playlist entries, each would automatically be balanced to 33.33% each. If one was manually set to 50% the remaining 50% would be divided equally among the two remaining entries making them 25% each. If two more sounds were added to the Playlist without altering the set percentage the four undefined entries would automatically be set to 12.5% each.

Remove Play Percentage

This context menu item is only displayed once a play percentage value has been added to a playlist entry. It removes the displayed play percentage and rebalances the percentages of all playlist entries without displayed percentages.

Add Event Sound

This adds an Event Sound Module to the playlist of this Module.

Remove

This removes the selected entry from the playlist.

Plug-in sound modules

FMOD Studio supports a number of third-party plug-in sound modules.

Trial versions of some third party sound modules are included in FMOD Studio. You can develop content that uses trial version plug-ins, but you will not be able to use such content in your game without first purchasing a licence and adding the appropriate libraries in your game's code. To obtain the licences and libraries your project requires, contact the plug-ins' developers.

In most cases, content created using the trial version of a plug-in will work with the full version. If you purchase a licence after already creating content, you will not have to create the content again using the full version.

AudioGaming AudioMotors

The properties of the plug-in sound module that appear in FMOD Studio correspond to the controls found in AudioGaming's tool. For more information about these controls, and about this plug-in, see the AudioGaming AudioMotors V2 user guide available from <http://store.audiogaming.net/content/audiomotors-v2-pro>.

This plug-in makes use of car data produced by AudioGaming's AudioMotors V2 pro. This tool can be purchased at <http://store.audiogaming.net/content/audiomotors-v2-pro>, and a trial version is available from <http://store.audiogaming.net/content/audiomotors-v2-pro-trial>. The trial version cannot produce the data required by the AudioGaming AudioMotors sound module.

Bulk Editing

FMOD Studio supports Bulk Editing within the Event Editor. This allows Properties common to the selected Events to be adjusted simultaneously for all selected Events. This can be extremely useful when working with large projects than contain lots of Events.

Bulk Editing is automatically made available whenever multiple Events are selected at the same time. The Event Editor's display changes to indicate which properties can be modified as part of the Bulk Editing function.

Hold CTRL while clicking on the desired Events to select multiple Events within the Event Browser. When multiple Events are selected, their bulk editable properties are displayed in yellow.



Bulk Editing in the Event Editor

Only objects that are eligible to be bulk edited are highlighted in yellow, anything that is not highlighted only changes the active Event if edited. Automation lines that can be edited in Bulk Edit mode are indicated in white to make them easier to see.

Note

Only Properties that exist in all selected Events are highlighted for Bulk Editing. For example, if you select three Events and one of the Events does not have a 3D Panner, the 3D Panner is not highlighted for Editing. This is because it is not possible for FMOD Studio to identify the intended result for all selected events when non-identical properties are edited. Also, if a property you expected to be able to bulk edit is instead not highlighted, it may be that you have accidentally added an event to the selection that you did not intend to edit.

Editing Properties

By default, Bulk Editing allows you to change almost any common Property values in the selected Events. Altering a Property value while bulk editing results in that Property changing relatively in all selected Events; If an altered property's value would exceed the range of allowed values for that property, it is instead set to the closest allowed value. Where possible, FMOD Studio also displays Property values that do not exactly match when multiple Events are selected for bulk editing.



Volume Property Dial Bulk Edit display

The above image illustrates the Volume Property dial for Audio Track 1 of an Event. The darker line on the dial is the volume value for the displayed audio track, the lighter line is the volume value for a second Event that has been selected as part of this Bulk Edit process. If the Property value is changed this display indicates the change to both of the selected Events via the dark line and light line. Alterations to values is relative, so the Property dial indicates the value change of the displayed Property and the lighter lines indicates the relative changes to any other selected Event properties.



Bulk Editing Three Events

This diagram illustrates a change of -16dB to the displayed Audio Track and two lighter lines on the Property dial indicate the relative property values of two other Events selected for Bulk Editing. In the example of Volume Properties, keep in mind that the Property Dial for Volume is not a linear change when the dial is rotated, so the relationship between the displayed lighter and darker lines on the dial shifts as you move through the negative dB value ranges.

Alternatively, holding down the 'Shift' key while bulk editing the values of dials causes those values to be set to the same absolute value as the dial in the active event, rather than receiving the same amount of relative change.

Range Extremes

FMOD Studio attempts to keep a relative relationship between all selected Property values. So you would see the following results if you altered the Volume of three bulk selected Events.

Alterations of Volume Property Values by -3dB

Original Value	New Value
0dB	-3dB
+3dB	0dB
-12dB	-15dB

All Properties have a maximum and minimum value that cannot be exceeded. Once a Property Value reaches an extreme value it stays at that extreme value, but FMOD maintains the relative relationship until that specific edit attempt is completed. So if increasing the value of Event number 2 forces Event number 1 to an extreme, Event 1 stays at that extreme value. If you then reduce the values of the combined Event Properties down without having released the mouse button, then Event 1 maintains its value relative to all other selected Event Properties.

Alterations of Volume Property values by +4dB

Original Value	New Value
0dB	+4dB
+7dB	+10dB (+10dB is the maximum value and cannot be exceeded)
-12dB	-8dB

Controlling Property

When Bulk Editing, the Event that is selected last is usually the active event, and therefore the Event whose Properties you use to control all selected Events. So by adjusting a Property value of the controlling Property you adjust all selected Event Properties. This choice can influence your range of control. Remember that a Property cannot be adjusted outside of its range extremes, so you can only turn a Property dial so far before it stops. If the controlling dial stops then obviously all other selected Property values stop as well, so make sure you select the most appropriate Event out of a group to be bulk edited to allow you to achieve the desired results. To alter the active event in an existing selection, press the 'Tab' key to cycle through all selected events.

While adjusting property values, the Shift key assigns all selected properties to the same value. The assigned value is defined by the active event's property, so whatever value is described by the active event's property when the shift key is held is the value all selected events' properties snap to.

Bulk Editing Parameters

Parameters can only be Bulk Edited when the selected Events have Parameters of the same name, the same type and the same value ranges. Parameters that can be successfully Bulk Edited are highlighted when multiple Events are selected.

This requirement is to prevent undesirable editing. If two Parameters have the same name but not the same value ranges bulk editing could result in undesirable results and significantly affect the playback behavior of your project.

Bulk Editing the 3D Previewer

The 3D Preview tool can also be used in Bulk Edit mode. As with other Bulk Edit functions simply select all the Events you wish to edit simultaneously. As with all Bulk Edit examples Properties can only be varied within their maximum extremes and so any change that would take them beyond those extremes stops at the extreme value range.

Module Playback Behavior

There are two toggle buttons within a Multi Sound Module that define the playback behavior of the Module.

The first has a icon of a die, and defines whether the entries within the Module's playlist are selected at random or played sequentially.



Multi Sound Module Random/Sequential Toggle Button

By default a playlist is set to random, and randomly selects from its list of entries while avoiding repeating the same entry too often. When this button is not highlighted, Studio selects the entries in the order in which they appear in the playlist.

To avoid over-repetition, if there is no playlist weighting, FMOD Studio handles the selection as if on 'shuffle' mode: The playlist is shuffled into a random order and played through once, then reshuffled and played through again, and again, and so on. So the maximum number of times an entry plays in a row is twice, and the frequency with which any given entry plays is always exactly $1/n$, where n is the total length of all entries in the playlist.

If there is playlist weighting, then it's effectively truly random; the chance of an entry directly following itself is equal to the probability of that entry being selected.

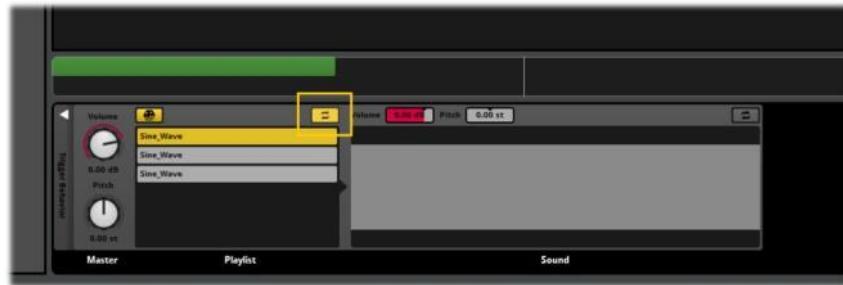
Note

Both Random and Sequential playback behaviours make global calculations. This means that FMOD Studio tracks which playlist entries have been played across all instances of the Module in the project. So if a Module has the entries 1, 2 and 3 and is set to sequential playback order, it selects them in the order 1, 2, 3. If that Module is stopped and then started again it plays whatever sound file was next in the sequential order, and does not reset back to 1 every time it is triggered. Similarly, if multiple instances of the Module are triggered, the next item to be selected is based on the most recently triggered entries across all instances, not just the one that's currently playing.

Module Looping Behavior

The looping toggle button specifies whether Studio applies looping playback to the module. In a Module with a playlist, there are separate looping toggle buttons for the module as a whole and the individual entries within its playlist.

A Single Sound Module set to loop loops the playback of its sound file.



Module Loop Mode Button

The Scatterer Sound Module is a more complex Module and its behavior when set to loop is more varied. The best way to think about the Scatterer Sound Module is to think of it as having two components. The right side of the Module is essentially the same as a Multi Sound Module and it behaves in the same way. It triggers sound files from its playlist, and if set to loop, strings those sound files together to loop the Module.

The left side of the Module provides the Scatterer functions, which are applied to the integrated Multi Sound Module, not to the individual triggered playlist entries. When the Scatterer Sound Module is triggered, each sound scattered is a separate instance of the integrated Multi Sound Module. This means that when the integrated Multi Sound Module is set to loop, the Scatterer applies all its functions to the looping Multi Sound Module. So if Polyphony is set to 3 the Scatterer allows up to 3 continuously looping sounds to be triggered depending on all the Properties, and how they have been defined.

Timelocked Sound Modules cannot loop, so any Module set to loop automatically is not Timelocked. More information on Timelocked sounds can be found in the relevant section.

Looping a Module's individual Playlist Entries

Multi Sound Modules and Scatterer Sound Modules allow access to the Properties of the individual entries contained within their Playlists. Each playlist entry can be individually set to loop. Within a multi entry playlist this function can produce various types of playback behavior.

Module set to single playback, all entries set to loop

The Sound Module triggers a sound file either sequentially or randomly depending on its playback behavior settings. The selected entry loops indefinitely until the Module is stopped.

Sound Module set to single playback, some entries set to loop

In this setup the Sound Module triggers a sound file depending on its playback behavior mode. If it selects a non-looping entry that sound plays and then ends, if it selects a looping entry that sound plays and loops until the Module is stopped.

Sound Module set to looping playback, all entries set to not loop

The Sound Module loops by stringing the entries together end to end to achieve endless playback until the Module is stopped.

Sound Module set to loop, some entries set to loop

The Sound Module triggers and loops by stringing the entries together until it triggers a playlist entry that is set to loop. The Module then ‘jams’ on that entry and loops it until the Module is stopped.

Sound Module set to loop with all playlist entries set to loop

The Sound Module triggers and selects a playlist entry which then loops until the Module is stopped.

Single vs. Multi Sound Modules

The choice between using a Single or Multi Sound Module is one of the most basic examples of extended sound design. Traditional methods of sound design would select an appropriate sound effect for an Event and trigger that sound when required. In linear media this method is completely appropriate. In a game project, the target sound may be triggered over and over during the course of a game, so for instance using only a single sound file to represent an Event can result in undesirably repetitive playback.

By creating a Multi Sound Module and populating it with multiple variations of the same sound type, Studio can be made to randomize which sound file is selected for playback each time the Event is triggered. This instantly provides a wider selection of sounds and reduces repetition.

Event Module

An Event Module object is an object designed to contain other Events instead of directly referencing sound files. The Module created in the Editor Region includes Volume and Pitch controls but does not display any sound file data or playlist information. Double-clicking on an Event Module object opens a new window containing a blank Event. This Event can then be populated like any other Event.



Event Module in the Event Editor

Event Reference Modules

FMOD Studio allows Existing Events to be referenced from within other Events and triggered in the same way Sound Modules are triggered. Event Reference Modules include a Trigger Region, positioned on either the Timeline or Parameter, and a limited set of Module Properties in the Deck area.



Event Reference Module

An Event Reference is triggered by cursor position just as any other Sound Module would and can have Automation and Modulation added to its Properties. Event Reference Modules are not an instance in the same way that other Modules are. An Event Reference Module always exactly reflects the original Event it was created from and any changes to the original Event are applied to all Event Reference Modules associated with the original Event.

Creating Event Reference Modules

An Event Reference Module is created by a different process than other Sound Modules. The first, most important, step is to create the Event you want to reference. This Event does not need to be completely finalized as all changes update automatically across all its associated references.

Because this process involves working with more than one Event at a time a second Event Editor Window should be opened from the Menu Bar. Go to Window/Event Editor (or alternatively use the shortcut keys CTRL + 1). Once you have two Event Editors open, select the Event you want to reference in one Editor and the Event you want to add the reference to in the other.

To create an Event Reference Module, simply drag the Event from the Event Browser in one editor window into the Audio Track of the target Event in another window. Studio creates a new Trigger Region in the target location and from that point you can move and resize the Trigger Region as usual.



Creating an Event Reference Module

Double clicking on the Trigger Region of an Event Reference Module opens that event in the Editor in the same manner as accessing an Event Sound Module. It indicates the location within the event the same as nested events as well. The breadcrumb display keeps track of the location within the nested Event structure.

Note

Event Reference Modules are routed through the Event in which they are created. This means they enter the Mixer via that Event. This means that any routing or processing of the original referenced Event in the Mixer is not reflected within the associated Reference Events.

Copying an Event Module

You can paste a copied event sound module into the folders browser as an Event.

Scatterer Sound Modules

A Scatterer Sound Module includes some complex functionality that allows a wide range of different sounds. It is also a basis for a lot of Generative Audio type Events. There is of course no requirement to utilize all of the functions all of the time. A Scatterer allows for continuous spawning of sound files as well as positional information for the sound Event in 3D space.



Scatterer Sound Module

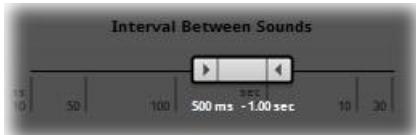
A Scatterer Sound Module has four main function sections

1. Trigger behavior
2. 3D Positioning behavior
3. Volume behavior
4. Pitch behavior

The Volume and Pitch behaviors are similar to many other functions within FMOD Studio except that where most other Pitch and Volume properties can have Randomization Modulators added to them in a Scatterer Sound Module the Randomization controls are always displayed.

Trigger Behavior

A Scatterer Sound Module allows an Event to repeatedly trigger the sound files added to the Module. The upper ribbon slider is titled “Interval Between Sounds”. The value applied to this property instructs the Module to count a defined period of time before triggering another Sound File from the modules Playlist.



Interval Between Sounds

Two values are defined in the Trigger Interval property. These represent the minimum length of time FMOD must wait before triggering a sound and the maximum period of time. If both of these values are set to 1000ms then FMOD triggers a sound exactly every 1000 ms or 1 second apart. Using these values FMOD sounds like a clock keeping perfect time. By setting the two values the same FMOD triggers on exact time values.

If the two values are set differently the result is that FMOD is given a window in time in which to randomly select a value. For example the default values of 500ms-1 sec instructs FMOD to count at least 500ms and no more than 1 second, but within those two values FMOD is free to randomly select a time to trigger the sound. Defining a range window such as this, results in FMOD randomly selecting a different value each time it triggers a sound. This is useful for producing untimed behavior as often appears in nature.

Related to this property is the Polyphony dial. When FMOD is instructed to trigger sounds over and over it has no concept of the length of the audio tracks it is triggering. Even though the default values may trigger a new sound between 500ms and 1 second it is highly likely that many sounds are longer in length than 500 milliseconds.

This means Studio must be allowed to play more than one sound at a time. If Polyphony is set to 1 and a new sound is triggered while the current sound is still playing, FMOD is limited to only playing one sound at a time. This means the new sound waits in queue until the current sound finishes and then plays. Increasing the Polyphony to 2 would allow both sounds to play. The default value of 16 is designed to allow multiple sounds to play concurrently without utilizing too many channels. Polyphony values should be defined to suit both the creative requirements of the sound as well as the technical limitations of the target platform.

The Total Sounds dial allows you to define a finite number of triggered sounds before FMOD stops playing the Event. By default this property is set to infinite, so once triggered an Event plays continuously until it is stopped. If the Total Sounds was set to a value of 5, Studio would trigger the sound with the appropriate interval defined by the Interval Between Sounds property, but would do so only 5 times before it automatically stopped playing.



The bottom ribbon is used to define minimum and maximum Scatter Distance. As with the Interval property this property can be set to define an exact number or a range window in which FMOD randomly generates a value. This property adds control of the 3D positioning of a Sound Event. By defining a minimum and maximum value you can define an area in which Studio randomly positions each triggered sound within 3D space.

So a value of 0 to 10 would create a circular area with a diameter of 10 in which FMOD would randomly place each sound triggered. Think about a group of birds singing in a tree and how each sound is triggered within the canopy of the tree area. As with Interval Time, defining an exact value means that the sounds remain positioned randomly around the emitter point in the 3D world, but at a constant distance defined by the Property. Setting this Property to 0-0 removes any random 3D position behavior.



Min and Max Scatter Distance

Note

If you delete the 3D Panner from the Event Master Track and a Scatterer Sound Module is using the 3D min and Max values, FMOD randomizes the position of the sound within a 2D stereo panning spread. This can be useful for creating dynamic stereo panning.

A word on Generative Audio

The Scatterer Sound Module is a key tool in creating Generative Audio, but what is Generative Audio? Generative audio is a method of sound design that allows you to create dynamic sound effects; that is sounds that are generated in real time. The concept is centered on using sound files as building blocks rather than as finished sound effects. A good example of this is creating an ambient effect of birds by using short individual bird calls. Using the Scatterer Sound Module a sound designer can instruct FMOD Studio to trigger a stream of bird sounds that overlap and spread across a defined area in 3D space. This can create the illusion of a group of birds within the area of a tree singing away. This is just one example, but there are many different methods of producing dynamic sound Events that can significantly improve the overall sound environment of a game project.

Scatterer Sound Modules in 3D Events

3D events are events that behave differently based on their location relative to the listener in 3D space. Any event that includes a 3D panner effect module or any of the position-based built-in parameters is therefore a 3D event.

An event is also 3D if it contains a scatterer sound module that's 3D. The playlist entries of a scatterer sound module can include 3D events, and placing a single sound module in a scatterer sound module's playlist automatically gives that single sound module a hidden 3D Panner effect; In either case, such playlist entries are known as '3D playlist entries.' If a scatterer sound module contains at least one 3D playlist entry, that scatterer sound module is 3D.

Every time a scatterer sound module spawns an instance of one of its playlist entries, it feeds that playlist entry positional information generated from the scatterer sound module. This is used by the playlist entry's 3D Panner effect module (if present) and its position-based built-in parameters (if present). This is what allows a scatterer sound module to alter the apparent positions of the sounds it scatters.

Note that, if a scatterer sound module has 3D playlist entries, and the event that contains the scatterer sound module features a 3D panner, the sounds scattered by the sound module will be affected by both panners. As this results in the same sound being panned (and thus attenuated) twice due to distance, we recommend against using scatterer sound modules with 3D playlist entries inside events with other 3D behavior.

Also note that the hidden 3D Panner automatically applied to single sound modules in a scatterer sound module's playlist has a Min Distance of 0 and a Max Distance of 20, regardless of the Min & Max Distance properties of other effects in the event.

Audio Weather Module

The Audio Weather Module object is a plugin object for FMOD Studio that provides a series of properties that combine to produce procedurally generated weather sound effects. Wind and water effects can be generated and controlled via a set of simple controls.

Single Sound Module

A Single Sound Module is what is created when a sound file is dragged directly into an Audio Track. The functionality is essentially the same as a Multi Sound Module except the object only references a single sound file. This is slightly more efficient from a code perspective than adding only one sound file to a Multi Sound Module and so is recommended when only one sound file is required.

Events within Events

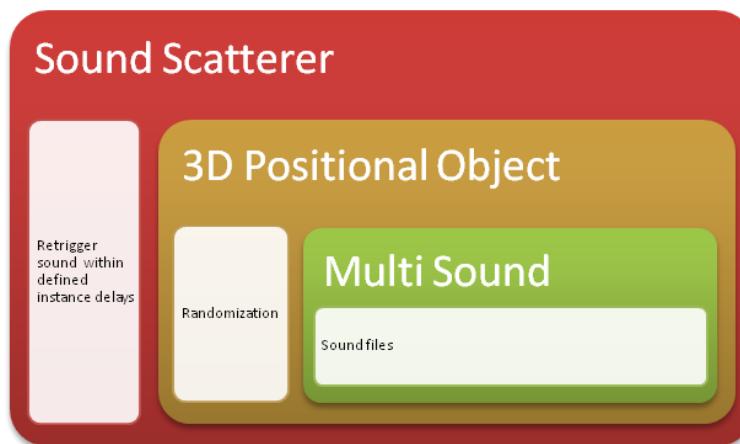
One of the most powerful functions of FMOD Studio is the ability to nest an Event within an Event virtually endlessly. This allows for a Module with one type of function to be nested within a Module with a separate function. In this manner it is possible to combine the entire range of Module functions to achieve a vast range of output results.



Nested Events

The above graphic illustrates Events nested to three levels. Each new level can be opened in a separate window for ease of workflow and editing. Combining different types of Modules in this way is the most efficient way of creating new types of functionality.

The base level of the chain could be a Multi Sound Module that selects randomly from a series of sound files. This object is then nested into a 3D Positional Module that allows for randomization of the sound source within a defined area on playback. Next, it could be nested within a Scatterer Module that constantly retriggers the sound with a defined window of delay between each instance being played. This process can be added to continually to add more functionality and control to the final output sound.



Nested Module objects combined to produce a complex result

Parameters in Nested Events

If an event contains a Parameter, and a nested event or event reference in that event contains a Parameter with the same name, the Parameters of the child events automatically update to have the same value, as a proportion of their range, as the similarly-named Parameters in their parent.

This even works for event sounds inside the playlists of multi sound modules.

Event Instances

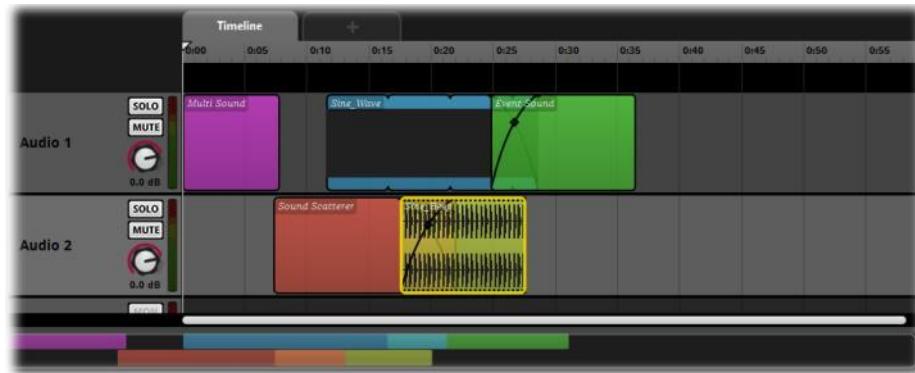
The ability to cross-reference Events or nest them within other Events is very powerful. A single event can be referenced in more than one place in an event, or even in multiple events. Each of these nested events creates an independent event instance when triggered, and each of these instances has its own parameter and timeline values.

Double-clicking on an Event shows the Properties and behavior of that event as a unique Instance in the Event window.

You can change its behavior in this window while still controlling it from the Parent Event without breaking the process. Effectively the Parent Event is controlling an exact copy of the nested Event within the Parent Event, while the window you have open is a duplicate instance. Upon playback, both versions play together and apply their individual Properties and controllers.

Trigger Regions

Trigger Regions represent Sound Modules in either the Timeline or Parameters and are a primary aspect of creating and controlling Sound Events within FMOD Studio. There are a series of functions that apply to working with Trigger Regions that need to be understood to get the most out of working in FMOD Studio.



Trigger Regions

Trigger Region Names

All trigger regions can be renamed by double-clicking on their existing name and typing a new one.

FMOD Studio automatically generates default names for newly created trigger regions. The exact name depends on the Trigger Region type, as shown in the following table.

Trigger Region Type	Default Name
Single Sound Module	The name of the sound file used to create the module.
Multi Sound Module	“Multi Sound”
Event Sound Module	“Event Sound”
Scatterer Sound Module	“Scatterer Sound”

Pasted sound module of any type	The name of the copied trigger region or event.
Event References	The name of the referenced event.

In all instances the Trigger Region's name is displayed on both the Trigger Region colored box in the Event Editor and the Deck Region for the Sound Module when it is selected in the Editor.



Trigger Region Naming

Renaming Trigger Regions

You can easily rename any type of Trigger Region by double clicking on the name in either the Trigger Region box or in the Deck area for the Sound Module. There is no limit on the number of characters that can be used to name a Trigger Region; however there is a limit on the number of characters that can be displayed, so it is best to keep names fairly short.

Note

How a name displays on a Trigger Region depends on the size of the Region Box on the Timeline or Parameter. In the Deck Area there is enough space for a name of approximately 25 characters including spaces, however the amount of space on the Region box depends on how big the Trigger Region is on the Timeline or Parameter. Smaller Trigger Region Boxes can only display a few characters. If it is important for your project to be able to clearly distinguish between the Trigger Regions on a Timeline or Parameter then factor this in when you decide how to name your Sound Modules.

Trigger Region Colors

Trigger Regions default to a light blue color, this color can be changed as required. Using a range of colors for complex Sound Events can make editing and implementing much easier.

To redefine the color of a Trigger Region simply Right Click on the Trigger Region to access the Context Sensitive menu for that Region and select Set Color. An additional menu provides a list of available colors that can be applied to your Trigger Region.

Note that the copy / paste function copies a Trigger Region exactly so the pasted version has the same name, color and any Properties of the original.

*Trigger Region Color Options*

Trigger Region Instances

A Trigger Region and its associated Sound Module can be copied and pasted as required within an Event or across any number of Events within a project. Right Click at the desired location and select Paste to place an exact duplicate of the copied Trigger Region and its Sound Module. This duplicate is an independent instance of the original copied object. This means that it can then be altered or redefined in any way without affecting its parent object or any subsequent child copies of itself. This is a quick way to use a Sound Module to rapidly populate a project with copies that can then all be edited as required.

Precise positioning for Trigger Regions

For basic positioning of Trigger Regions you can simply drag it to its desired position. More advanced methods for positioning Trigger Regions can offer more exact positioning solutions. These methods are accessed via the Trigger Region context menu that is displayed when you right click on a Trigger Region

Move To...

The Move To... command accesses a pop-up window that allows the position of a Trigger Region to be precisely defined. By default the length of the Trigger Region is locked to maintain the current size of the Trigger Region. This means defining Start and End values that exceed the locked length of a Trigger Region is not possible. Studio accepts the last inputted value and redefines the other value to match the locked length. If the Length value is unlocked then its value is redefined to appropriately reflect the Start and End values.

This function allows you to define the exact size and start and end points of a Trigger Region. This is best used when the Trigger Region position is a critical aspect of Event creation and needs to be precise.

Move to Cursor

The Move to Cursor command is a quick way to reposition a Trigger Region to the location of the cursor. This can be particularly useful for working with Multi Track Events as this command repositions a Region horizontally along the Timeline or Parameter while maintaining its position on the relevant Audio Track. Select this option to move the Trigger Region to align its start point with the same value on the Timeline or Parameter as the cursor.

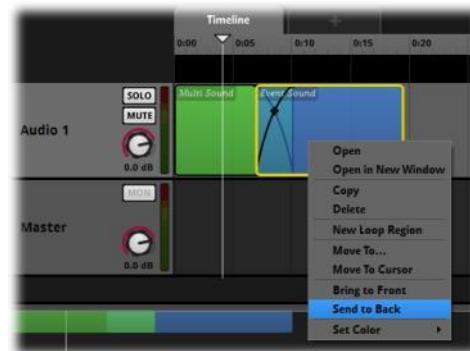
Bring to Front

You can position Trigger Regions on an Audio Track so that more than one Trigger Region occupies the same area. By default, FMOD Studio creates cross fade regions when Trigger Regions overlap,

Studio also displays the last selected Trigger Region in front. The Bring to Front command repositions a Trigger Region that shares an overlap with another Trigger Region to the front position.

Send to Back

The Send to Back command is the opposite of the Bring to Front command. It repositions a Trigger Region that shares an overlap with another Trigger Region to the back position.



Send to Back Function

The Logic Track

Located just below the Timeline in the Event Editor, the Logic Track's black strip provides specific and detailed control of how the timeline cursor moves during playback of Events.

You can change whether the Logic Track is displayed or hidden by clicking on the Logic Track disclosure triangle. This allows you to maximize screen space when working with multiple audio tracks or large amounts of Logic content.

Loop Regions

A Loop Region defines two points on the Timeline. If the Cursor reaches the right end of the Loop Region it immediately snaps back to the left end of the region and continues along the Timeline from there. This effectively ‘traps’ the Cursor into a looping cycle within the boundaries of the Loop Region.

Creating a Loop Region on the Logic Track



Creating a Loop Region

Take the following steps to create a Loop Region:

1. Select an Event to make it active
2. Right click on the Logic Track to display the context menu (this menu is only available on the Timeline Parameter).
3. Select Add Loop Region from the menu.

The two edge markers of the Loop Region define the beginning and end of the loop. When this Event is auditioned, the Timeline Cursor advances along the Timeline and moves past the start of the Loop Region as normal. Once the Cursor reaches the Loop Region end it snaps back to the Loop Region start and continues to move through the Loop Region. The playback continues indefinitely until the Event is stopped.

A Module does not need to be set to looping for a Loop Region to function; Studio simply loops cursor movement within the defined region. How effective a Loop Region sounds depends on the type and content of the Module and the placement of the Loop Region's start and end points.

Creating a Loop Region on a Trigger Region

Take the following steps to create a Loop Region on a Trigger region:

1. Select an Event to make it active
2. Right click on a Sound Module's Trigger Region to display the context menu.
3. Select Add Loop Region from the menu.

Creating a Loop Region from the context menu of a Trigger Region instructs FMOD to create a Loop Region that exactly fits the target Trigger Region. Technically the two loop edges are just outside the boundaries of the Trigger Region. They need to be in this position to correctly trigger the Trigger Region.

Sustain Points

Sustain Points are a type of Marker that can be placed on the Logic Track of the Timeline. The function of a Sustain Point is very simple:

When the Timeline cursor makes contact with a Sustain Point it stops advancing forwards (this does not necessarily pause all audio content of the event). There is no minimum or maximum time period that a Timeline can be paused for.

To deactivate the Sustain Point and allow the Timeline cursor to continue moving forward FMOD must receive a Keyoff command. This command must be sent in code. This means the behavior needs to be defined in game code by a programmer on your team. The Keyoff command can be simulated for audition purposes in FMOD Studio using the Cues button, marked A; Pressing this button causes the timeline cursor to ignore the next sustain point it comes across, or to cease being affected by the sustain point it is currently stopped at.



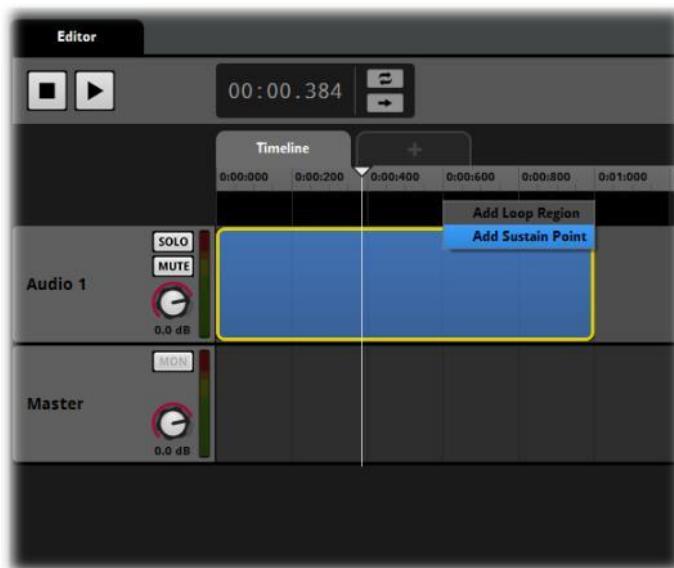
Sustain Points and the Cues button

There is no limit to the number of Sustain Points that can be added to an Event, but regardless of how many Sustain Points are created there is only ever a single Cues button that controls all Sustain Points in an Event.

Creating a Sustain Point

Take the following steps to create a sustain point:

1. Select an Event to make it active
2. Right click on the Logic Track to display the context-sensitive menu.
3. Select Add Sustain Point from the menu.



Creating a Sustain Point

Use the mouse to drag Sustain Points along the Logic Track. Delete a Sustain Point by right clicking on the handle at the top of the Sustain Point to open the context menu and selecting Delete.

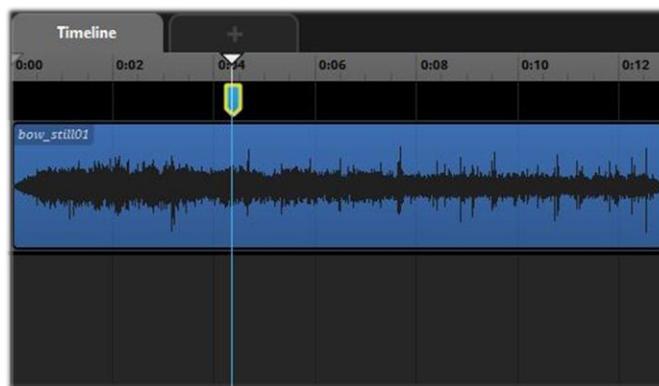
Working with Sustain Points

The way a Sustain Point affects the playback behavior of an Event depends on how it is placed. If a Sustain Point is placed where it does not intersect with any Sound Module Trigger Regions, then it pauses the advancement of the cursor in the same position. This effectively pauses the Timeline playback on a silent region of the Timeline. This can be used to silence an Event without completely stopping its playback if you need to, provided there are no Sound Modules on the event's other Parameters.

If a Sustain Point is positioned so that it intersects with a Sound Module Trigger Region it also pauses the advancement of the Timeline cursor, but the exact effect this has on playback depends on the type of Sound Module.

Timelocked Sound Modules

A Timelocked Sound Module behaves in the same manner as a linear editor in relation to cursor movement. The cursor 'reads' the sound-file and tracks its progress visibly across the displayed waveform, so if the cursor movement is halted by a sustain point, playback is also halted at the exact position of the cursor.



A Sustain Point on a Timelocked Sound

Once the Keyoff command is given the Timeline Cursor continues tracking time and the Timelocked sound continues play from the exact position of the Sustain Point. It is of course possible to add multiple Sustain Points across just a single sound so that playback can be paused and resumed in multiple positions.

Non-Timelocked Sound Modules

Non-timelocked Sound Modules behave differently in relation to the cursor position on the timeline. A non-looping non-timelocked Sound Module, once triggered on the Timeline, plays through to the end and stops playing, so a single shot non-looping non-time locked Sound Module is not be affected by the pausing Timeline cursor by a Sustain point. A looping non-timelocked Sound Module continues to play so long as it is triggered, and so does not cease playback if the timeline cursor pauses on it due to a sustain point. This is another method of creating a loop effect within FMOD Studio.

Note

You can combine Loop Regions and Sustain Points. The Sustain Points need to be placed inside a Loop Region and Studio pauses at each Sustain Point until Keyoff is triggered. Once the Cursor reaches the end of the Loop Region it returns to the beginning and all Sustain Points function again as expected.

Timelocked and Non-Timelocked Sounds

FMOD Studio has two types of Sound Module for the purposes of playback behavior:

1. Timelocked Sound Modules.
2. Non-timelocked Sound Modules.

The way in which these two types of Sound Modules behave can be considerably different depending on the layout of an Event and the behavior of the Cursor. In all situations the basic playback behavior in FMOD Studio comes down to the following.

- Sound Modules are represented by their associated Trigger Regions on either the Timeline or on a Game Parameter.
- A Sound Module is triggered when the Cursor intersects with its Trigger Region and untriggered when the Cursor ceases to intersect it.
- The nature of the playback behavior when triggered or untriggered depends on the kind of module and its settings.

Timelocked or non-timelocked status, position on the Timeline or Parameter, trigger logic and playback logic can all affect playback behavior.

Timelocked Sound Modules

A Timelocked Sound Module is any Single Sound Module that is not set to loop and whose Trigger Region is on the Timeline. Unless it is subject to pitch adjustment, when a Timelocked sound plays, the position of the cursor on the waveform indicates the part of the sound that is playing.

So, just like in a sound editing tool you can position the cursor at a specific place on the wave form and FMOD plays from that point. This means that a Loop Region that forces the Cursor to loop on the Timeline loops any areas of a Timelocked sound that are intersected by the Cursor's movement. It also means that if the Timeline cursor stops moving due to a sustain point, that trigger region produces no audio, as the cursor does not progress through the waveform.

Non-Timelocked Sound Modules

Any Sound Module that does not meet the criteria of a Timelocked Sound Module is treated as a non-time locked Sound Module. Non-timelocked Sound Modules do not display a wave form in the trigger region because it would make no sense for them to do so. This could be because the Sound Module contains multiple sound files and it would be impossible to display just one, such as in a Multi Sound or Scatterer Sound Module; It could also be because the Sound Module is set to loop and so the relationship between the Cursor and the part of the waveform displayed on the Trigger Region is unpredictable.

A non-timelocked Sound Module, like any Sound Module, is triggered by the Cursor intersecting with its associated Trigger Region. If not set to loop, a non-timelocked Sound Module triggers when the

Cursor makes contact with its Trigger Region. Once triggered, the sound plays out in its entirety and then stops. If the a Non-Timelocked Sound Module is set to Loop, the sound triggers when intersected by the Cursor and loops so long as the Cursor continues to intersect the Trigger Region, and stops playing immediately that the Cursor ceases to be in contact with the Trigger Region. This behavior is the same no matter how long the actual sound files and playlist entries are.

Note

You can combine both Loop Regions and Sustain Points. If Sustain Points are placed inside a Loop Region, Studio pauses at each Sustain Point until Keyoff is triggered. Once the Cursor reaches the end of the Loop Region it returns to the beginning point and all Sustain Points function again.

Loop Region Logic

Selecting a Loop Region causes its Module to be displayed in the Deck area.

A loop region can be made to function only while a certain condition is met by assigning it a parameter condition through the loop region's logic panel. Select which parameter you wish to use to control the loop region by right clicking on the section of the deck panel marked with a '+' and selecting the desired parameter from the context sensitive menu. This causes FMOD Studio to add the logic condition to the panel. Use the ribbon slider that appears to define a range of values by setting a minimum and maximum value, or a single value by setting both ends of the ribbon slider to the same value.

During playback, the loop region functions normally whenever the value of the game parameter is within the range defined in the logic panel, but when the parameter value is outside of the specified range the timeline cursor behaves as if the loop region were not there.

Note

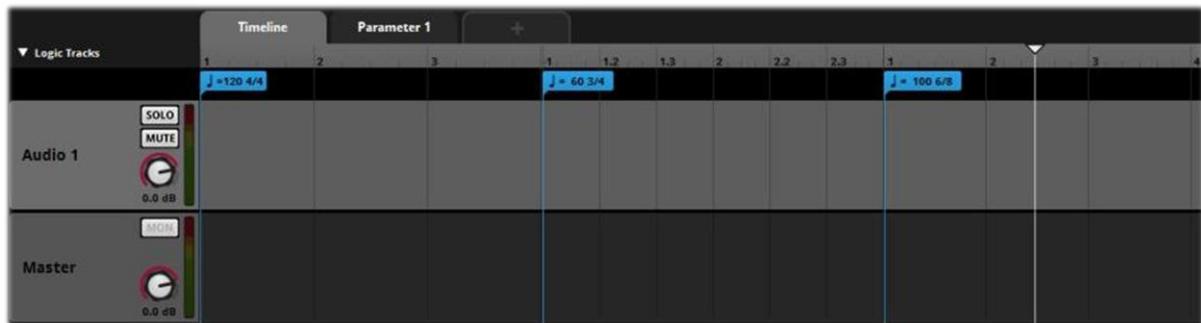
There are four ways to create looping playback within FMOD Studio:

1. A Module's looping playback setting sets one or more sounds or playlist entries within that Module to loop continuously, so long as that Module is intersected by the cursor.
2. A Loop region within the Event Timeline causes the Timeline cursor to loop over that section of the timeline.
3. A destination marker and transition marker can produce a similar effect.
4. Dragging a Timelocked single Sound Module's right edge to the right causes its waveform to be repeated a fixed number of times.
- 5.

Tempo Markers

Tempo Markers are used to define the Tempo and Time Signature measurements when working with the Timeline. Defining values with Tempo Markers alters what information is displayed on the Timeline ruler, but it does not alter the speed at which the Timeline cursor advances. The Timeline

cursor always measures milliseconds, seconds, minutes and hours. The Timeline Display simply adjusts to display bars and beats accurate to real time depending on the defined values.



Three Tempo Markers on the Logic Track

FMOD Studio measures bar and beat values and adjusts the rate of measurement as it reaches each Tempo Marker. Quantization information appears relative to the tempo in the Logic system. Beat and Bar measures on the Track Ruler adjust to correctly display Beats and Bars in relation to the Timeline according to the tempo.

FMOD Studio applies a defined tempo to the right of each Tempo Marker. The Timeline to the left side of the leftmost of an Event's Tempo Markers is essentially Tempo 0, and as such does not support quantization.

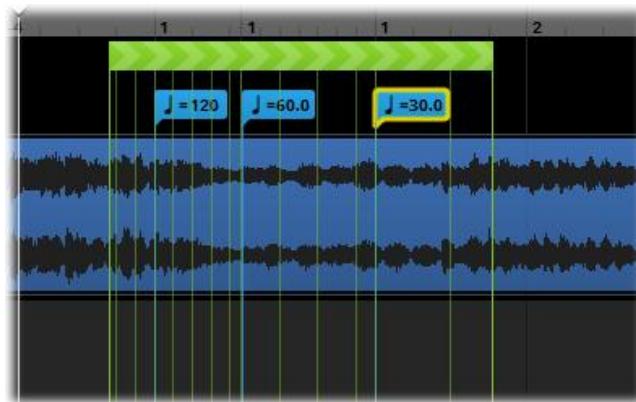
Time Signature values can be defined as follows:

Numerators: 1-999

Denominator: 1,2,4,8 or 16

There is no limit on the defined tempo for Tempo Markers.

Tempo Markers also function as quantization points, making them useful for working with music that includes variations in its tempo, such as almost any recording of live musicians. Each Tempo Marker redefines the bar and beat measures to its right as if a new bar started from that point, so the marker itself can be used to define specific quantization points.



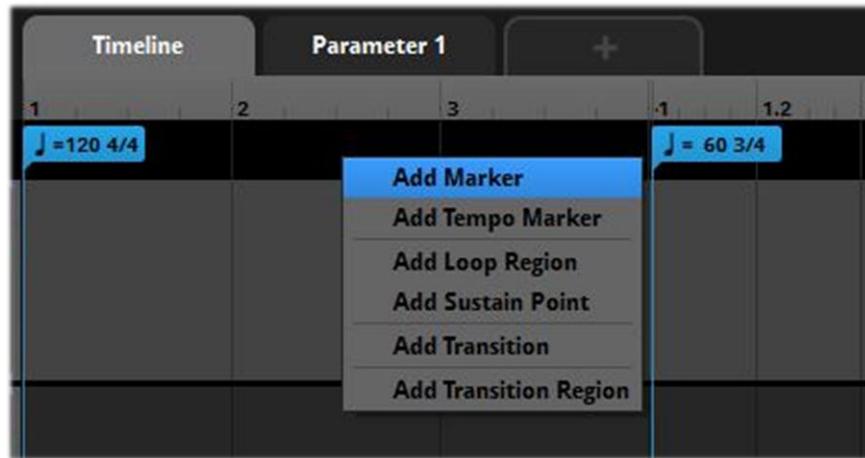
Tempo Markers function as Quantization points

Markers

Markers in FMOD Studio serve two purposes. Firstly they can be very useful for simply indicating certain points along the Timeline and providing a label with useful information. In this capacity a Marker has no functionality beyond being a label. It can be created, named and repositioned as required but it does not affect the playback behavior of the Event in any way.

When used as destination points, Markers provide a landing point for the Cursor when it is redirected by Transition Markers and Transition Regions.

Creating a Marker



Creating a Marker

1. Select an Event to make it the active Event
2. Right click on the Logic Track to open the context menu.
3. Select Add Marker from the menu.
4. Type a name for your new Marker.

Markers can be moved along the Logic Track by dragging them with the mouse. A Marker can be deleted by right clicking on the head at the top of the Marker and selecting Delete from the context menu.

Transition Timelines

Transition Timelines are a feature of the FMOD Studio Logic System. A Transition Timeline is a self-contained miniature Timeline associated with a Transition Marker. When the Timeline cursor reaches the transition marker, it moves along the associated Transition Timeline before arriving at the destination Marker.

If the values of an existing Automation create a significant difference between the Transition point and the Destination Marker of a Transition Timeline, the Transition Timeline automatically blends between the automation values. This produces a smooth transition without the need to rework the existing Timeline or Parameter automation data.

Programmer Sound Modules

Programmer Sound Modules can be added to Event Audio Tracks. These are a special type of sound Module that allows the source audio to be specified in game code at run time. More details about Programmer Sound Modules can be found in the FMOD Studio Programmer's API Documentation.

Multiple Projects

You can open more than one FMOD Studio project at the same time. Each such project displays in its own set of windows, just as if it is the only project open.

To open an additional project when there is already a project open, select 'Open...' from the 'File' menu then select the project. To have only one project open at a time you need to manually close any projects that you do not want open, or exit and restart Studio and open only the project that you want.

Note

If the current project that is open is an empty project with no changes FMOD Studio closes this project when a new project is opened.

You can copy and paste project elements between projects. To do this, both projects need to be open when an object is pasted from one project to another. You cannot copy from Project A, close project A and try to paste into project B. This is because some objects may take up more memory than can be comfortably contained on the operating system's clipboard.

FMOD Studio Music System

FMOD Studio includes a series of tools designed to assist in the creation and implementation of musical works for games. The distinction between music and sound design in games is often blurred. For this reason, all features of FMOD Studio are designed to be flexible enough that the 'Music' tools are suitable for sound design purposes and vice versa. The following tools are classified as musical tools because they deal with many aspects of production more often related to music than to other sound.

The Logic Track

The Logic Track allows you to control the playback behavior of the Timeline of an Event. The Logic Track can have markers placed along it horizontally to define specific behaviors as occurring at points along the Timeline. It can also be expanded vertically to allow multiple levels of placement on a vertical axis. Placement vertically applies a priority to the various markers within the Logic Track. A marker placed higher on the Logic Track is prioritized more highly than markers positioned lower on the Logic Track.



Logic Track Marker Hierarchy

Markers

The simplest way to use a Marker is to indicate a section or significant area by placing and naming a Marker. In this case, Markers are being used much like a street sign. They mark an area with a word or message that can provide useful information to you and your team while developing a project. This function can also be useful for educational purposes as you can clearly indicate various aspects of a project with Markers.



Markers

Markers also have a purpose within the Logic system; as Transition destinations. In this case they quite literally work as street signs as Studio can be instructed to jump to a Marker location when the Transition Marker's conditions are met.

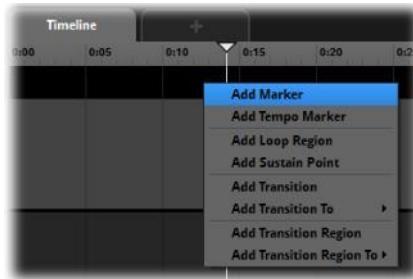
Any number of Transitions and Transition Regions can point to the same Marker.



Markers and Logic

Creating Markers

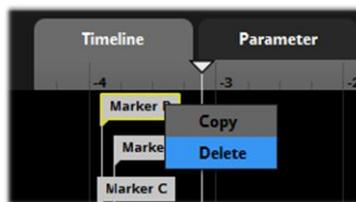
Right click on the Logic Track and select Add Marker from the context sensitive menu to add a Marker. A new Marker is created at the point where you clicked on the Logic Track. Note that logic markers of any kind can only be created on the Timeline.



Add a Marker

Working with Markers

Move Markers by dragging them back and forth or up and down on the Logic Track; it is not possible to set a precise location for a Marker through text entry. Rename a Marker by double clicking on the head of the Marker. To delete a Marker, either click on it to make it active and press the Delete key, or right click on the head of the Marker and select Delete from the context sensitive menu that appears.



Deleting a Marker

Tempo Markers

Tempo Markers look similar to Markers and share most of the basic workflow. The difference is that Tempo marker defines the musical tempo applied to the Event Timeline from the position of the Tempo Marker onward. Multiple Tempo Markers result in the Tempo changing on different parts of the timeline. Tempo Markers also define the time signature of the Timeline making regular changes of time signature possible using multiple Tempo Markers.



Tempo Marker

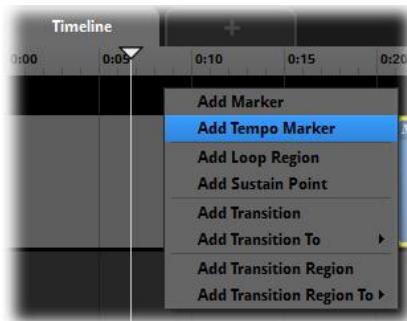
The Tempo of an Event does not affect the rate at which the Cursor progresses along the Timeline. A Tempo Marker applies tempo measurement to the Timeline for the purposes of Quantization for Trigger Region activation. So a Tempo Marker is used in conjunction with the Trigger Behavior Drawer that is part of all Sound Modules.



Defining Quantization

Creating Tempo Markers

Tempo Markers are created by right clicking on the Logic Track and selecting Add Tempo Marker from the context sensitive menu. A new Tempo Marker is then created at that point. Tempo Markers, like other Markers, can only be created on the Timeline and not on other Parameters.

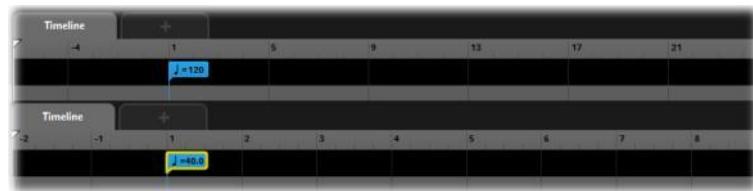


Adding a Tempo Marker

Placing a Tempo Marker in an Event changes the Timeline to display Bars and Beats. The distribution of the Bar and Beat measures depends on the tempo defined in the Tempo Marker.

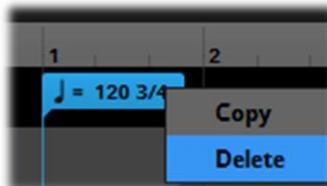
Defining Tempo

The default value for a new Tempo Marker is 120 BPM. To change this, double click on the Tempo Marker head and input the desired tempo value. Altering the Tempo value results in the Bar and Beat measures on the Timeline being repositioned accordingly.

*Default Tempo Markers*

Working with Tempo Markers

Tempo Markers can be moved by simply dragging them back and forth or up and down on the Logic Track; it is not possible to set a precise location for a Tempo Marker through text entry. Right click on the head of the Marker to select Copy or Delete from the context sensitive menu (you can use the shortcuts Ctrl +C (to copy) or the Delete key).

*Deleting a Tempo Marker*

Note

The position of a Tempo Marker indicates where the tempo of an Event is defined. Because the speed of the cursor along the Timeline usually reflects real time, the placement and positioning of the Bar and Beat measures automatically change to reflect the tempo as defined by the most recent Tempo Marker. So, while a distance on the Timeline that equals ten seconds of time still equals ten seconds of time if the tempo is changed, the number of bars and beats within those ten seconds changes to fit depending on the defined tempo.

Transitions

Transitions are a type of Marker that directly affects the flow of Event playback. A new Transition can be linked to an existing Marker or created as a blank Transition to be linked at a later time. A Transition serves no purpose unless it is provided with appropriate information to define the transition logic. A Transition is automatically named to indicate its destination.

*Transition to a Marker*

Creating Transitions

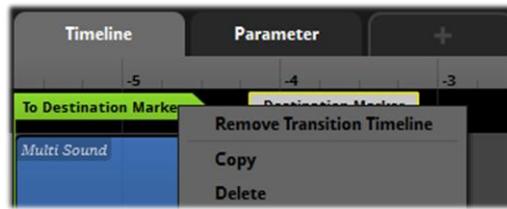
To create a Transition Marker right click in the Logic Track and select Add Transition or Add Transition to.. from the context sensitive menu. If Add Transition To.. is selected a further context menu displays to allow you to select from the available Transition targets. A new Transition is created at the point where you clicked on the Logic Track.



Creating a Transition

Working with Transitions

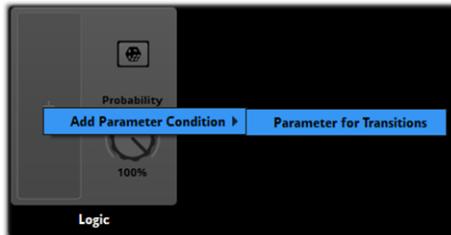
To delete or copy a Transition, click on it to make it active and press the Ctrl +C (to copy) or the Delete key. You can also right click on the head of the Transition and select Copy or Delete from the context sensitive menu that appears.



Transition Context Sensitive Menu

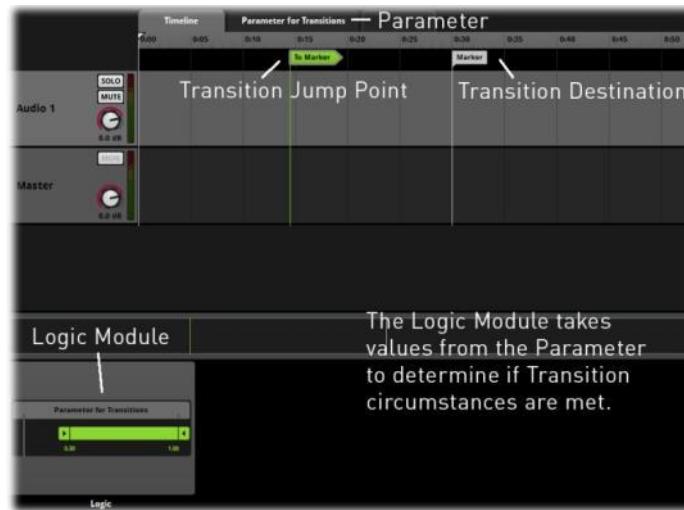
Transition Functionality

A selected Transition displays its Logic Module in the Deck area. Transitions function by defining a Parameter as a Parameter condition.



Selecting a Parameter for your Transition

Select the Parameter you wish to use to control the Transition by left or right clicking on the Logic Module's '+' area and selecting the desired Parameter from the context sensitive menu. FMOD Studio adds the Parameter condition to the Logic Module. The Logic Module allows you to define the value or the Parameter that triggers the Transition behavior. Define either a single value by entering the same number into both ends of the Ribbon Slider or a range of values by defining a minimum and maximum value with the Ribbon Slider. Double click on the number field to enter the number directly.



Transition Functionality

During playback FMOD Studio references the Logic Module when the Cursor makes contact with the Transition Marker. If the value of the Game Parameter is in the range defined in the Logic Module, Studio instantly jumps the cursor position to the target Marker and continues Cursor progression from that point. If the Parameter does not equal the defined values, the Cursor continues to move along the Timeline as normal.

Transition Regions

Transition Regions are a type of Marker Region that directly affects the flow of Event playback. A Transition Region works just like a Transition Marker but can be extended over an area instead of a single point on the Timeline. When a Transition Region is created it can be defined to link to an existing Marker or it can be created as a blank transition to be linked at a later time.

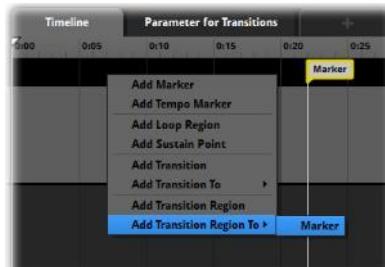
A Transition Region serves no purpose unless it is provided with appropriate information to define the transition logic.



Transition Regions

Creating Transition Regions

To create a Transition Region right click in the Logic Track and Select Add Transition Region or Add Transition Region To from the context sensitive menu. If Add Transition Region To is selected, a further context menu allows you to select from the available destination markers. Whether or not you select a destination marker as a target, a new Transition Region is created at the point where you clicked on the Logic Track.



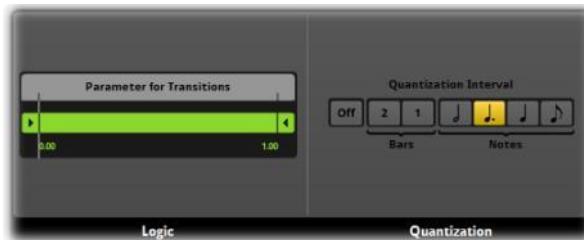
Creating a Transition Region

Working with Transition Regions

Move Transition Region end points by dragging them back and forth or up and down on the Logic Track (moving up or down repositions the entire Region on the new vertical axis). A Transition Region is automatically named to indicate its destination. To delete a region click on it to make it active and press the Delete key, or right click on the head of the Marker and select Delete from the context sensitive menu that appears.

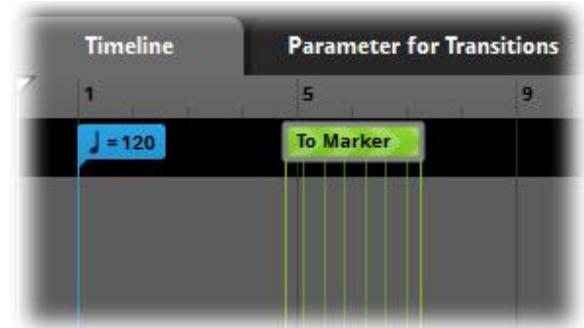
Transition Region Quantization

A Transition Region's Module also includes a Quantization section displayed in the Deck area. The Quantization section allows you to define acceptable transfer points within the region. This means that Studio does not simply perform the transition at any point in time. Instead it waits until it reaches a suitable bar or beat as defined by the Quantization section.



Quantization

When a particular quantization value is selected on the Quantization section, the Transition Region displays those points on the Timeline with additional marker lines. This makes it clear exactly where transitions can occur within the region.



Quantization Marker Lines

Transition Timelines

Both Transition Points and Transition Regions allow you to create Transition Timelines. A Transition Timeline is a sub Timeline that exists only as part of its associated Transition. This function allows you to add content to a Transition Point or Transition Region that is activated when the Transition is triggered. Essentially the cursor on the main Timeline is diverted to the destination Marker via the

Transition Timeline. This can be very useful in dynamic music as a way of providing actual transitional content to be played to blend the two points of transition.



A Transition Timeline opened from its Transition Point

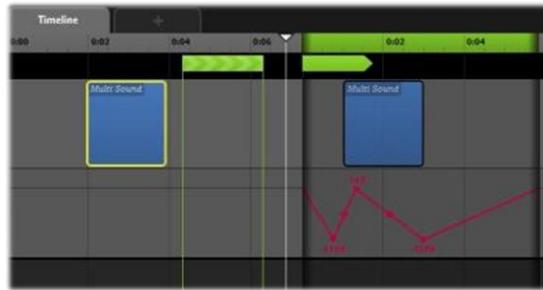
To create a Transition Timeline, double click on the target Transition Point or Transition Region. FMOD Studio then creates or opens the Transition Timeline as displayed in the previous image. Once created, a Transition Timeline can have content created, copied and pasted into it in the same way the main Timeline can.



Creating Modules within a Transition Timeline

You can add Automation to objects within the Transition Timeline and Automation Tracks continue to function through a Transition Timeline. This allows you to automate the properties of Modules outside a Transition Timeline within a Transition Timeline.

Transition Timeline duration is displayed by the values of its green Timeline display. Adjust the duration by moving the edges of the Transition Timeline Region on the ruler.



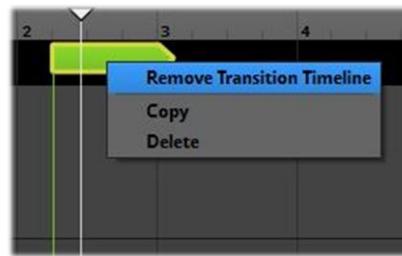
The Automation within the Transition Timeline is the for Module outside of it

A Transition Timeline cannot include Logic elements within it so the Timeline displays and measures Tempo, bar and beat measurements relative to the most recent Tempo Marker to the left of the associated Transition Marker. The Transition Timeline's ruler displays seconds and minutes unless a Tempo Marker has not been added to the event.

To close a Transition Timeline simply double click on its associated Transition Marker or Region. Double clicking again opens the Transition Timeline and displays its contents. Dragging the right edge

of a transition timeline to meet its left edge automatically deletes the Transition Timeline and all its contents, so be careful when creating very small Transition Timelines. We recommend you zoom in using the Birdseye view to avoid losing any data.

It is also possible to delete a Transition Timeline from a Transition Marker or Region. First make sure the Transition Timeline is closed and then right click on its associated Marker or Region and select Remove Transition Timeline from the context menu.



Deleting a Transition Timeline

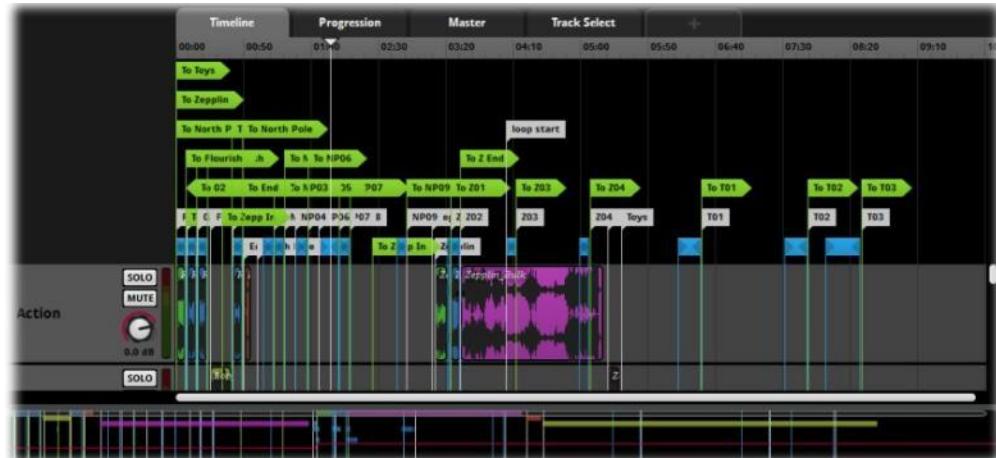
Technically, there are no limits to the number of Transition Timelines you can create and work with, however practical considerations during design are another matter.

Loop Regions and Sustain Points

Both Loop Regions and Sustain points function in the exact same manner they do in non-musical FMOD Studio events, and can be used in conjunction with all the other Logic Track markers. This means, for example, that a Loop Region can be combined with Transitions and Markers to create a looping section of an event that, under the correctly defined Parameter values, jumps from the loop to another area of the Timeline.

Note

Destination Markers, Transitions and Transition Regions can be placed anywhere on the Timeline in relation to each other. This means that Transitions can jump either forward or backward along the Timeline. The head section of a Transition points in the direction of its associated Marker as a visual aid for how the Logic of the Event plays out. Transitions cannot be moved during Playback as this can change the entire relationship of Playback to time and cause errors and issues.



Loop Regions and Sustain Points

Trigger Behavior

All Sound Modules contain a Trigger Behavior drawer in the Deck area. This drawer is used to define Delay and Quantization behavior for a Sound Module. The Trigger Behavior drawer can be accessed by clicking on the white disclosure triangle at the left edge of the module. This extends the drawer and allows access to the Trigger Behavior Properties.



Trigger Behavior

By default the Trigger Behavior of a Sound Module is set to immediate triggering with no Quantization. This means that by default, a Sound Module triggers the instant the Cursor makes contact with the Trigger Region in either the Timeline or in a Game Parameter. It also means the Sound Module untriggers the instant the Cursor is no longer in contact with the Trigger Region.

Note

'Untriggering' means slightly different things depending on the placement of the Trigger Region and the type of Sound Module. A looping Sound Module stops playing as soon as the Cursor moves off its Trigger Region, while a non-looping Sound Module plays through until its end. A Timelocked Sound Module in the Timeline falls silent immediately when untriggered.

Defining Trigger Behavior

There are two methods of defining Trigger Behavior in FMOD Studio. Time allows an absolute period of time or range of time to be defined as a delay before a Trigger Region commences playback. Tempo is used to provide quantization information for triggering playback using musical measures of Bars and Beats.



Time or Tempo

Time

The Time Property of the Trigger Behavior allows you to define the period of time that Studio counts before a Sound Module begins to play. While a Sound Module is triggered the instant the Cursor makes contact with its associated Trigger Region, it does not actually have to start playing the sound at this point. The Time Property can define a delay in the playback.

The Time area has a Trigger Delay Property that works via a Ribbon Slider. The slider defines the minimum and maximum time values in milliseconds or second before a Sound Module starts to play its sound once activated. If both the min and max values are the same then a fixed delay is always applied to that Sound Module every time it is activated. If the min and max values are different then they create a window in time. FMOD Studio counts at least the min time value and no more than the max time value, but between those two it randomly selects a period of delay to apply to the Sound Module.



Trigger Delay Property

Define the values by either dragging the ends of the Ribbon Slider to the desired values, or double clicking on the displayed numbers to directly input exact values.



Entering Exact Values

Note

You can define both Delay and Quantization Properties and combine the results of both. Be careful with this to avoid introducing undesirable playback behavior.

Tempo

The Tempo area defines Quantization behavior for event using various Logic Track markers. A Tempo Marker must be included in an event to make use of Tempo quantization.



Tempo Marker

When an event includes a Tempo Marker, FMOD Studio applies tempo tracking as well as counting regular time. Tempo quantization allows you to define when Sound Modules trigger in relation to the event Tempo. This allows Sound Modules to be delayed slightly so that they trigger in time with the defined Tempo and as such sound more "musical".

Once a Tempo Marker is added to an event, the Timeline changes to display tempo measures in Bar and Beat format. Sound Module Trigger Regions can be moved to align with these measures, but the Quantization property ensures they trigger in sync with the musical measures.

The Quantization Property can be set to either bar values of 1 or 2, or beat values of minim, crotchet, dotted crotchet or quaver measures.

Seek

The Seek panel allows you to specify a start offset for a sound module. This means you can define where within a Sound Module it actually starts to play. So a Seek value of 5% means that when the cursor triggers the Sound Module, FMOD Studio commences playing from 5% into the sound file.

The Seek Property has a range from 0-100% with a default value of 0%. Like most dial-based Properties, Start Offset can be automated to add further control.

Note

The Start Offset value is applied to any sound file within the Playlist of the target Sound Module. Because of this, it is important to be aware of the length of all the sound files within a Multi Sound or Scatter Sound Module.

Polyphony

The Polyphony Property defines the number of instances of a particular Sound Module that can be voiced simultaneously. So if set to a value of 1, the target Sound Module only ever has a single instance of it played at the same time. The Polyphony Property has a range from 1-64/Infinity with a default value of infinity.



The Polyphony Property

Note

Once the maximum defined Polyphony value is reached, any requests for further instances of the Module to play simply fail until a voice channel is available. FMOD Studio does not queue requests; if a voice channel is not available the request fails and is discarded. Once a currently playing voice channel is available any future requests for that Sound Module are free to play.

Parameters

A parameter is an event property whose value can be updated while the event is playing. Each instance of an event has its own parameter values, independent of all other instances of the same event. This allows your game's code to control each event instance independently.

The Timeline Parameter is included by default in all events. An event can also have “built-in parameters” that are automatically updated based on the event’s location relative to the listener, and

“game parameters” that can be updated by function calls in your game’s code. There is no limit to the number of parameters than can be added to an event.

Parameters are used to represent any element of your game that alters the behavior of an event. For example, an event that represents a car engine may feature “RPM”, “Speed” and “Engine Load” parameters, and an event that plays different grunts based on the player character’s current state might have a “playerHealth” parameter.

Note

Each newly created Parameter has a tab just like the Timeline. Click on the tabs to switch between Parameters and Timeline as required.

How Parameters Can Affect Events

There are a number of different ways a parameter may alter the behavior of an event. The most common one is triggering sound modules. If a sound module’s trigger region is present on a parameter, that parameter can trigger that sound module: If the value of the parameter is within the range of values described by the trigger region, and all other trigger conditions for that sound module are met, that sound module is triggered.

Parameters can be used to create trigger conditions for sound modules, snapshot modules and logic markers. If a module or logic marker has a parameter listed in the trigger conditions list of its ‘Trigger Behavior’ drawer in the deck, that module or trigger region will only trigger or function while the parameter’s value is within the range specified by that condition’s ribbon slider.

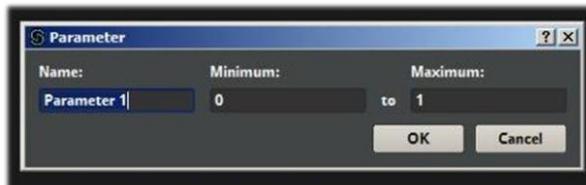
Finally, parameters can be used to create automation.

Important

Sound and snapshot modules don’t have to be placed on the timeline. They can also be placed on other parameters. If a sound module is on a parameter other than the timeline, it will be triggered by that parameter instead of the timeline.

Parameter Properties

When a Parameter is initially created a dialog box allows you to define the name of the Parameter as well as its minimum and maximum range values. These values can be edited at any time by right clicking on the parameter’s tab and selecting Edit from the context-sensitive menu.



Parameter Properties

Parameters have additional properties that can be displayed in the Deck by clicking on the name of the parameter where it appears to the right of the transport controls.



Parameter Property Module in the Deck

Value

The Parameter Value section of the parameter module contains a dial that represents the current value of the parameter. For ease of access, this dial is also found in the transport bar at the top of the Event Editor.

Set as Initial Value

If you right-click on a parameter's value dial and select 'Set as Initial Value' from the context-sensitive menu, the current value of the dial becomes the value to which the parameter is set by default when a new instance of the event is created. Note that the current value of a parameter can be altered after the event instance is created but before it is played by your game's code.

Hold

The 'Hold' button determines whether the value of the parameter can change after the event begins to play. This does not prevent you from updating the values of an event instance after that instance is created but before it starts to play.

Smoothing

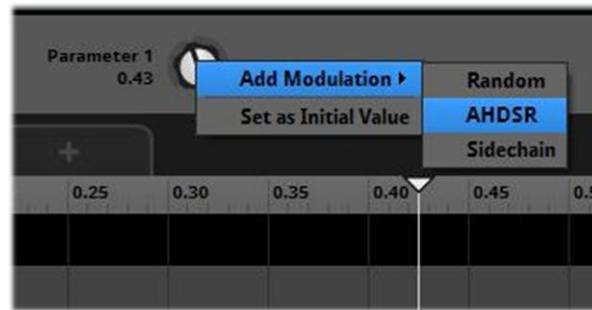
The Smoothing section of the parameter module contains a dial that represents the parameter's seek speed.

If this dial is set to 'Instant,' any changes to the parameter value are applied instantly.

If the dial is set to anything other than 'Instant,' the parameter is said to have a 'seek speed.' If a parameter with a seek speed is updated while the event is playing, the value of that parameter transitions gradually from the existing value to the new value at a rate of X units per second, where X is the value of the seek speed dial. For example, if a parameter whose current value is 0, and whose seek speed is 2, is set to a parameter value of 10, the value of that parameter increases at a rate of 2 units per second until it reaches 10.

Parameter Modulation

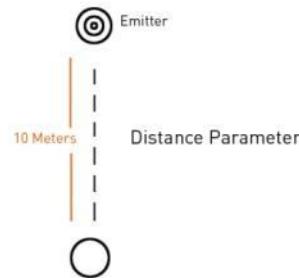
Random, AHDSR and Sidechain modulation can be added to a Parameter. To add modulation to a Parameter, right click on its dial in the Transport area or in the Deck and select the desired modulation type from the context menu's 'Add Modulation' submenu.



Adding Modulation to a Parameter

Distance Parameter

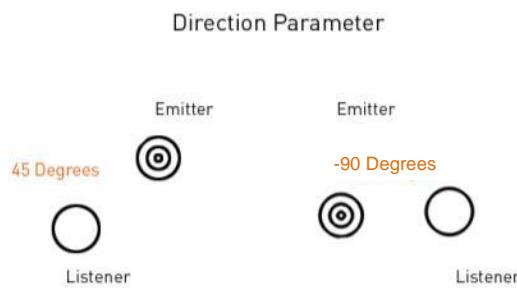
The Distance Built-in Parameter automatically calculates the distance in the game world between the Listener and an instance of the event's Emitter.



Distance is measured in game distance units

Direction Parameter

The Direction Parameter relates to the direction the listener is facing in a 3D game, it represents the angle to the Emitter from the Listener. This is shown in degrees from -180° to 180° with 0° being directly in front of the listener and 180° positioned directly behind.



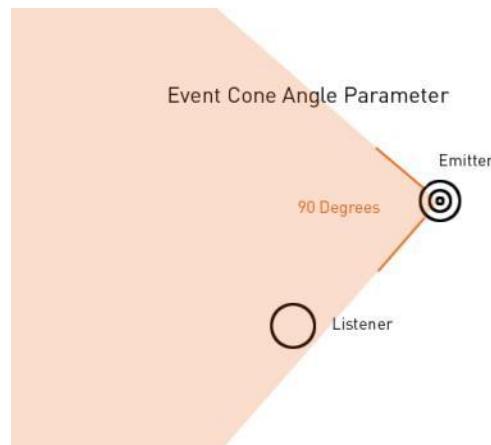
Two examples of 3D position setting Direction Parameter

Elevation Parameter

The Elevation Parameter functions in a similar manner to the Direction Parameter but instead of measuring direction on the horizontal plane it measures it on the vertical plane. Thus, an event five distance units away from the listener and at an altitude three distance units greater than it has an elevation of about 36.9 Degrees.

Event Cone Angle Parameter

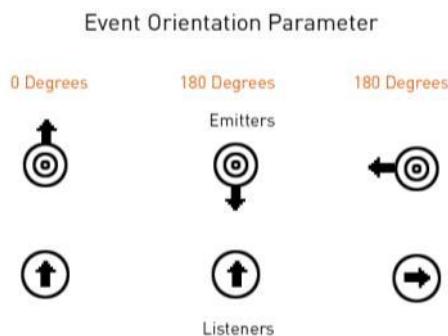
The Event Cone Angle represents the angle between the direction towards the Listener and the event's facing, and is often used to define the size of the emission cone for an event. Some sounds emit in a non-spherical manner and this Parameter can be used to define playback behavior for these conditions.



The Event Cone Angle Parameter defines how the sound projects at different angles

Event Orientation Parameter

This Parameter measures the difference between the orientation of the Listener and the orientation of the Sound Emitter for the event. If both the Listener and Emitter are facing the same direction the difference is 0 degrees, if the Listener turns to the right by 90 degrees and the Emitter turns to the left by 90 degrees the difference would now be 180 degrees. This only takes into account orientation on the horizontal plane.



Event Orientation based on the orientation of both Emitter and Listener

The Timeline Parameter

The Timeline Parameter is the default setup for events created in FMOD Studio and measures time.

The Timeline functions in a very different manner to other Parameters. The most obvious characteristic of the Timeline is that it not only does it measure time with its values but it forces the cursor to advance through time as part of the play process in a way similar to many linear audio editing programs.

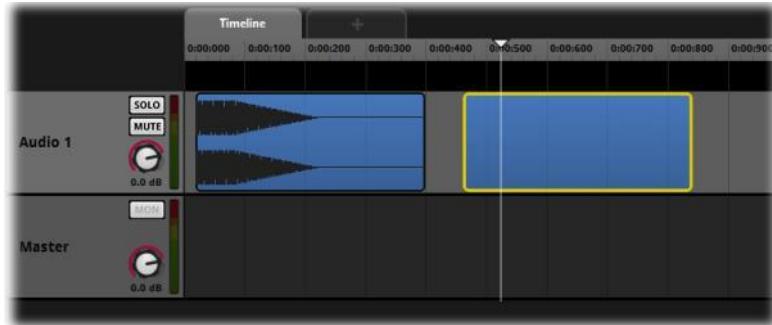
The Timeline incorporates the Logic Track that allows the creation of Loop Regions, and Sustain Points, and other logic markers. The nature of these markers means they need a time-based Parameter in which to work. The Timeline also includes the Time display that corresponds to the value of the cursor as it moves along the Timeline.

Aspects of Time-based Events

There are several aspects of timeline-based events that need to be understood to avoid unexpected playback behavior. The non-linear nature of FMOD Studio means that even the Timeline does not always reflect the standard passing of time during playback.

Timelocked sounds

Timelocked sounds are Sound Modules that contain a single consistent and predictable sound file. The Trigger Region for a Timelocked sound actually displays the wave form of the sound file. Files that are not Timelocked cannot display the wave form for the simple reason that there is no consistent wave form to display. A Multi Sound Module is an example of a Non-Timelocked sound. The fact that a Multi Sound Module can contain any number of sound files means that it is impossible for Studio to display a single wave form.



A Timelocked and Non-Timelocked Sound Module

Timelocked sounds also display different playback behavior to Non-Timelocked sounds. Because of the state of a Timelocked sound it is possible to place the cursor on the Timeline at any point within the range of the sound and trigger the sound from exactly that point. FMOD is aware of the exact nature and timing of that trigger region as it is locked and so can commence playback from the point that corresponds to the cursor position in relation to the waveform.

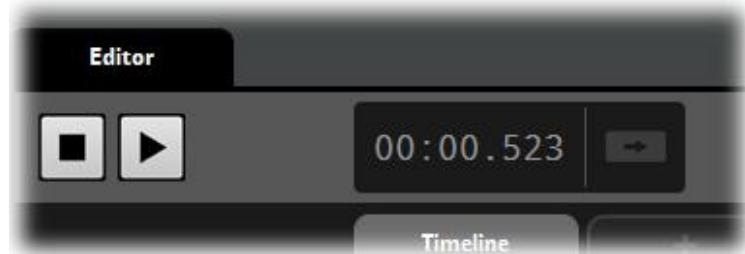
This is not possible with Timelocked modules. The cursor intersecting with any point of a non-locked trigger region always triggers the sound from the beginning of the sound file, or from the point in the sound file specified by the module's Start Offset property.

Note

Setting a Timelocked Single Sound Module to looping stops it from being Timelocked.

Logic Markers and Time

Loop Regions and transition markers, by their very nature, change the way the timeline parameter functions. The Time Display on a Timeline indicates the position of the cursor on the Timeline, not the actual real time that has passed since the event began auditioning. For example, if an event contains a loop region from 0:00 to 5:00, the time display will continually cycle through the values from 00:00.000 to 00:05.000, repeating as the cursor loops.



Timeline Value Display

The Distance Parameter

The Distance Parameter is a unique and specific type of Parameter available for FMOD Studio events. Distance Parameters allow Modules and Effects to be influenced directly by the relative distance between the sound source and the listener.

When you create an event in FMOD Studio by default it has certain behavior assigned to it. Events are 3D sound objects by default, including functions in relation to how the sound works in a 3D space depending on the relationship between the sound source and the listener. The Distance Parameter adds further functionality based on this distance information. Modules and Effects can be added to a distance Parameter and the value of distance can be used to define playback behavior.

Step 1 Creating a Distance Parameter



Create a Parameter

1. Right click next to the Timeline tab above the Timeline ruler
2. From the context menu select Add Built-in Parameter, then Distance...
3. (Optional) Enter the desired range for the new Parameter

Note

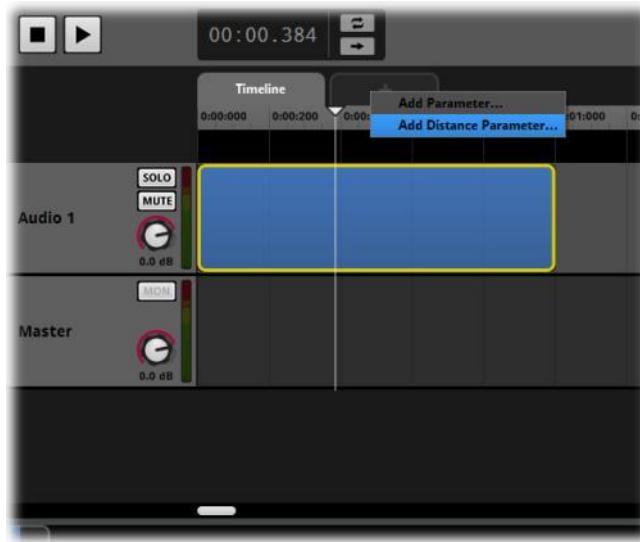
An Event in FMOD Studio can only include a single Distance Parameter.

The Event Cone Angle Parameter

The Event Cone Angle Parameter is a unique and specific type of Parameter available for FMOD Studio events. Angle Parameters are influenced by the relative angle between the facing of the sound source and the listener.

Because events are created as 3D sound objects by default in FMOD Studio, there are functions that define how the sound works in a 3D space depending on the relationship between the sound source and the listener. The Angle Parameter allows you to add further functionality based on this information. Modules and Effects can be added to an Event Cone Angle Parameter and the value of angle can be used to define playback behavior.

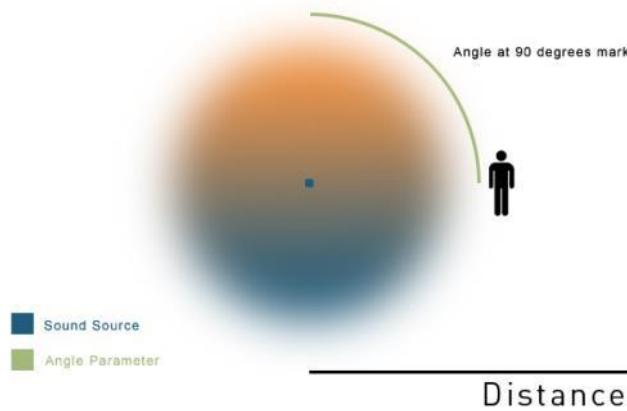
Step 1 Creating an Event Cone Angle Parameter



Create a Parameter

1. Right click next to the Timeline tab above the Timeline ruler
2. From the context-sensitive menu select Add Built-in Parameter, then Event Cone Angle Parameter...
3. (Optional) Enter the desired range for the new Parameter

The Event Cone Angle Parameter describes the relative angle of the listener from the forward orientation of the sound source. An angle of 0 degrees represents a position directly in front of the sound source while 180 degrees represents directly being the sound source. This Parameter does not in itself reflect any change in the sound being produced; it provides a blank canvas with which you can create the type of audio manipulation over angle that their project requires.



The Angle Parameter

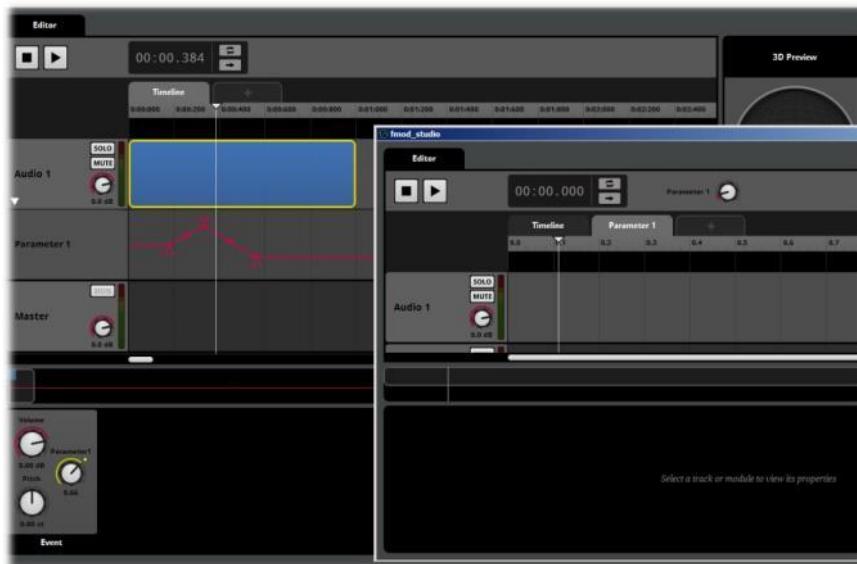
The above diagram includes an area of transition between two colors that represents how a sound could alter as the Angle Parameter increases. These effects would be achieved by adding Automation to event Audio Tracks that alter over the range of the Angle Parameter. The position of the figure indicates listening at a position of 90 degrees on the Angle Parameter where there is an equal blend of the front and rear positional effects.

Note

An Event in FMOD Studio can only include a single Angle Parameter.

Controlling Events External to an Event

The ability to nest events within events is supported by the ability to control the Parameters of any event from its parent event. This allows complex interconnection of events and event Parameters.



Nested Event Parameter

The above graphic illustrates controlling a nested event's Game Parameter. The event in the foreground window is nested in the Timeline of the event in the main Studio Edit region in the background. The Game Parameter Track directly below Audio 1 Track containing the red Automation line is Game Parameter 1 from the event in the foreground window. This series of controllers lets you make changes within the parent event to influence playback of the event nested on that Timeline. The nested event also progresses through its own playback process independently until it is affected by the controlling parent event.

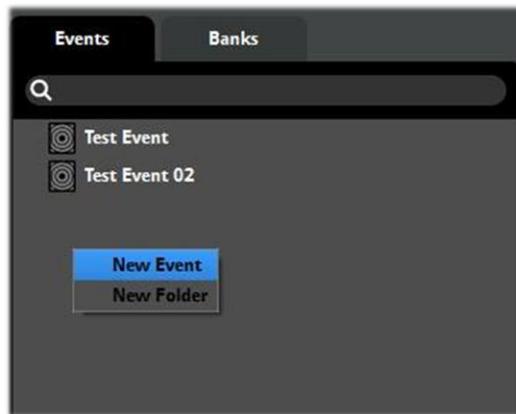
Automating Parameters of Nested Events

The most versatile form of external Parameter control is to assign a Parameter Automation to the direct parent of a nested event. This increases the functionality of the nested event beyond simple playback when triggered.

Tutorial

This tutorial explores the steps in to create a nested event within an event and define an external Parameter controller.

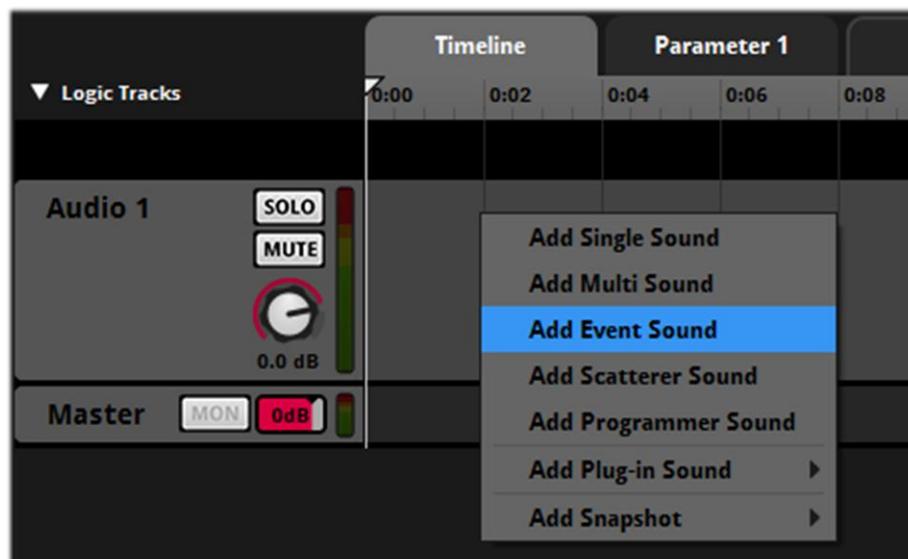
Step 1 Create a New Event



Creating an Event

1. Within the Event Browser, right click and create a new event.
2. Name the event as required.

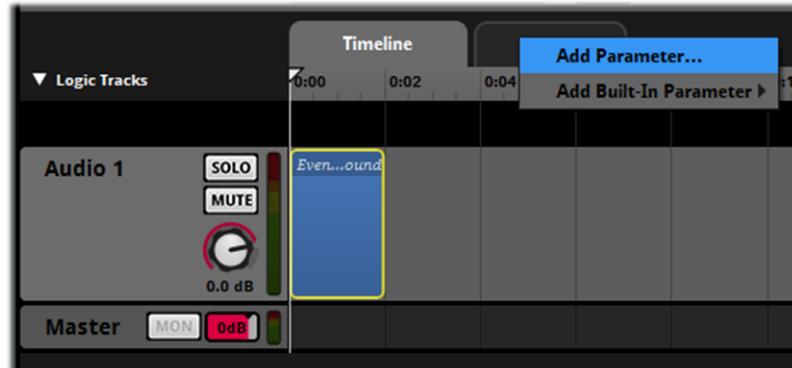
Step 2 Create an Event Sound Module



Creating an Event Sound Module

1. Within the Editor Region right click in an Audio Track and create an event Sound Module.
2. Double Click on the event Sound Module's Trigger Region to open the nested event in the event editor.

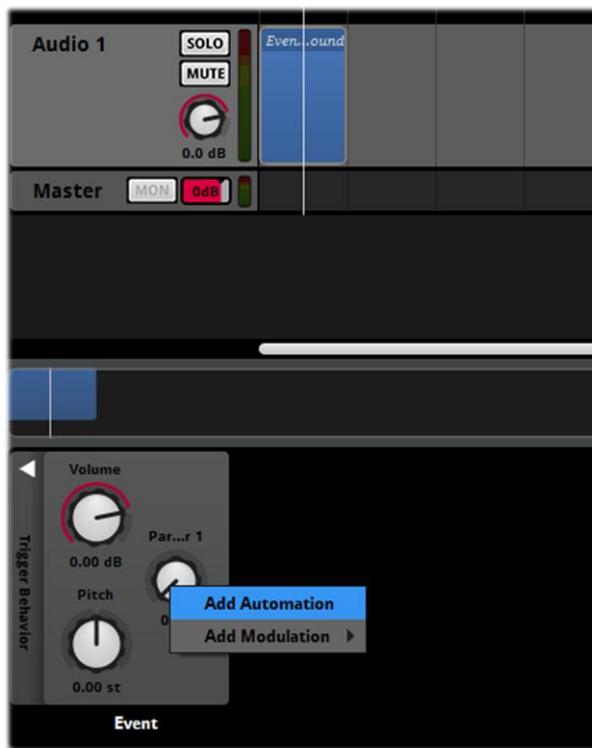
Step 3 Create a Parameter



Creating a Parameter

1. Right click next to the Timeline tab and select Add Parameter... from the context menu.
2. Name the new Parameter as required and click OK.
3. Click on the first item in the breadcrumb trail to return to the parent event.

Step 4 Create a Parameter Automation Controller



Creating an Automation Controller

1. In the original event Click on the Event Module's Trigger Region to make it active.
2. In the Deck Region right click on the Module's Parameter knob to open the context menu and select Add Automation.

FMOD Studio creates a new Track beneath the Audio 1 Track. This is a Parameter Automation Track and it can be used to control the Parameter of the nested event. Click anywhere within this Track to add points and define Automation curves.

Linking Parameters of Nested Events

A nested or reference event's parameter can be controlled via the parent event's parameter values under certain circumstances. This technique can be very useful, but requires some specific settings.

If a nested or reference event features a parameter with the same name as a parameter in the parent event, the nested event's parameter value will match that of the parent event's parameter unless told otherwise. This even works for nested events in the playlist of a multi sound or scatterer sound module.

If the child Parameter values are different to the Parent Parameter values then the child's value range is scaled to match the parent Parameter's range.

Although this linking function only occurs from a parent event to its direct child it can be chained to extend the process. As long as each event in a chain has a Parameter of the same name, it is possible to create a parent event and then multiple levels of nested events all with the same Parameter.

Breaking the Link

There are various situations where the default behavior of the linked Parameters may be undesirable so under these situations the link is broken.

If the child event's Parameter is automated or modulated in the parent event, the link is broken. Any alteration to the events that might break this link functionality would need to be removed for the link to re-establish its behavior.

It may in some circumstances be undesirable for the referenced event to track the Parameter values in every instance, so for instances where it is not desirable it is possible to purposefully break the link without changing the playback behavior. Creating Automation for the nested event's Parameter breaks the link behavior, but if no values are defined on the Automation line then the playback is not altered. This technique can be used to specifically disrupt the linking behavior when needed.

Tags

FMOD Studio allows user to create Tags to assist in administering events within a project. Tags provide a method of assigning extra information to events beyond their event name.

The Event Browser Search Bar allows for search terms to be used to filter what is displayed in the Event Browser. This is very useful for finding specific events within large projects that may contain hundreds or thousands of events within it. Tags allow for extra information to be defined for event management and also appears in Browser Searches other than the Event Browser depending on circumstances.

All events have a Properties area where Tags can be assigned to the event. The exact nature of the Tags is entirely up to you and the requirements of a project. Tags such as Weapons, Vehicles, Dialogue, Player, and Environment would be common examples but anything can be used. They are words selected to provide useful information to a production team so create Tags as you see fit.

*Properties Tags*

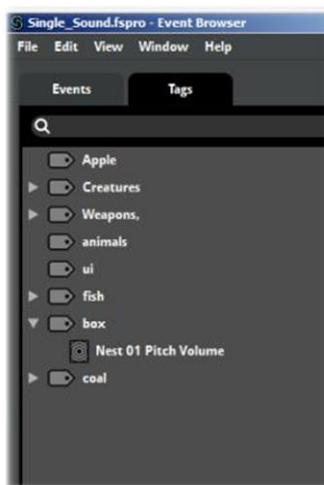
Searching for Tags

Tags can be searched for in the Browser Search Bar by prefixing a search term with the # symbol. Unlike regular searches, Tags are displayed across search Bars in several areas of Studio. The following list outlines how the tags are displayed in different areas:

- Event Browser tag searches displays all the appropriate events within a project that correspond to the defined Tag
- Mixer Browser Tag searches displays the relevant tagged events
- VCA Browser Tag searches only displays associated events that have been assigned to VCAs. As not all events are necessarily assigned to VCAs this may not be a complete list of all the relevant events within a Project
- The Banks Browser is similar to the VCA Browser in that not every event is assigned to a Bank so a Tag search only displays matching Tags that have been assigned to a Bank.
- Tags are not displayed within the Audio Bin Browser Search Bar.

The Tag Browser

The Tag Browser functions like any other Browser in FMOD Studio but displays a list of the Tags within a Project. This is useful for administering a Projects Tags, deleting unwanted Tags as well as checking if events have been correctly assigned to Tags and alerting assignments if required.

*The Tag Browser*

Tags can be created and deleted in the Tag Browser. Events can also be assigned to a Tag in the Event Browser via the context menu. Once assigned, an event is displayed in the Tag Browser under

the appropriate Tags. The Sort by function in the context menu reorders Tags alphabetically. If the Menu is accessed in the general Browser area Studio sorts all Tags alphabetically. If the context menu is accessed while a Tag is selected, Studio sorts all the assigned events under that Tag into alphabetical order.

Creating and Assigning Tags

Tags are a useful method of assigning extra information to events and grouping events into categories.

Creating a new Tag

There are two methods of associating an event with a Tag.

Method 1: Adding Tag information to an Event



Event Properties area

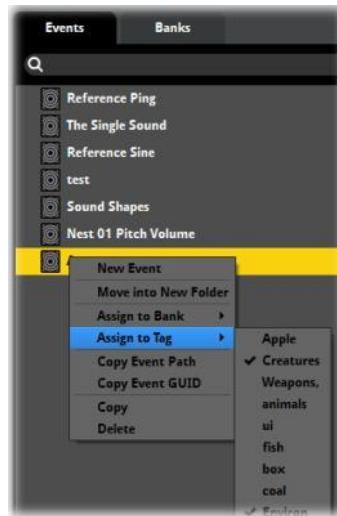
1. Click on an event in the Event Browser to select it
2. Within the event Properties area Click in the Tags region
3. Type in the name of the Tag you want to add to this event
4. Press Space or Enter to complete your Tag and start a new one, or
5. Click elsewhere in the interface to finish adding Tags.

Note

Tags can only be single words; the space bar finishes the Tag you are typing and opens a new tag box to start creating a new Tag in the same way hitting Enter. Tags can include numbers and other characters.

Method 2: Assigning an Event to a Tag

1. Click on an event in the Event Browser to select it
2. Right Click/CTRL Click to access the context menu
3. Select Assign to Tag from the menu
4. Assign the event to an existing Tag or select New Tag...
5. Define the name of a new Tag is creating a new Tag



Event Context Menu

Creating a Tag in the Tag Browser

1. Access the Event Browser via the Menu Bar/Window/Event Browser
2. Select the Tag Browser via the Tags Tab
3. Right Click in the general Browser area to access the context menu
4. Select New Tag from the context menu
5. Define the name of the new Tag



The Tags Browser

Events can be assigned to Tags or Tags can be assigned to events, either method results in an association between an event and any and all tags associated with it. There is no practical limit to the number of Tags that can be created and the number of Tags that can be assigned to an event or the number of events that can be assigned to a Tag.

Note

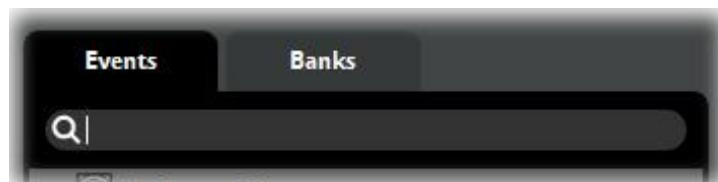
Each Tag in the Tag Browser is unique; there cannot be two instances of the same Tag.

The Browser Search Bar

FMOD Studio includes a Browser Search Bar in various locations within the interface. The Browser Search Bar allows for elements within a project to be searched for and displayed.

Large projects often contain hundreds, thousand or even tens of thousands of objects. Being able to find specific elements of a project quickly and efficiently is a critical aspect of project management and workflow.

You can type in complete or partial words or even single letters to define a search. FMOD Studio instantly updates the items displayed in the browser that match the search terms. The items listed are relevant only to the type associated with the active browser, so searching in the event browser displays matching events, searching in the Audio Bin only displays matching sound files and so on.



The Browser Search Bar

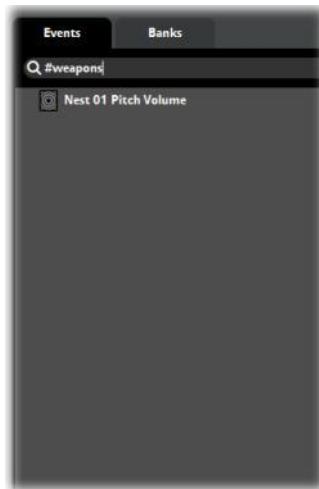
GUID Searches

You can enter GUIDs into the Search bar to filter FMOD Studio's Browser display. This results in the browser displaying only items that have a matching GUID. It is only possible to do this with complete GUIDs so the result is almost always a single entry of the event, Snapshot or Bus with the matching GUID.

Tag Searches

Tags can be searched for in the Browser Search Bar by prefixing the search word with the # symbol. Tags are unique in that they are displayed across different Browser Search Bars depending on circumstances.

- All events associated with a defined Tag search are displayed in the Event Browser
- All events associated with tagged events are displayed within the Mixer Browser
- Only events assigned to VCA that match a defined tag search are displayed in the VCA Browser.
- The Bank Browser is similar to the VCA Browser in that not all events must be assigned to a Bank, so a tag search in the Banks browser may also display an incomplete list of tagged events as only those that have been assigned to Banks are displayed
- Tagged events are not displayed in the Audio Bin Browser Search Bar



Tag Search

Note

Because of the nature of how stereo sound files work it is important to understand the consequences of panning channels throughout the signal path. A stereo sound file contains information that provides the special effect of hearing things to the left and rights and the illusion of distance. Mixing a stereo file down to mono removes this information permanently, so even if you mix a mono file back up to "stereo" or pan it from 100% either left or right, back to the middle, the stereo effect is lost. You end up with the same mono signal duplicated in the left and right channels, not a stereo signal.

Routing and Assigning

FMOD Studio has several Browser interface areas. They all look very similar but there are two different ways in which they can function. The Event Browser allows you to assign events to Folders, events to tags, and events to Banks. The Mixer Browser allows you to Route events to Group Buses and to perform other Routing functions. Routing and Assigning have very different consequences and it is important that these are understood.

When you assign an object in FMOD Studio you are creating an association between that object and another object. By assigning an event to a Folder you are allocating the name for the event with the Folder in which you are placing it. The event itself does not alter in any way and the act of assigning is more administrative in its effect. Objects can be assigned and reassigned without altering their function or how they operate within a project.

In the Mixer Browser the action is called Routing because the association between objects physically alters the signal path within FMOD Studio and can have significant effects on the output of a project. Routing an event into a Group Bus actually changes the signal path and can result in the sound altering depending on the setup of the Group Bus. Rerouting elsewhere can result in further changes.

Routing needs to be planned carefully with full realization of where the signal path travels and what processes are applied to it, otherwise unexpected playback behavior may occur.

The Mixer

The Mixer functionality in FMOD Studio brings it in line with many common hardware and software Digital Audio Workstations. You can create and route signal paths and create complex signal networks.

The Mixer in FMOD Studio is the only screen not displayed by default. To access the Mixer go to the main menu and under the **Window** menu select **Mixer**.



The Mixer

The Mixer can be as simple or complex as a project requires. By default Studio routs all signals out the master channel as simple projects do not require the Mixer. For more complex projects the Mixer can control large numbers of events assigned to Group Buses with complex routing paths, effects chains and automation. Because of the virtual nature of FMOD Studio it can create far more complex signal paths than even the largest of hardware devices.



Mixer Interface

In Depth: The Mixer Functionality

The Mixer tool set in FMOD Studio has been designed to allow a broad range of level access. At its most basic level you can create and implement simple sound environments with no need at all to

interact with the Mixer interface. By default Studio rounts signals out of the Master bus exactly as they have been setup to play in the projects events. In this example there is no requirement to make use of the Mixer in a project.

The other extreme is the ability to create complex audio events routed through practically limitless signal paths passing through a wide range of effects and other signal processors with complex mixing and manipulation control groups.

The level to which you take advantage of the Mixer functionality is entirely up to the requirements of a project and their familiarity with creating signal paths.

The Mixer Tool Set

This chapter covers exactly how the Mixer tool set in FMOD Studio functions and provides you with an understanding of the basic signal path flow through Studio. Tutorials provide practical steps towards creating various types of signal paths within FMOD Studio.

Multiple Mixer view

FMOD Studio supports multiple Mixer views. This means it is possible generate more than one Mixer window and assign specific Mixer objects to the desired Mixer View. This can allow for viewing of several Mixer view Windows across multiple monitors and can be very convenient for mixing larger projects.

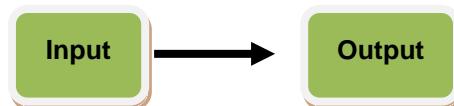
To assign an object to a new Mixer View, right click on the object in the Mixer Browser and select Assign to New Mixer View from the context-sensitive menu.

Group Solo

When editing groups of Channel Strips in the Mixer activating the Solo function solos the selected group for playback. This can be reverted by again clicking on the solo button to switch off the solo for the group.

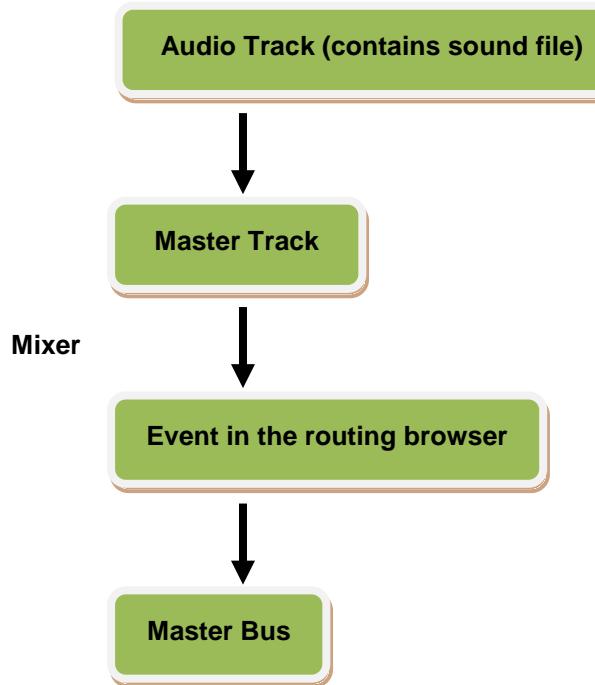
Basic Signal Paths

A signal path at its most basic requires an input stage and an output stage. A Sound-File added to an FMOD Studio project is the input or starting stage of a basic signal path. The sound that is played out from Studio is the output of final stage of the signal path.



Within FMOD Studio a sound file must be added to an event to be audible. The sound file is added to an Audio Track within the event (By default Audio Track 1). All events also contain a Master Track.

With the exception of nested events, every event in the Events browser is also present in the Routing browser. By default all events in the routing browser are routed directly into of the Master Bus. As a result the most basic signal path produced in FMOD Studio is as follows.

Event**Event 01**

This is the path a sound takes from initial input to ultimate output.

This basic signal path is the core of every sound that can be created within FMOD Studio. Even the most complex signal routing set-ups still have this basic path at their core. Without the chain from input to master output, no sound is produced.

How this looks in Studio:



Basic Signal Path Diagram 1

At the event level,

1. The Audio Track contains a Module containing a Sound File
2. The Master Track is where all Audio Tracks in an event are sent prior to routing to the Mixer.



Basic Signal Path Diagram 2

A Word on Effects

Effects are objects inserted into the path of a signal. The extent to which they affect the signal is entirely up to the values defined for each Effect object.

Effects can be added to signal paths at many stages along the route of the signal. Effects are added to the Deck regions and can be added to an individual Audio Track, to the Master Track of an event or to a Group Bus in the Mixer. It is possible to have the same Effect object inserted into the signal path of a single sound in all three of these stages and for each Effect to be controlled individually and uniquely.

Effects are discussed in more detail in a later chapter, but it is important to understand their setting as one of the main reasons for creating alterations to a signal path is to insert an effect into the path to manipulate the output sound. As such they are used as examples in explaining the Properties of the Mixer tool-set.

Important Note

FMOD Studio can create complex signal paths and there are a great variety of functions that let you manipulate signals and sounds.

Do not feel you need to try and combine all of these features into projects. The complex possibilities that the FMOD Studio tool set provide for are more about providing a wide range of options rather than trying to overcomplicate the creative sound design process.

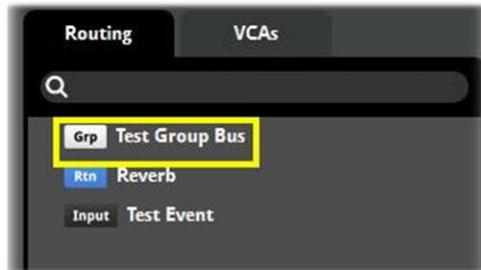
The ability to insert Effects into multiple locations within a signal path is the perfect example. You are rarely, if ever likely to need to create a project that includes the same Effects object multiple times within a single signal path. However some Events work better if the Effect is on an individual Audio Track, while others work best if the Effect is on the Event Master Track and still others might need it to be a part of a Group Bus in the Mixer tool.

The functions provided within FMOD Studio may seem complex and overwhelming if you attempt to apply all of them to all of your projects all of the time. The tools exist to provide options to the Sound Designer. Do not overcomplicate things for yourself.

Group Buses

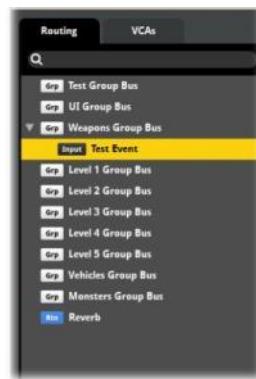
FMOD Studio populates the Mixer Browser with all the events, other than nested events, that exist in a project. By default, these events are all routed directly to the Master Bus for playback output. This means that FMOD Studio plays all of the events included within a project without the need to access the Mixer at all.

If a project requires a further level of control over the output behavior of the signal paths then they can be assigned to specific groups within the Mixer View; these groups are called Group Buses.



A Group Bus

A Group Bus mixes the output of multiple events and buses together, allowing you to apply effects and routing behavior to all the signals within a group.



The Mixer Browser

Any event that is not nested and does not route into a Group Bus instead routes into the Master Bus. Each event must route into exactly one group bus or master bus.

Why Group Buses?

FMOD Studio provides a wide range of options for creating, editing and optimizing both individual signals paths and the overall signal chain for a project. Effects Modules can be inserted into the signal path of a sound in various locations to control and alter the final output sound. FMOD Studio is a complex application and there are several best practice processes that help you get the best results. One of these involves the use of Group Buses.

Keep Things Simple

Keeping your signal chain as simple as possible helps you in some key ways. The first and most important is simply for your own sanity. While it might seem straight forward to add an effect directly into an audio track or master track of an event while you are creating it, this could lead to issues later on in development.

With any large project, it is best to have a well thought out and planned structure for the audio required. This plan should include the types of effects and processes you are likely to need to achieve the end result. Working from such a plan allows you to think forward and work efficiently. If you require a particular type of effect for several sound events then the most direct way to apply that effect is to add it to a single group or return bus and route all events that are relevant through this single bus. This means there is only one type of this effect to keep track of and control. Should anything need to be altered or edited you know exactly where to go rather than having to search through each event trying to find where an issue may lie.

This process is also very efficient from a resource point of view as there is only a single Effect Module requiring resources to operate rather than multiple modules applied individually to each event.

Creating a Group Bus

All events in the Mixer View route into the Master Bus by default. Group Buses provide a further level of control over signal paths in FMOD Studio.

This tutorial covers the steps in to create a Group Bus in the Mixer View.

Step 1 Display the Mixer View



Selecting the Mixer

1. Within the default FMOD Studio view go to the Main Menu – Window
2. Select Mixer

Step 2 Create a Group Bus

1. Within the Routing Browser right click to display the context-sensitive menu
2. Select New Master Group Bus
3. Type in the desired name of the new Group Bus

FMOD Studio creates a new Group Bus and adds it to the Mixer Browser. This Group Bus can now have events and buses routed into it as required.

Routing Events into Group Buses

This tutorial demonstrates how to route Events to Buses.

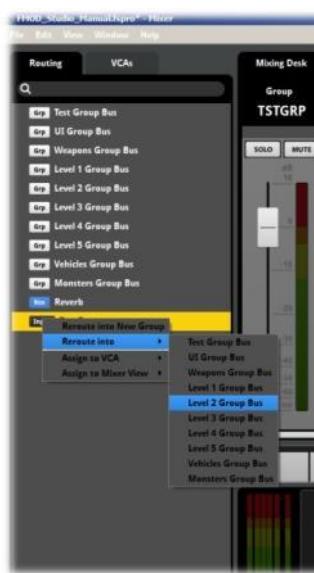
Step 1 Open a Project with existing Events and Group Buses



The Mixer

1. Select a Project that contains both events and Group Buses
2. Select Mixer from the Main Menu – Window context-sensitive menu

Step 2 Route an Event into an existing Group Bus

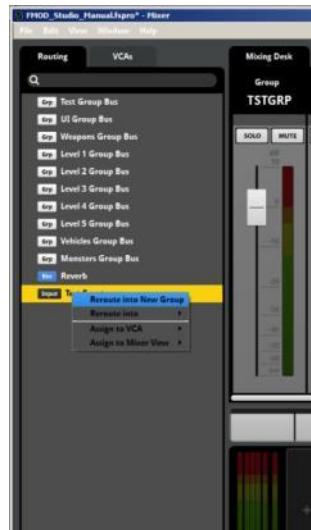


The Mixer Browser

1. Within the Mixer Browser right click on an event to display the context menu.
2. Select Reroute into and navigate via the arrow to the list of existing Group Buses.
3. Select the Group Bus you wish to assign the event to.

FMOD Studio reroutes the selected event to the desired Group Bus.

Step 3 Rerouting an event to a New Group Bus



FMOD Studio Main view

1. Within the Mixer Browser right click on an event to display the context menu.
2. Select Reroute into new Group Bus.
3. Define the name for the new Group Bus.

FMOD Studio routes the selected event to the newly created Group Bus. In the Routing browser, the event changes its position to reflect this rerouting.

Step 3 Route an event to the Master Bus



The Mixer Browser

1. Within the routing Browser right click on an event that is routed to a Group Bus.
2. Select Reroute to Master Bus.

This command can be used to route an event to the Master Bus if you no longer wish for it to be routed to a group bus.

Signal Routing

It is important to understand some of the implications of signal routing and what the process entails.

A Send Module creates a duplicate of the signal path and routes it to a Return Bus in the Mixer. For many routing situations this is the best step to apply. But it can create adverse effects if used in the wrong circumstance. For example duplicating a signal via a Send Module and then reducing the original signal by 6dB and then amplifying the Send Duplicate by 6dB results in the two signals cancelling each other out and no noticeable effect being produced. This is a very basic example of how a duplicate signal could cause an issue. The physics of sound waves means that applying a band pass filter of any type to a Send and combining it with the original signal could have very different results when combined than either of the two signals in isolation.

Certain Effects work very well when included as part of a Send / Return signal path. Reverb, Chorus and Delay as well as some Flanger Effects work well because their functionality involves the mixing of the original signal with a subsequent versions anyway. So a Send Duplicate signal can become part of this mixing process to produce the desired result.

Other types of Effect Module such as Distortion and filters are altering the signal itself. Combining them with a duplicate at best dilutes the alteration effect or at worst alters the signal by combining the material in a manner difficult to predict or control. For this reason we recommend that Effects that alter a signal should be inserted directly into the signal chain rather than as part of a Send Return routing step.

Deck Order

The Deck area is not only part of the signal chain process, each Deck is an extension of each individual signal path. This means the order in which Modules are placed into the Deck has a significant impact.

Modules can be placed in either the Pre Fade or Post Fade position, but also a signal is processed in order from left to right as it passes through the Deck area. If a Send Module is inserted into the Deck Area between two other Modules the duplicate signal is created at the exact point of the Send.

Planning the order and layout of Deck areas is an integral part of planning your overall signal chain.

Working with Group Buses

A new Channel Strip

Once you create a Group Bus a new Channel strip is added to the Mixer View to control the output of the Group Bus. This channel strip includes the standard controls for Mute, Solo and a fader slider for output levels. There is also a Peak meter to allow you to visually monitor the levels for the Group Bus's output levels.



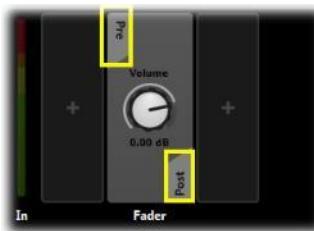
Group Bus Channel Strip

The Deck

Each Group Bus also has its own Deck region. By clicking on the Channel strip for any Group Bus you make it the active selection. The Target Channel Strip is outlined in yellow to indicate it is active and the Deck region for that Bus is now active and can be populated with effects as required.

By default the Deck region includes input and output peak meters at the left and right extremes of the deck as well as a Fader control dial. The fader control dial is bookended by markers for pre and post effect placement. By right clicking on these markers you can insert effects objects, but define whether you want them to be inserted in the signal path before the Fader (pre) or after the Fader (post) in the signal path.

There are a great many reasons as to why an effect object may be needed either pre or post fader in a signal path. The exact needs of your project often dictate the best course of action, but it is worth experimenting with your various options so you understand how each may affect the playback behavior of your sounds.



Deck Region Pre and Post Fader Positioning

Buses within Buses

As a Group Bus is essentially a way in which to mix the outputs of other events and buses together so that they can be manipulated as one signal. Studio allows for Group Buses to be created within existing Group Buses.

Creating Group Buses

Group Buses can be created inside existing group buses to provide finer control of a single path. It is possible to create an almost endless chain of Group Buses.

This tutorial takes you through the steps to create a group bus that routes into another group bus.

Step 1 Open a Project with existing Events and Group Buses



FMOD Studio Mixer view

1. Select a Project that contains both events and Group Buses
2. Select Mixer from the Main Menu – Window context-sensitive menu

Step 2 Create a Group Bus within an existing bus



Create a New Group Bus

1. Select a Group Bus from the Mixer Browser, Right Click on the Icon
2. Select New Group Bus from the context-sensitive menu
3. Type in the name for the new Group Bus



Create New Sub Group Bus

Studio creates the New Group Bus, but it is displayed in the Mixer Browser inset and nested inside the Bus it was created in. This means that all effects, panning or levels applied to the parent bus are also applied to the nested bus as well as any effects or Properties it may include.

Note

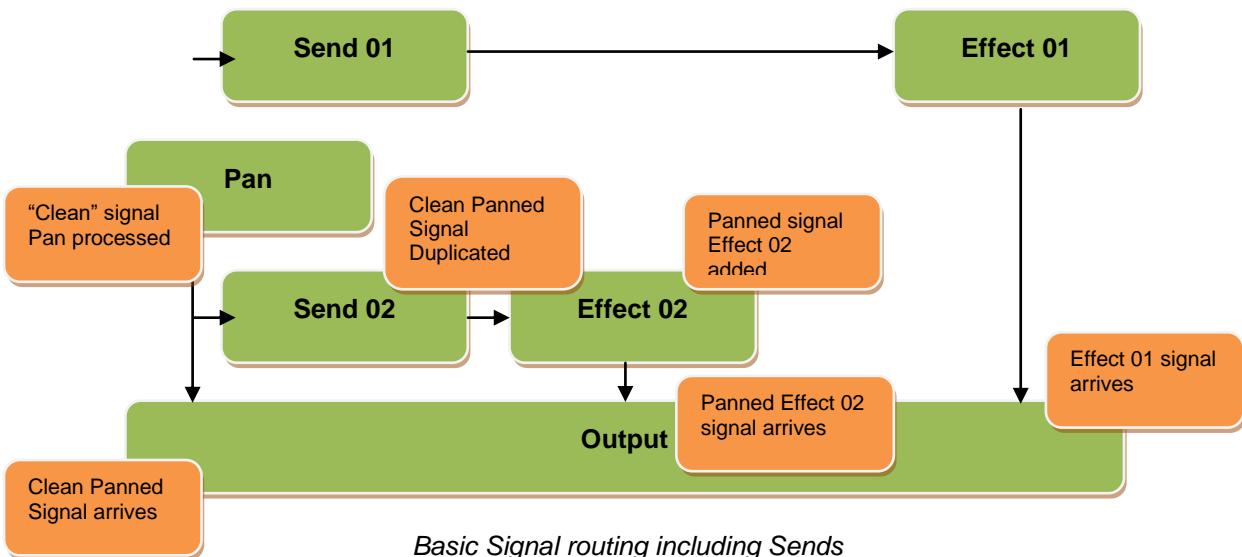
Because every level of nested bus within a parent bus also adopts all of its behavior, it can be easy to lose control of your signal path if you bury it too deeply within nested buses.

Nested buses can be a very powerful method of controlling playback behavior, but take careful note of what you apply to each level of the hierarchy to avoid confusion at later stages of development.

Sends

A Send is a way of duplicating a signal path at the exact point at which it is inserted in the path. Sends can be added to Audio Tracks within events and Buses within the Mixer. A signal path can contain multiple Sends. The primary function of a Send is to allow a copy of a signal to be created so it can be processed and manipulated while the original signal continues unaltered.





Basic Send routing in Events



Creating a Send in an Event

Basic Send routing in Mixer



Creating a Send in the Mixer

Adding Sends to Events

Sends within events allow for duplicate signals from Audio Tracks to be created allowing manipulation of the signal while still preserving the original track path to the Master Track

This tutorial takes you through the steps to create a Send within an event and route Audio Tracks through an event Send.

Step 1 Creating a Send



Creating a New Send

1. Within an existing event select and Audio Track by clicking on it
2. Once an Audio Track is active you can access its Deck region, right click in the Deck region to display the context-sensitive menu.
3. From the context-sensitive menu select Add Send

Studio creates a new type of Track within the event; this is a Return Track. Return tracks are different from Audio Tracks, they cannot contain Modules within the track, but they have an active Deck area where Effects objects can be added. This allows for Tracks routed to the return track to have the full range of Effects applied to them, but the original Audio Track that was routed via the Send is preserved.

Note

Return Tracks exist in the signal chain just like any other Audio Track. By default Return Tracks are routed directly to the Event Master Track, but they can be mixed to other tracks prior to the Master. Ultimately all tracks within an Event route to the Master Track including Return Tracks.

Send/Return Functionality

A Send object creates a duplicate signal at the point in the signal chain where the Send is inserted. It has a simple setting and control set.

When a Send object is created in the Deck region of either an event or a Mixer Channel Strip it creates an object that includes a level control. The signal sent to the associated Return Track or

Return Bus is a duplicate of the original signal in all ways. The Send does however include a volume control to be able to adjust the level of this duplicate signal as a function of input.



A Send

The Send object is essentially the input source of the new duplicate signal and can have its levels controlled via the volume dial. A “sent” signal requires a destination at which it can be controlled by the FMOD Studio Properties. This is the purpose and function of Return Tracks and Return Buses.

Return objects provide a new location for the signal path where it can be processed and manipulated. It is possible to send signals from many sources to a single Return object where they can all be processed in a same manner. Any object that includes a Send object mapped to the relevant Return object has a duplicate signal routed to that Return.

Return objects must ultimately be routed out the master channel. Either the Master Track in the case of Return Tracks in events, or the Master Bus for Return Buses in the Mixer. Return objects can be routed to other tracks and buses instead of directly to the Master channels by using Send objects in the same manner as in regular tracks and buses.

Routing Audio Tracks to Return Tracks

Once a Return Track has been created in an event Audio Track signals can be routed to the Return Track using Send objects. This tutorial goes through the steps to create a Send within an event that route to an existing Return Track.

Step 1 Creating a Send



Signal Routing via a Send

1. Within an existing event select and Audio Track by clicking on it
2. Once an Audio Track is active you can access its Deck region, right click in the Deck region to display the context-sensitive menu.

- From the context-sensitive menu select Add Send and via the arrow select an existing Return destination.

There is always an option within the menu Add Send setting to create a New Return, however all existing Returns are also displayed in this menu space. This allows you to select from the existing list of Returns or to create new Returns as required.

Note

While creating a new Send object in the Deck is part of creating the overall signal chain it is different to routing within the Event. In the drop-down menu of a Track Header you can mix an Audio Track to either the Master Track or to one of the other existing Audio Tracks.

To “Route” an Audio Track signal to a Return Track it must be done by creating a Send object in the Deck region of the Audio Track in question. This is because Event signal routing is directing the original signal of an Audio Track within and ultimately to outside of the Event. A Return Track can only receive signals from a Send object which is always a duplicate of a track’s signal.

While this is a subtle difference that in many projects may not even be noticeable it is a very important distinction to understand when working with the signal chain.

Adding Sends in the Mixer

Sends can be inserted into any Mixer function that has a Deck region. This includes Group Buses, Sub Group Buses, the Reverb Bus and Return Buses. It is also possible to add a Send object into the Deck region of the Master Bus. Sends can also be created that route a signal to the Reverb Bus. By doing so you create a copy of the signal rather than routing the original signal through the reverb.

Adding a Send

The process for adding a Send is consistent across all Mixer functions that include Deck regions. This tutorial goes through the steps to create a Send within a Mixer Deck region.

Step 1 Creating a Send



FMOD Studio Main view

1. Within the Mixer click on any channel strip to select it and make it active
2. Once the Channel is active and the Deck region for the channel is displayed, Right Click in the Deck region to display the context-sensitive menu
3. From the context-sensitive menu select Add Send

Studio creates a new type of bus within the Mixer Browser; this is a Return Bus. Events and buses cannot be routed into Return Buses, but return buses do have signal chains, and can receive signals from sends. This allows a return bus to apply effects to its signal, without affecting the signal in the buses that contain the sends.

Note

Return Buses exist in the signal chain just like any other Bus. By default Return Buses are routed directly to the Project Master bus, but they can be sent to other Group Buses and Return Buses prior to the Master. Ultimately all buses within a Project route to the Master Bus including Return Buses.

Routing Buses to existing Return Buses

Once a Return Bus has been created in a Group Bus signal can be routed to the Return Track using Send objects. This tutorial goes through the steps to create a Send within a Bus that routes to an existing Return Bus.

Step 1 Creating a Send



Routing to an existing Return Bus

1. Within an existing Mix select a Channel Strip by clicking on it
2. Once a Channel Strip is active you can access its Deck region, right click in the Deck region to display the context-sensitive menu.
3. From the context-sensitive menu select Add Send and via the arrow select an existing Return destination.

There is always an option within the menu Add Send setting to create a New Return, however all existing Returns are also displayed in this menu space. This allows you to select from the existing list of Returns or to create new Returns as required.

Routing at the Event Level

The signal path a sound follows through FMOD Studio starts from the sound file added to the Audio Track of an event and ends at the Master Bus in the Mixer where it reaches the playback stage. The Mixers primary purpose is to create and control signal routing within a Project, however there are some options and Properties available within the event editor that also provide signal routing possibilities.

Routing to tracks

The simplest event in FMOD Studio consists of a single Audio Track and the event Master Track. An event can have extra Audio Tracks added as required and Audio Tracks can contain a wide range of Effects and automation controls as well as Parameter controls. By default all Audio Tracks are routed to the event Master Track, but it is possible to route any Audio Track to pass through another Audio Track prior to it being sent to the event Master Track. This allows for sub mixing of several tracks into a new track prior to being sent to the event Master Track. This is similar to the functionality of the Mixer and Group Buses but at the event level.

Routing to Audio Tracks

Audio Tracks can be routed to other Audio Tracks within an event prior to routing to the event Master Track. This tutorial goes through the steps to route an Audio Track to another Audio Track.

Step 1 Create an new Event



Creating a New Audio Track

1. Create a new event in the Event Browser
2. Within the Event Editor right click on the Track Header region
3. Select Add Audio Track from the context-sensitive menu
4. Add a total of three new Audio Tracks to the event

The event should now have a total of four Audio Tracks. At this stage none of the tracks contain any sound files, but for this tutorial they are not needed. Audio Tracks 2 and 3 are now routed into Audio Track 4.

Step 2 Assigning Audio Tracks to Audio Tracks

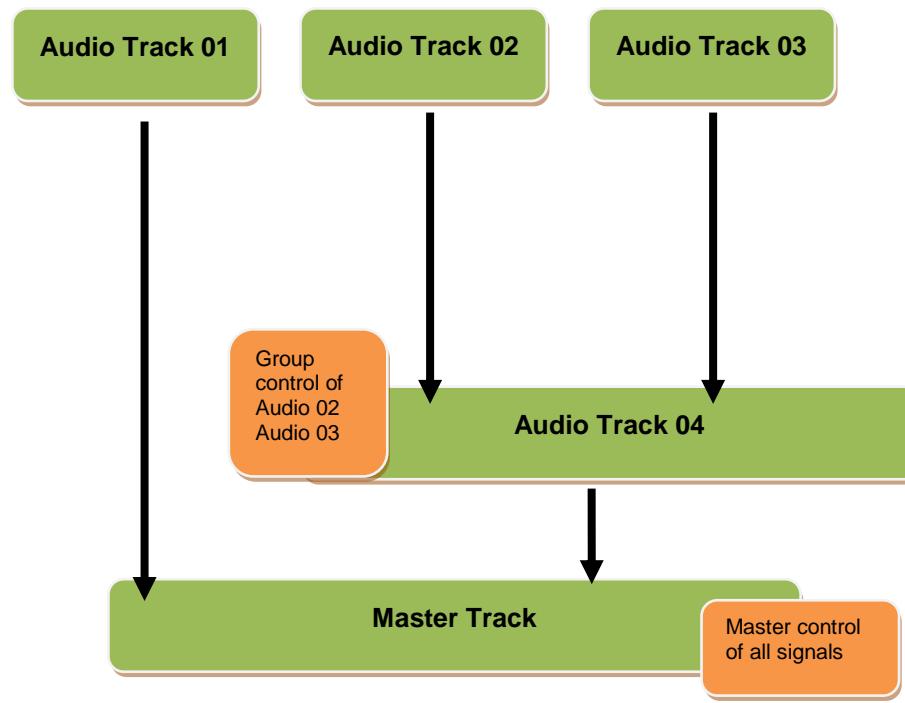


FMOD Studio Main view

1. Right Click on the Track header of Audio Track 2
2. Select Output and via the arrow select Audio 4
3. Right Click on the Track header of Audio Track 3
4. Select Output and via the arrow select Audio 4

The following routing paths have now been established internally within the new event.

Signal Routing within an Event



This diagram is very similar to the basic routing diagram within the Mixer. It demonstrates that signal path control can start from within each event prior to the signal reaching the Mixer. The requirements of each project need to be assessed to determine the best path layout, event level routing simply provides more control.

Note

Sends introduce a delay of one audio block to the sent signal compared to the original. This means that the exact amount of delay introduced by a send varies by platform. While the amount of delay is generally so infinitesimal as to not cause any problems, a signal that passes through a very large number of sends (such as any signal that passes through a signal loop) may be noticeably affected.

VCAs

Voltage Control Amplifiers (VCAs) are Mixer objects that can be applied to the signal chain within FMOD Studio projects. VCAs are an efficient way of grouping sounds for the purposes of volume control and exist in parallel with the normal signal chain.

Implementation

VCAs are created and exist in a unique Tab area within the Mixer view. Within the VCA browser there is a flat hierarchy; all VCAs exist on the same level. VCAs are also not part of the routing chain, they exist alongside it.

VCAs provide a further level of volume control designed to allow for groups of sounds to be volume adjusted relatively across the group. The most common use situation for VCAs is as master controls within a game project main menu setup. Sound Effects, Dialogue and Music are excellent examples of typical VCA grouping in a game project.

VCAs can also be very useful for general project balancing and level mixing as they provide a quick way of grouping and adjust sound events without affecting the signal chain in any way.

Events, Group Buses, Return Buses and the Reverb Bus can all be assigned to VCAs. The term Assigned is specifically relevant because signals are not routed through VCAs they are assigned as parallel controls of the main signal chain.

VCA Controls

All VCAs include a standard channel strip controller, volume fader, solo and mute controls.



VCA Channel Strip

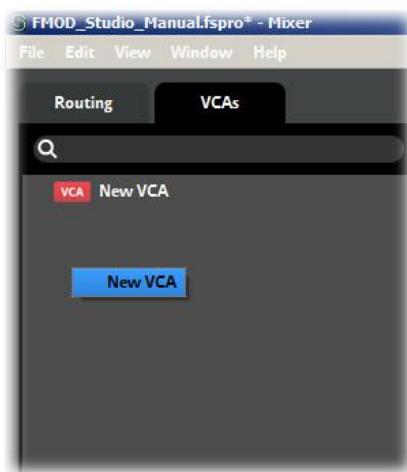
Note

Because VCAs exist outside of the signal chain they allow for objects to be grouped together in any way. Within a VCA there is no relationship other than the VCA group so objects from different Group Buses, Returns and events can all be grouped and controlled without fear of interfering with their routing pathways.

Creating VCAs

VCAs are created within the VCA Browser in the Mixer view and can have events, Group Buses, Return Buses and the Reverb Buses assigned to them. This tutorial goes through the steps to create a VCA.

Step 1 Create a new VCA



Creating a new VCA

1. Navigate to the Mixer view
2. Within the Mixer View hit the VCA tab in the Browser region to display the VCA Browser
3. Within the VCA Browser Right Click to display the context-sensitive menu
4. Select Add VCA from the context-sensitive menu
5. Type in the desired name for the new VCA, hit Enter or click away from the VCA to confirm

The new VCA with the defined name is displayed in the VCA Browser and a new corresponding Channel Strip is created in the Mixer region. Events, Group Buses, Returns and the Reverb Bus can all be assigned to VCAs.

Assigning to VCAs

Mixer objects can be assigned to VCAs from the Routing Browser. Objects can also be assigned to a new VCA, which creates a *new* VCA as part of the process.

This tutorial goes through the steps to assign Mixer objects to a VCA.

Step 1 Assigning to VCAs



Assigning to an existing VCA

1. Navigate to a Mixer view that contains events or Group Buses
2. Within The Router Browser Right Click on an object to access the context-sensitive menu
3. Select Assign to VCA and navigate through the arrow to the selection menu
4. Select an existing VCA or Add VCA from the selection menu
5. If a new VCA is selected, type in the desired name of the new VCA

Any Mixer object assigned to a VCA appears in the VCA Browser under the appropriate VCA
Adjusting the levels of a VCA relatively adjusts the volume levels of all objects assigned to that VCA.

Multiple Mixer Views

You can create multiple Mixer views that are all live simultaneously. This is a useful tool for balancing a project by section, level, sound types or any other desirable grouping that can assist in improving the efficiency of production.

Using Mixer Views

A Mixer view is simple a visual representation of a selection of Mixer Channels. The choice and inclusion of a Channel in a view does not alter how a project works, it is simply a way to view specific Channel strips while obscuring others. You can assign any Mixer Object to a Mixer view. In this way you could have one mixer view that contains all the character sounds, one of adversary sounds and a third for environmental sounds. A fourth view could then include all of the above.

With multiple monitors it is possible to view and work with multiple views simultaneously. Obviously adjusting any single specific channel strip results in all versions of that Channel that exist in the various views changing in the same way.



Multiple Mixer Views

Note

Mixer views are not instances. They are simply a different view of the same information. Each view of a specific channel and its information is identical and adjustment to any values is reflected across all views of the same channel.

Administering Mixer Views

The Mixer Region includes a Tabs area that displays all Mixer views included in a project. You can switch between Mixer views by selecting the relevant tabs. Right clicking in the tabs region provides options for:

- Creating new Mixer Views
- Renaming Mixer Views
- Opening Mixer Views in a new window
- Deleting Mixer views

Bus Naming

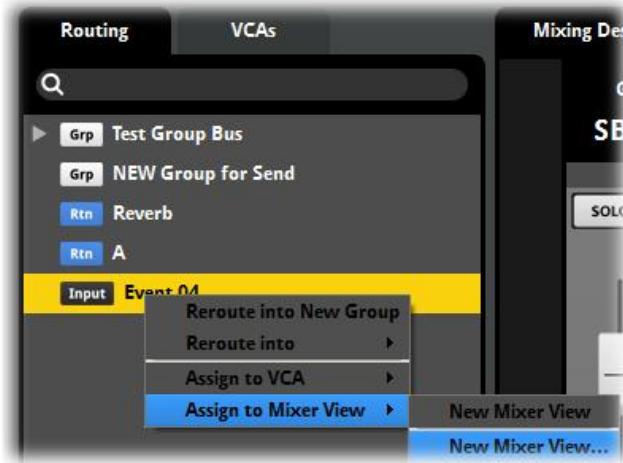
The Mixer in FMOD Studio allows you to rename Buses in multiple ways. A Bus can be renamed in the Mixer Browser simply by double clicking on the Bus name within the Browser, or by selecting it and pressing the 'F2' key. This activates the text edit mode for the name and the new name can be directly typed in. Once the name is complete hitting enter commits the change.

Another way to rename a Bus is by double clicking on the Bus name on a Channel Strip. This opens a text field for that Strip and the new name can be directly typed in. Hitting Enter commits the new name.

Assigning to Mixer Views

All Mixer objects including VCAs, Group Buses, events, Return Buses and the Reverb Bus can all be assigned to Mixer views to assist in production workflow. This tutorial goes through the steps to assign Mixer objects to Mixer views.

Step 1 Assigning to Mixer views



Assigning to a Mixer View

1. Navigate to a Mixer view that contains events, Group Buses or VCAs
2. Within The Router Browser Right Click on an object to access the context-sensitive menu
3. Select Assign to Mixer View and navigate through the arrow to the selection menu
4. Select an existing Mixer View or Add Mixer View from the selection menu
5. If a new Mixer View is selected a type in the desire name for the new Mixer view

Mixer Objects are assigned as defined to the selected Mixer View. All Mixer Views are listed in Tab form in the Mixer region and can be switched between by selecting the relevant Tab.

Snapshots

Snapshots are a method of changing Module and Bus Properties in FMOD Studio's Mixer. Specifically a Snapshot stores specific values for any applicable properties you specify (scope in), and can override usual values when required. You can create transitions between various snapshots to change the values of Mixer Properties.

When used individually, FMOD Studio transitions all current Property values within the Mixer from current values to the Snapshot values. When multiple snapshots are used, Studio can transition between Snapshots.

Snapshots are actioned in a hierarchy from the top down. You can organize these in Snapshot tab on the left of the Mixer window.



Snapshots Tab

Mixer Snapshots

Unlike event Properties, Mixer Properties do not normally have the facility to add automation. The solution to this problem is to use Mixer Snapshots. A Mixer Snapshot is not, as the name suggests, a capture of all the Property values within the Mixer at the time when the Snapshot is created. Instead, it is a collection of values for a specific list of 'scoped-in' properties that you specify. New properties can be scoped in and out of an existing snapshot at any time, and their property values can likewise be adjusted.

A Snapshot can be used individually or in combination with other snapshots. When used individually, FMOD Studio sets the values of all the properties scoped into the snapshot to the values specified in that snapshot, and properties not scoped into the snapshot are completely unaffected. When two or more snapshots are active at once, Studio simply applies them both – and if a property scoped into one snapshot is also scoped into the second snapshot, Studio determines which property value takes precedence based on the order in which the snapshots appear in the Snapshots Browser, with snapshots listed higher up having a higher priority.

The Intensity Property (visible in the deck when selecting a Snapshot Trigger Region in the Event Editor) allows you to specify a "mix" between the property values specified in a snapshot and those that would be in place were the snapshot not active.

As the Intensity Property is increased, the closer the properties' values are shifted to those of the snapshot. So an Intensity value of 0% results in none of the Snapshot being applied, whereas a value of 50% is an equal mix of both the current values and the Snapshot values.

The Snapshot Intensity Property multiplies with the intensity applied by Snapshot Trigger Regions when the Snapshot is applied.

Note

A Snapshot can be as simple or complex as a project requires. A Snapshot can change a single property value, or it can be used to make changes to multiple property values across a project.

There are two different kinds of Snapshots in FMOD Studio; Overriding and Blending.

Overriding Snapshots

Overriding snapshots act like masks. When active, an Overriding Snapshot 'overrides' the values of the properties that they include with the values stored in the snapshot. If the snapshot's intensity is less than 100%, this means that the snapshot only changes the value of the property partway to the value specified in the snapshot. When more than one overriding snapshot applies to the same property, they are applied in order, with the highest-priority snapshot being applied last (meaning it has 'the last word' on the values of the properties it affects).

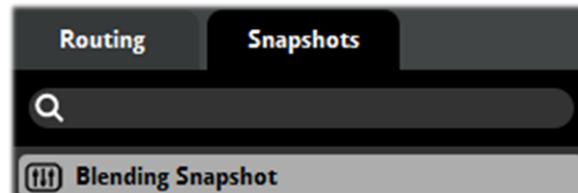


Overriding Snapshots are identified by a star icon

Blending Snapshots

Blending Snapshots operate in the same way as Overriding Snapshots for all property types except for volume. In the case of volume, Blending Snapshots combine the values. For example, if you have

a Bus with more than one Snapshot active and adjusting volume, it adds the values together. So, if both Snapshots specify a volume of 3db, the total is 6db. The Snapshot hierarchy is set from the top down and can be reorganized.



Blending Snapshots are identified by a slider icon

Creating a Snapshot

The first stage in working with snapshots is to create the snapshot.

1. Go to the Mixer.
2. Go to the Snapshots Tab.
3. Right click on the Snapshots Browser and select New Snapshot or New Blank Snapshot.



Create a new Snapshot

Once a snapshot has been created you can select which Properties you wish it to have control over. While a snapshot is selected, the context menu for bus and bus effect module properties includes a Scope In command. All Properties that have been Scoped In are included in the Snapshot, and can be assigned values to be applied when the snapshot is active. If a Property is not scoped in (is 'scoped out'), the user interface changes to reflect this, so at a glance it is very easy to see which Properties are scoped in to the selected snapshot.



Scoping In a Property

Once a Snapshot has been created and defined in the Mixer it can be activated and controlled from within an event. To do this a special Trigger Region object is created in an Audio Track or the Master Track of an event. There are two methods of creating a Snapshot Trigger Region.

Method 1

Open both the Mixer and the Event Editor and drag the desired Snapshot from the Snapshot area onto an Audio Track or the Master Track of the target event. This causes FMOD Studio to create a new Trigger Region that can be resized, repositioned, renamed and have its color set in the same manner as any other Trigger Region. When a Snapshot's trigger region is selected, the associated Snapshot Module appears in the deck, where the Intensity Property and Parameters appear as controls. These properties can be automated and modulated within an event.



Creating a Snapshot Trigger Region

Method 2

Right click within an event Master Track and select Add Snapshot from the context sensitive menu. You can either create a new Snapshot for the trigger region or create a Trigger Region for an existing Snapshot. Adding a new Snapshot creates a new trigger region on the Master Track, and associated Snapshot Module, and a corresponding new Snapshot in the Snapshot Browser in the Mixer. Creating a Trigger Region for an existing Snapshot only creates the Trigger Region and Module.

Note

A Triggered Snapshot Module is an instance. This means there can be multiple Trigger Regions for the same Snapshot across various Events, or even to have multiple instances of a Snapshot triggered by multiple instances of a single Event. It is also possible to rename and adjust the properties of Snapshot Modules independently of each other and of the Snapshot. Because of this we recommend you name your Snapshots and associated Trigger Regions very carefully to avoid confusion.

When multiple instances of a snapshot are simultaneously active, their intensities are averaged to determine the intensity with which the snapshot is applied to the mix.

Working with Snapshots

A Snapshot is “controlled” via the Intensity Property in the related Snapshot Trigger Region within an event. There are various ways in which a Snapshot can be controlled within an event.

A Snapshot Trigger Region functions just like any other Trigger Region. It activates when the cursor in an event makes contact with the Trigger Region on either the Timeline or a Parameter. If there is no automation or modulation this behaves in a very basic ON/Off method just like switching on a light.



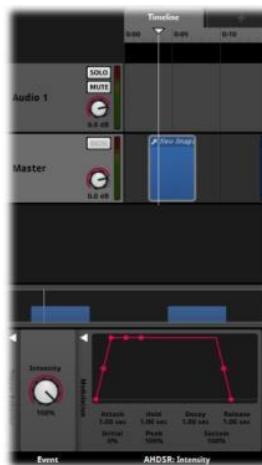
Triggering a Snapshot

The values of properties scoped into a Snapshot can be edited by selecting that Snapshot in the Snapshots browser, and adjusting the appropriate property where it appears in the Mixer. This does not edit the value of that property outside of the context of the selected snapshot.

Snapshots also have an internal intensity that is blended with the instance-specific intensity to determine the final value. This property can be viewed in the Deck by selecting a Snapshot in the Snapshots browser.

Modulation

Like all Property Dials in the Deck area you can add Modulators to the Intensity Property of a Snapshot Trigger Region. This allows you to create a “fade-in” and “fade-out” behavior for the Snapshot Trigger.



Snapshot Modulation

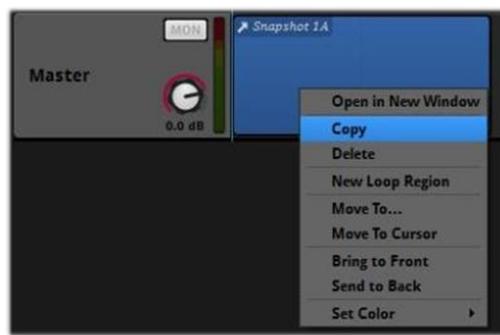
Automation

Automation can also be added to the Snapshot Intensity Property. This allows for precise control of the Snapshot intensity over the course of time on the Timeline or other values via a Parameter. Automation for Snapshots is created in the same method as other Property Automation by creating and adjusting nodes and curves.

*Snapshot Automation*

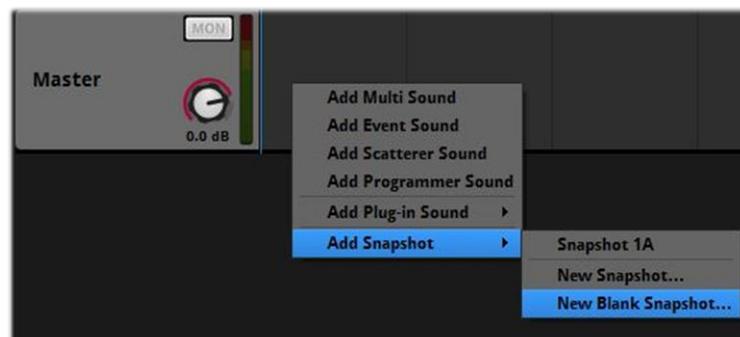
Copying Snapshots

Snapshots and Snapshot Trigger Regions can be copied and pasted. Copying and pasting a Snapshot Trigger Region produces a copy of that Snapshot Trigger Region and Module. This means both the original and the copy reference the same Snapshot. Copying and pasting a Snapshot in the Snapshot Browser creates a new Snapshot with the same settings as the old Snapshot. In such a case, existing Snapshot Trigger Regions continue to trigger the original Snapshot.

*Copying a Snapshot Trigger Region*

Blank Snapshots

The context menu provides an option to create blank Snapshots. This menu option creates a Snapshot that does not have any Properties Scoped In. In contrast, the New Snapshot menu option automatically Scopes In all the Group and Return buses in the Project. Aside from their initial scope, there is no difference between these two kinds of snapshots, and either allows you to Scope In and Scope Out properties as desired.

*Creating a Blank Snapshot*

Snapshots and Stacking

Because it is possible to have multiple instances of a snapshot controlled across several events with multiple Trigger Regions it is possible to stack the effects of the Snapshot Intensity Property. For example, the way a gunshot that has a Snapshot created to reduce the volume by 80% sounds depends on both the level of intensity and the number of instances being triggered of the Snapshot.

A single instance set to 100% Intensity applies 100% of the change, which adjusts the output by 80%, causing the sound to drop considerably. A single instance set to 50% applies 50% of the Snapshot value and reduces the sound by 40%.

If however there were two instances of the snapshot being controlled by two events to trigger at the same time they would both trigger and their values would combine. So if they were both set to 20% Intensity they would stack to apply 40% of the effect, which in this case would reduce the volume by 40%.

The Snapshot Priority List

The Snapshots Tab in the Mixer is more than a simple browser window. It provides functionality for how Snapshots are processed. When multiple Snapshots are triggered simultaneously Studio responds differently depending on the nature of the Snapshots being triggered.



Snapshot Priority List

Multiple Instances of the same Snapshot

Triggering multiple versions of the same Snapshot causes the values of the Intensity Property to be stacked and provide the final intensity value to be applied.

Multiple Snapshots with no shared Properties

If there are multiple Snapshots triggered and none of them share any Properties then they can all freely be applied at the same time and Studio adjusts the relevant Properties to influence the output signal.

Multiple Snapshots with Shared Properties

If multiple Snapshots with shared Properties are triggered then FMOD Studio must apply them in order of the priority they have been assigned. The order in which Snapshots appear in the Snapshot Priority List defines this behavior; See Priorities.

Priorities

The Priority list works from the bottom up. This means that the further up the list the higher a priority a Snapshot has. From a processing point of view when Studio detects Snapshot conflicts, it works its way up from the bottom of the list, each level higher is a higher priority and takes precedence over lower priority entries.

Note

FMOD Studio only overrides Snapshots values when absolutely necessary and does so only to the Properties it needs to and only by as much as it needs to. This allows the Snapshot system to work effectively while allowing the creative requirements of a game to be met.

Snapshot Automation

FMOD Studio includes an Interface display that is designed to support Snapshot Automation from within The Mixer. The Mixer Display buttons toggle between the regular Mixer view and a horizontal Track view of Mixer elements. This provides an interface to add automation to Properties within a Snapshot.

The Snapshot Track interface looks very like the Event Editor. It includes a Timeline, can also have Game Parameters added and supports the creation of Automation Tracks and their relevant functionality.

Snapshot Timeline and Parameters

Snapshots have the option to include a Timeline and Parameters. Neither is required but they can both extend the functionality of a Snapshot significantly.



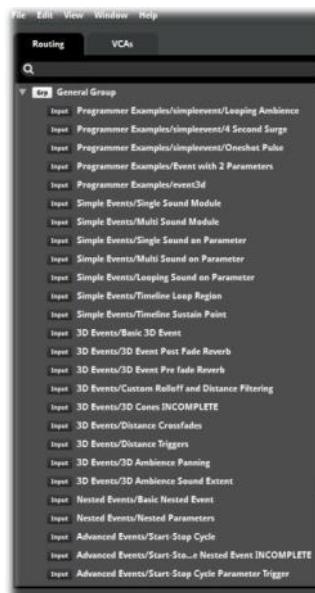
Snapshot Timeline and Parameters

Each time a Snapshot is triggered, the Timeline position starts again from zero on the Timeline. A Parameter starts from whatever value it is set to in the snapshot module's properties.

Automation for Mixer Properties can be added within the Timeline or Parameter views. Only Properties that have been Scoped In for the Snapshot in question can have Automation applied. Properties can of course be Scoped In at any time, but this needs to be done before Automation can be created for a specific Property.

Mixer Routing Window

The Mixer Routing Window can be accessed via the Menu Bar Window/Mixer Routing. FMOD Studio displays a pop-up Window similar in appearance to the Event Browser. The Mixer Router defines Signal Paths within the Mixer; it is not just an administrative display. This replicates the Routing functions of the Mixer Routing area within the general Mixer view.



Mixer Routing Window

The Mixer Routing Window also includes the Voltage Control Amplifier (VCA) Tab and this is the only way to access the VCA area. Within the VCA Tab, VCAs can be created and organized and Mixer Buses can be dragged from the Mixer Router onto VCAs within the area to assign them to a VCA.

If you drag an item in either browser over the other tab, the browser associated with that tab displays.

Assigning Mixer Buses to VCAs

1. Open the Mixer Router Window and select the VCA Tab
1. Drag the desired Bus onto the target VCA from the main Mixer Router area or from a second pop-up Mixer Router. The object is assigned to that VCA.

Any number of Mixer Router Windows can be open at a given time, which can make it easier to assign objects to VCAs as well as to set up your projects Routing system by dragging objects from one window to another.

Note

Remember that objects dragged into the VCA area are simply assigned to a VCA whereas objects dragged within the Mixer Router are positioned within the Signal Chain. Consider these actions carefully before progressing.

Add Sends: To Mixing Desk

This is a sub menu of the Add Sends context sensitive menu within the Deck area. This command provides routing access to Return Buses within the Mixer to allow for the Signal from an event to be routed directly to a Mixer Return Bus rather than just to a Return Track within the event. It also allows direct routing access to the Reverb Return Bus.

Channels and Panning

FMOD Studio provides a range of different Panning controls for both event objects and Mixer objects. The Deck region displays the input levels, output levels and Panner control appropriate for the channel setup selected.

There are three channel setups available for objects within FMOD Studio

- Mono
- Stereo
- Surround

The input and output levels as well as the Panner Controls automatically update to reflect which channel setup has been selected. By default Studio assigns Surround channel setup to all objects except Return Buses which default to mono.



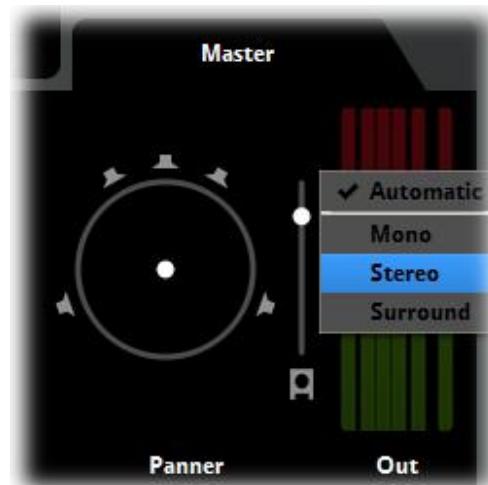
Pan and Channel Controls

FMOD Studio also has an automatic system of channel setup by default. If an event has its channel setup switched to Mono, Studio automatically switches all channel inputs and outputs along the signal path to match this switch. If the automatic setting is overridden somewhere along the signal path however the channel is set at whatever setup is defined by you at each stage of the signal path and stays in that mode.

Setting Channel modes

All Channel and Pan Controls Properties are consistent across both the Event View and the Mixer View in FMOD Studio. Channel controls default to automatic but can be altered at any time. This tutorial covers the steps to switch between channel modes.

Step 1 Switching channel output modes



Deck region with channel setup

1. Navigate to any object that contains a Deck Region
2. Within the Deck region right click on the output peak meters
3. From the context-sensitive menu select the desired Channel output mode

The display changes the Channel output mode to the appropriate number of meters to correspond with the selection; 1 for mono, 2 for stereo and 6 for surround. The Panner control also updates to provide the appropriate controller for the channel setup. There is no Panner control for channels defined as Mono as by its very definition Mono sends a single signal of equal strength to all outputs.

Note

It is only possible to define the output channel setup for an object in FMOD Studio. The input is by necessity defined by the previous stage in the signal path. An Event that has a its channel output set to stereo always generates a corresponding Mixer Input with its input set to stereo.

Using the Panner Controls

The Panner controls are used to define the strength of a signal through each of the available output channels. A Panner works by dividing the signal among the available channels.

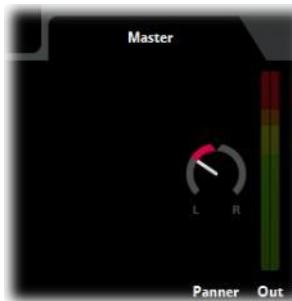
Concept

A Panner is not the same as applying volume to channels as the overall 100% signal is divided among the available channels. A Stereo Channel setup when evenly balanced sends 50% of the signal to each channel evenly. To adjust the Panning weight to increase the signal to one channel must result in a corresponding and relative reduction in signal to the other channel. So setting the right channel to 65% must result in the left channel being set to 35%. The two values always combine to make up the 100% signal output.

Surround Panning has one exception to this rule. The sub frequency channel is isolated from the surround channels and can be defined without influencing the surround channels. The values of the surround channels all combine to create the 100% signal output.

Stereo Panner

The stereo Panner has a control display that lets you adjust values from 100% left to 100% right and all values in between. By default the Stereo Panner is set to the center position of 50% left/50% right. Clicking on the Stereo Panner and dragging the mouse up and down changes the balance between left and right. The control visually displays the weighting between the two channels



Stereo Panner Controls

Surround Panner

The Surround Panner incorporates two controllers: the sub channel level slider and surround Panner speaker controller.

The sub channel level is an independent controller that can be used to define the signal strength sent to the sub frequency channel. Dragging the mouse up and down moves the controller through its range of values from 0-100%

The surround Panner speaker controller lets you move the signal strength around within the surround output region. By default the Panner is set to the center of the display region, meaning that the track outputs a signal equally to all channels. Dragging the mouse around the surround region alters the weighting of the output signal between the available channels. The display updates with a green highlight to indicate the weighting strength of the output signal.



Surround Panner Controls

Regardless of the method of channel output setting selected or the type of Panner that is being controlled, the output of the sound signal should reflect the changes. If you are unable to hear changes to your audio output that correspond to the changes you are defining then there may be an issue with your computer sound setup or your hardware setup, or an error somewhere further down the signal path. Consult your system setup manual as well as any hardware setup manuals to ensure your system is functioning as expected.

Pan Automation

The Pan Property can have Automation added. The available options depend on the currently selected Pan mode.



Pan Context Menu

Mono

No Panning options available.

Stereo

Automation

Surround

Surround Direction

Automation

Surround Extent

Automation

Stereo Axis

Automation

Stereo Width

Automation

Automation Properties function in the same manner as for all other FMOD Studio Properties.

Surround Direction

The Sound Direction Property defines where within the Surround Pan area the sound is located. Automation of modulation of this Property allows for the sound source to be repositioned within the Panning plane to achieve pan ‘movement’

Surround Extent

Surround Extent defines the degree to which a sound appears to be ‘around’ the listener. At its smallest value a sound appears to occupy a single point in space. As this Property is increased in value it expands across the available speakers and wraps around the listener position.

Note

Surround Extent and Surround Direction are linked in their functionality. A sound set to fully envelop the listener does not have a discreet directional location, so the lower the value of Surround Extent the more important the values defined by Surround Direction become.

Stereo Axis

Stereo Axis defines the central line of the pan space and how it is oriented.

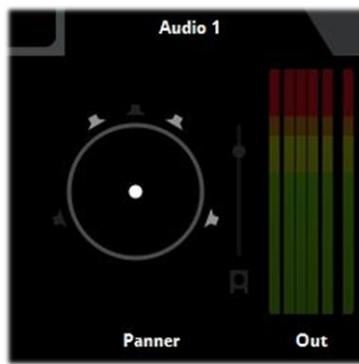
Stereo Width

Stereo Width defines the ‘broadness’ of a sound. Increasing this Property makes the sound appear wider and more open, conversely reducing the Property makes the sound seem closer to the listener.

Switching Speakers On and Off

When set to Surround Panning it is possible to switch speaker channels on and off as required. Clicking on a channel’s Speaker icon grays it out to indicate that speaker output is now switched off. No signal is routed to a speaker channel while it is switched off.

When a speaker channel is switched off it is not just muted, its signal is routed to the remaining speaker channels appropriately. For instance if an Audio Track has been panned to the center speaker in the Surround Panner and then the center speaker is switched off the signal is rerouted to the front left and right speakers with a 3dB attenuation applied.



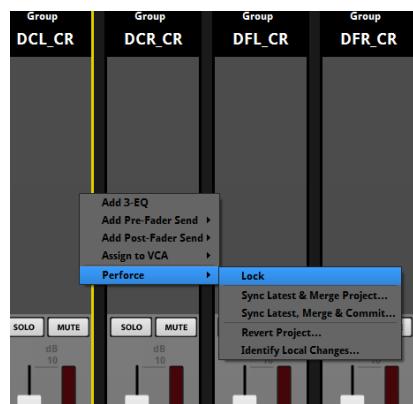
Surround Panner with Speaker Channels switched off

Mixer Strip Source Control Lock

FMOD Studio is designed to directly integrate with the Perforce and Team Foundation Server source control systems. These systems are ideal for audio teams to use to ensure the security of project data and allow multiple users to work on a single project while mitigating many of the risks of altering other team members' work.

One of the features of the source control integration is to allow individual Mixer Channel Strips to be locked in the source control setup to prevent unwanted alterations. To lock a Mixer Channel Strip, right click on the Channel Strip to access the context menu and from the revision control submenu and select Lock.

Once a Mixer Strip has been locked it is clearly indicated as locked to all other users of the project to ensure all team members are aware of its status. If the user who locked the Channel Strip manually unlocks the Strip or checks in any changes to that Channel Strip the Channel Strip is then unlocked.



The Perforce Lock command

The Perforce Lock function only exists for Mixer Channel Strips. A similar function can be applied to events to lock an entire event. A revision control sub menu also exists in the event context menu accessed by right clicking on an event in the Event Browser.

All functionality related to source control and revision control specific actions are only accessible if a revision control integration has been specified in the project's Preferences window (Menu Bar/Edit/Preferences...).

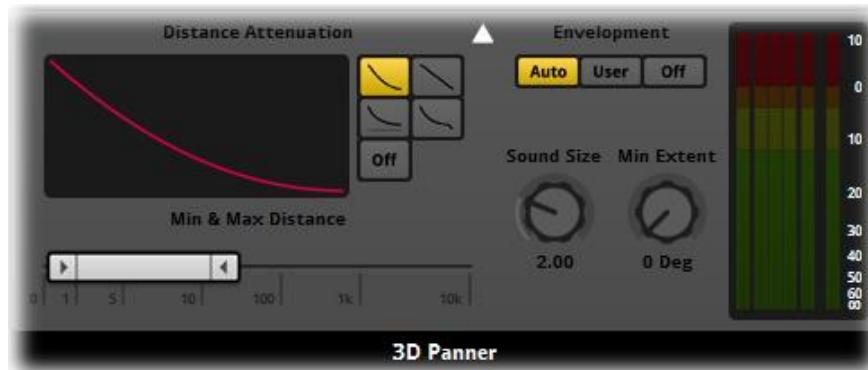
Note

The use of source control locks should only be considered during the final stages of a project and when a team of multiple audio members is working on the same project. In general this system is designed to provide stability of project data, but it can also limit workflow as once an element is locked by an individual it cannot be accessed by other team members until it is unlocked. If a team member is away sick or unavailable this can cause workflow issues. Consider carefully how you establish your workflow practices before utilizing this feature.

The 3D Panner

The 3D Panner defines the behavior of a sound event within a 3D world space. By default all sound events in FMOD Studio are created as 3D sounds. The 3D Panner directly influences how sounds relate to the Listener in 3D audio.

Many games are created in a 3D world space and utilize 3D audio. 3D audio has far more complex playback behavior than traditional 2D audio and as such there are a range of settings to define the exact behavior required of a sound played in a 3D space. The Master Track of every Sound Event created in FMOD Studio contains a 3D Panner in the Deck region. The default setting values of the 3D Panner allows a general standard form of playback behavior from all sounds.



3D Panner Controls

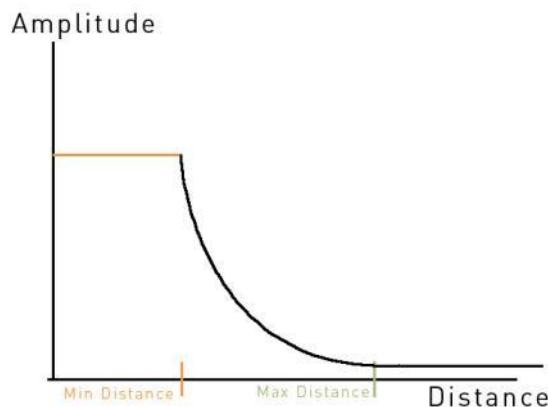
Min and Max Distance

There are two primary settings that define how a sound is heard in a 3D space. Min and Max Distances refer to the Minimum Distance a sound travels within the 3D space before attenuation begins. The Maximum Distance is the point at which attenuation stops. Understanding these two functions is an essential part of working with 3D audio.

The Min and Max distance values can be used to influence a sound within a 3D space by defining how far that sound travels before drop-off occurs (Min Distance) and then defining a point where drop off ceases (Max Distance). The Distance Attenuation settings display the type of curve applied to the drop-off and provides four different curve types and the ability to switch off drop-off completely.

3D Sound Properties

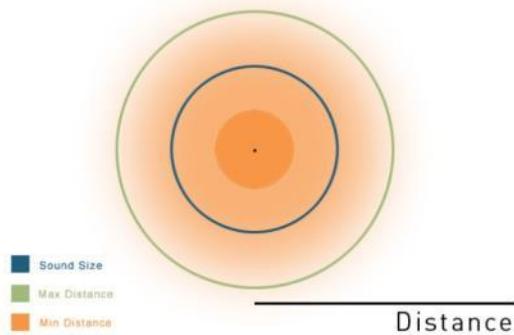
Understanding the relationship between the 3D Min and 3D Max distances is essential for creating 3D audio environments. These two properties define a major aspect of how audio functions in 3D spaces. There are other properties that also influence sound behavior but they are usually closely related to the 3D Min / Max properties.



3D Min and Max Distance Behavior

When presented on a graph the relationship between 3D Min and Max properties can be examined. The vertical axis indicates the amplitude of the example sound. The exact amplitude is not important for this example, what is important is that the amplitude remains constant at its defined value from its point of origin out to the Min distance amount. This means that wherever the listener is in the area between the point of origin and the Min distance the sound is heard at its defined volume.

Once the Min distance is reached the amplitude drops off and the sound starts to fade. The exact way in which the sound fades is determined by the attenuation curve. The amplitude of the sound continues to reduce (following the curve characteristics) until the Max distance is reached. At this point the amplitude no longer reduces and the sound level is constant from the Max distance point to infinity. Note that if the Max distance is set sufficiently far away from the Min distance the attenuation curve most likely results in the sound signal dropping to zero.



3D Min and Max distance behavior (top-down view)

If we examine this same information from above we can extend our understanding of the process. The sound emanates outwards from the point of origin in a spherical manner just as sound does in the real world. The 3D Min Distance value defines how far it travels before it starts to attenuate. In this example this is represented by the solid orange circle. This Min Distance region essentially has a volumetric quality in that even though the sound is generated from a single point (which all sounds are) the sound is maintained at its defined volume within the entire region. Once outside of the min distance the sound starts to attenuate towards the 3D Max Distance point.

Sound Size

This diagram also includes an indication of the Sound Size property. This is separate and independently defined to the 3D Min and Max distances but its behavior is related. The Sound Size defines how big an area the sound occupies and while you are within that area you can perceive that

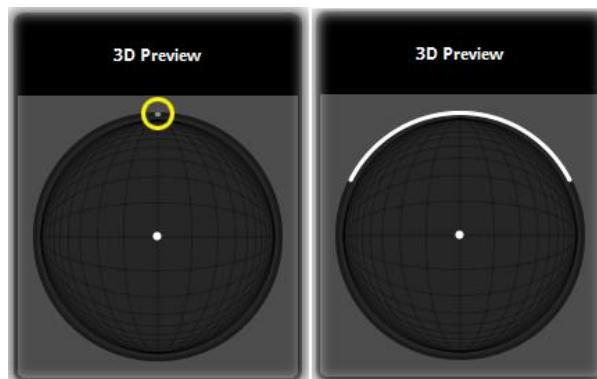
the sound is all around you, encompassing you. Once you move outside of the sound area it occupy a specific location in 3D space and you can perceive its direction and distance from the listener's point of view.

To explain this example in real-world terms, think about how a car engine sounds if you have the bonnet open and your ear close to the engine. The engine is producing lots of different sounds that combine to create the overall sound of the car running. Listening so close, you are surrounded by the sound of the engine as if you are in among the many components. As you step back and walk away from the car the sound is no longer all around you and it now occupies a specific point in space, as you walk further away you perceive that the sound of the car becomes smaller in space and all the sound is focused into that small area.

By defining the Sound Size as larger than the 3D Min distance you extend the volumetric area of the sound beyond where it starts to attenuate, so while you are still "inside" the sound, you start to hear it reduce in volume as you move away from the Min Distance point. Once you move outside the Sound Size point you notice the sound has a discernable location in the 3D world space, and once you move past the 3D Max distance the sound no longer reduces in volume and if it is still audible it remains so indefinitely.

Min Extent

The Min Extent property allows you to expand the speaker spread of a 3D sound. The default value of zero results in a 3D sound occupying a single point in the 3D world. This is represented by speaker playback of a discreet signal panned to a single location that corresponds to the location of the sound relatively in the 3D world. By increasing the value of this property the sound expands across a greater area of the speaker pan range. This can make a sound seem bigger as it encompasses a larger "area" of the speaker output. Large dynamic events such as explosions may benefit from considered use of this property.



Min Extent values displayed in the 3D preview

Note

All Events are created by default with a 3D Panner module in the Master Track. If your sound is designed to function in a 2D manner it is best to remove the 3D Panner from the Event. This optimizes performance of your project.

Pan Override

The Pan Override property allows you to define specific speaker output for a Sound Event. This override output can be mixed with the 3D Panner's output to produce different effects, from all 3D Panner signal and no override signal, to all override signal and no 3D Panner signal.

3D Preview

The 3D Preview is a graphical display that provides information for various properties controlled from the 3D Panner. The white arrow icon represents the point of origin of the emitter, and the center of the circle is the location of the listener. The circle itself represents a top-down view of a spherical 3D space, where the upper half of the diagram represents locations in front of the listener, and the lower half of the diagram represents locations behind it. The emitter icon can be dragged to different positions in order to preview the panning behavior of the active event. The Sound Size is represented by a grey zone it expands outward from the emitter point and moves in direct relation to it.

The circumference of the 3D Preview is where the envelopment value is displayed. The grey line that appears there indicates how sound is being panned to the speakers as the event is moved around 3D space. By altering the Min Extent value you can override this value by defining its minimum effect.



Various 3D Preview views

Holding down the Ctrl key while dragging the emitter in the 3D Preview allows you to alter the vertical height of the Emitter, such that dragging away from the Listener increases the Emitter's altitude and dragging closer to the Listener decreases it. This allows further auditioning of 3D cone volume as well as auditioning of the 'Elevation' built-in property.

The maximum distance the Emitter can be from the Listener is still fixed by the maximum value of the 'Distance' built-in property, so the movement of the Emitter is constrained to within a sphere rather than to within a cylinder. This may occasionally lead to the Emitter changing distance or height when the other property is being changed.

The facing of the emitter is indicated by the direction the emitter icon's arrow faces. This facing can be changed by spinning the mouse wheel while the cursor is over the 3D Preview. The facing of the listener is always straight 'up' (forward).

Distance Attenuation

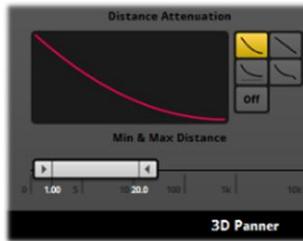
The Distance Attenuation display and selectors allow you to define the most suitable drop off curve for your sound in a 3D space. The display updates to indicate the selected drop-off curve. The Distance Attenuation can also be switched to Off which results in the sound having no drop-off curve. The various attenuation curves change the behavior of how the sound diminishes over distance.

Note

The 3D Panner automatically upgrades the Panning of the Signal Path it is included in to surround output. This can be manually overridden but doing so may affect the accurate representation of the sound event in a 3D space.

There are four attenuation curve options in the 3D Panner.

Linear Squared

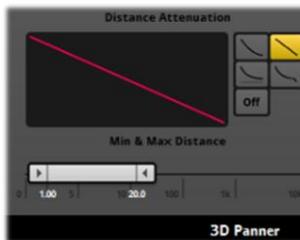


This is the default curve used by 3D Panners in FMOD Studio.

In the Linear Squared attenuation curve, the signal's amplitude is reduced according to a linear squared curve, such that it is at its normal amplitude at Min Distance and at -oo at its Max Distance. While not totally realistic, this still bears a close resemblance to how real-world attenuation over distance works and is representative of it under most circumstances.

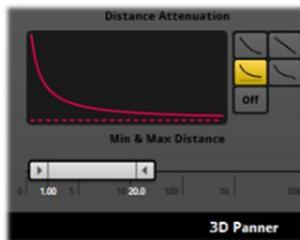
Because this attenuation method results in absolute silence at the Max distance and beyond, the maximum distance at which a sound can be heard can be known in-engine, which may be an important to the gameplay or mixing of some projects.

Linear



With this curve, the signal's amplitude is reduced at a fixed rate, starting at its normal level at the Min Distance, to -oo at the Max Distance. While not particularly realistic, this is very easy to predict and understand, and so may be useful for some projects - particularly as a supplement to other sources of attenuation and distance-based processing.

Inverse



In the Inverse curve, the signal's amplitude is attenuated by -6dB every time the distance from the source doubles, starting at the Min Distance. Under this model the sound never, in theory, reaches $-\infty$, although it can come infinitely close. The Max Distance here is the distance at which the sound stops attenuating, not the distance at which it falls to silence. This is the most realistic model of the four, but probably the least used, as its not falling to absolute silence at a predictable distance can make it tricky to work with.

Inverse Tapered



In the Inverse tapered curve, the signal's amplitude is attenuated by -6dB every time the distance from the source doubles, starting at the Min Distance. At the Max Distance point the signal tapers off.

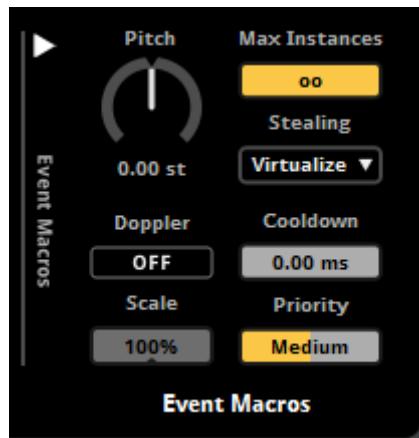
Turn Off Distance-based Attenuation



Turning off distance-based attenuation causes the signal to maintain the same amplitude no matter how far it is from the listener.

Event Macros

Each event includes a set of properties collectively known as 'event macro controls.' These properties can be found at the right end the Deck whenever the event's master track is displayed there.



Event Macros

Pitch

The event's Pitch property determines both pitch of the event and the rate at which the timeline cursor advances along the timeline. Increasing the pitch increases the speed of the timeline cursor, and conversely, decreasing the pitch reduces the rate at which the cursor moves.

Doppler

The Doppler property determines whether or not the event is subject to the Doppler Effect. By default, it is set to 'off.'

Scale

The Scale property multiplies the speed of the event for the purposes of calculating how much Doppler Effect it should be subject to. Setting this property to 0 prevents any Doppler Effect from being applied to the event.

For the purposes of calculating Doppler when auditioning, FMOD Studio assumes that your project's distances are measured in meters. However, the unit of measurement used in your game depends on the dopplerscale and distancefactor specified by `System::set3DSettings()` in your game's code. See http://www.fmod.org/documentation/#content/generated/FMOD_System_Set3DSettings.html for more details.

Max Instances

The Max Instances property sets an upper limit on how many instances of this event can play simultaneously. By default, it is set to allow infinite simultaneous instances.

If that your game's code attempts to create a new instance of the event, and creating a new event instance would result in the number of simultaneous instances of the event exceeding the event's Max Instances property, FMOD Studio culls or virtualizes event instances according to the event's 'stealing' macro control.

Stealing

The Stealing property determines how FMOD Studio behaves when your game's code attempts to create a new instance of the event, and the number of instances of that event is already equal to the event's Max Instances property. The various stealing behaviors are as follows:

Oldest

The oldest instance of the event is immediately stopped and replaced by the new instance.

Quietest

The quietest existing instance of the event is immediately stopped and replaced by the new instance.

An event instance's 'Quietness' is derived from the volume of the mixer hierarchy, the event's master track volume, and the gain applied to the event by its 3D panner effect module. The volume settings of the event's audio tracks, nested events, and playing sound modules are not taken into account.

Virtualize

All new event instances are created and behave normally, but instances in excess of the event's Max Instances property are not routed to the project mixer and so produce no audible output.

Only the quietest instances of an event are made virtual. If a virtualized event instance ever becomes louder than a non-virtual event instance, FMOD Studio automatically makes the quieter instance virtual and begins routing the louder instance into the mixer. Virtualized event instances are also made audible if the number of event instances drops below the event's Max Instances property.

An event instance's 'quietness' is derived from the volume of the mixer hierarchy, the event's master track volume, and the gain applied to the event by its 3D panner effect module. The volume settings of the event's audio tracks, nested events, and playing sound modules are not taken into account.

Off

New event instances are not created.

Cooldown

The Cooldown property specifies the minimum period of time after starting an event instance that a new instance of the event can be started.

When your game's code attempts to play a new instance of an event, the time elapsed since it last started an instance of that event is compared to the event's cooldown. If the time elapsed is less than the cooldown, the new instance does not play.

This limits how frequently new instances of an event can be started.

Priority

The Priority property is passed to the low level virtual voice system. For more information about this, see <http://www.fmod.org/docs/content/generated/overview/virtualvoices.html> in the FMOD Studio Programmer's API documentation.

Effect Modules

FMOD Studio supports a variety of effects. Effects can alter a signal in a variety of ways.

Effect modules can be added in the Decks of Audio Tracks, Return Tracks and event Master Tracks in the event editor, and Mixer Group Buses, Return Buses and the Master Bus in the Mixer.



Effects modules

Effect modules can be added to the Deck region of Audio Tracks, Master Tracks, Group Buses and The Master Bus. To add an effect module to the deck, right-click on an empty area of the deck to open the context-sensitive menu, select the 'Add Effect' menu item, then select the specific effect you want to add.



Adding a new Effect

Each effect module contains controls that represent its properties.

To delete an effect module, right-click on it to open the context-sensitive menu, then select the 'Delete' menu item.

You can bypass an effect for the purpose of auditioning. To do this, right-click on the effect module you want to bypass to open the context-sensitive menu, then select the 'Bypass' menu item.



Bypassing an Effect

Third-Party and User Created Plugins

FMOD Studio supports third-party and user-created Plugins. Because of this the various context menus that apply to adding Effects Modules to the Deck region may vary in the content they display, so the menus described or pictured in this manual may differ slightly from those that appear in the version you are using.

Effect Automation

Effect modules added to event Audio tracks and event Master Tracks can have their properties automated.

Automation allows you to define the value of a property for any given value of a parameter.

To automate a property of an effect, right-click on the property to be automated to open the context-sensitive menu, then select the 'Add Automation' menu item. This causes an automation track to be added to the Audio or Master Track that holds the effect.

You can add automation points to an existing automation track by double-clicking on the automation track, and can click and drag existing points to move them.

If a property has a red automation curve on multiple parameters, the property's actual value is the average of the values assigned to it by those parameters.

3-EQ

The 3EQ Effect provides a simple method of balancing the major frequency groups of a signal path. Low, mid and high frequency regions can be easily attenuated to apply a broad range of audio effects to a signal's output sound.



3-EQ Effect

3 EQ Properties and Default Values

- Low 0.00 dB
- Mid 0.00 dB
- High 0.00 dB
- X-Low 400 Hz
- X-High 4.00 kHz
- X-Slope 24

The 3 EQ can drastically alter the output of any type of sound. Short pulses as well as sustained signals can be significantly altered in character. The 3EQ is a quick method for simulating "radio talk" type effects as well as simulating low fidelity playback. The X-Low and X-High functions allow for the cross over points between Low, Mid and High to be defined, so "Low" can be whatever frequencies you define as low.

Properties

Low, Mid and High

Each of the Low, Mid and High Properties controls the attenuation of the frequencies assigned to that particular group. Because the X-Low and X-High setting define the exact range all of these setting essentially have the same functionality but is applied to the defined frequency range. The Low, Mid and High Properties have values from -80 dB(-∞ dB) – +10 dB.

X-Low

The X-Low setting defines the crossover point between the Low and Mid setting values. This function defines the point where the Low frequencies finish and the Mid frequencies begin. The terms Low, Mid and High are relative as this crossover point can be defined anywhere within the available frequency spectrum. The X-Low Setting has a value from 10Hz-22 kHz

X-High

The X-High setting defines the crossover point between the Mid and High setting values. This function defines the point where the Mid frequencies finish and the High frequencies begin. The terms Low,

Mid and High are relative as this crossover point can be defined anywhere within the available frequency spectrum. The X-High Setting has a value from 10Hz-22 kHz

X-Slope

The Slope setting allows for three values to be applied defining the gradient of the crossover curves between the Low, Mid and High ranges. The three Properties refer to the following values

- 12 dB per octave
- 24 dB per octave
- 48 dB per octave

12 dB defines the shallowest curve for the crossover while 48 dB defines the steepest curve.

The cross over curves significantly influence the playback behavior of a sound depending on its frequency makeup and the values defined for Low, Mid and High Properties.

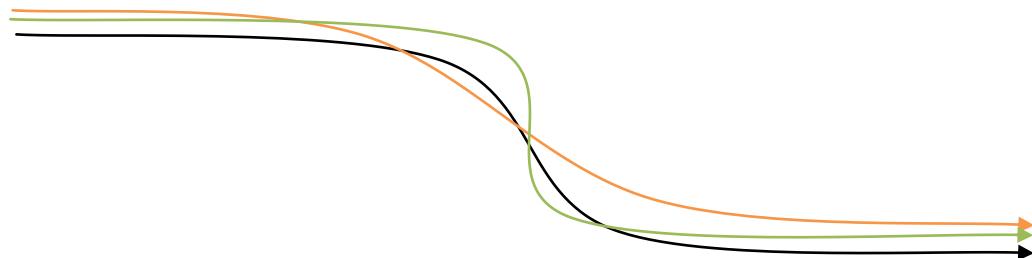
Note

Slope selection has a direct impact on CPU usage. The greater the resolution of the slope value the more CPU is required. A 24 slope uses twice the CPU of a 12 slope while a 48 slope requires 4 times the CPU power.

12 dB per Octave

24 dB per Octave

48 dB per Octave



Compressor

The Compressor Module narrows a signals dynamic range by reducing the loudest parts of the sound. This is a common process applied to media to ensure that softer passages are not lost in the mix and so you do not need to constantly alter the playback volume.



Compressor Effect

Compressor Properties and Default Values

- Threshold 0.00 dB
- Ratio 1:2.5
- Attack 20 ms
- Release 200 ms
- Gain 0.00 dB
- Sidechain Off

The Compressor default settings are a good starting point to work with, but each signal needs to be specifically tuned to get the best results out of the Compressor Effect. Dynamic Range Compression is a very complex aspect of audio production and entire books have been written on the subject. For more information look for the tutorial video on FMODTV on YouTube that covers compression for FMOD Studio.

Properties

Threshold

Threshold is the level at the compression effect is activated. The default value of 0.00dB is nominally the loudest a signal can be (above 0.00dB it risks distortion). Reducing the value of the Threshold

Property lowers the level at which the compressor activates and as a result the overall dynamic range is more compressed. Threshold has a range from -80dB(-∞ dB) – 0.00dB.

Ratio

Ratio sets how much compression is applied to the signal. Only signals that exceed the threshold are compressed. The default value of 1:2.5 means that every signal that is 2.5dB above the threshold is compressed to 1dB above the threshold. The greater the ratio, the greater the compression. The Ratio Property has a value from 1:1.0-1:50.0.

Attack

Attack defines how rapidly the compression effect is applied. This Property is an important part of working with signal compression successfully as it tweaks how your sound “sounds” when passing through the Compressor. Too rapid an Attack time makes the output sound very harsh, while too long an Attack time can be just as bad. The Attack Setting has a range from 10ms-500ms.

Release

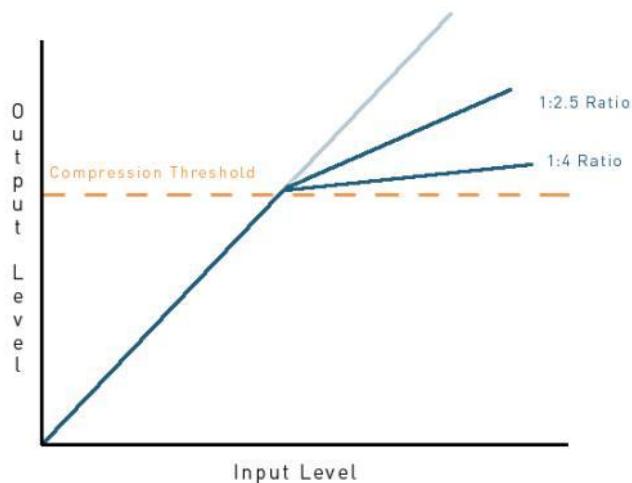
Release also effects how the eventual playback of your sound is affected and needs to be carefully considered alongside the Attack Property. The default values for both Attack and Release are good values to start with when working with Compression. The Release Property has a range from 10ms-500ms.

Gain

The process of compressing the dynamic range of a signal, by its very nature, reduces the overall volume of the sound as it compresses the loudest elements downwards. The Gain property provides the ability to increase or decrease the output volume of the newly compressed sound. The Gain Property has a range from -30dB -30dB.

Sidechain

The Sidechain button toggles between the on and off function for Sidechaining. Check out the Sidechain section of the manual for more information on Sidechains.



Dynamic Range Compression

Note

Compression is an important aspect of many audio mixes. A thorough understanding of the process improves the quality of your audio. We recommend you invest some time in learning good techniques.

Chorus

The Chorus Effect Module produces a “widening” of the sound by adding slightly altered duplicate versions of the original sound to create the illusion of multiple voices playing the same sound. The Chorus Effect works similarly to the Flange Effect except it utilizes three delay lines instead of only one.



Chorus Effect

Flanger Properties and Default Values

- Rate .80 Hz
- Depth 3%
- Mix 100%

The Chorus Effect is more noticeable with sustained or looping input sounds. Single pulse notes do not produce much of an effect when chorused. A sustained note sounds like there are slightly “out of tune” duplicates being played alongside similar to three people singing the same line of a song. The lines in a Chorus Effect are out of phase with each other by 90 degrees, so the Dry signal is a 0°, tap 1 at 90°, tap 2 at 180° and tap 3 at 270°.

Properties

Rate

Each delay line in the Chorus oscillates on a sine wave. The rate property defines the speed or the sine wave that each delay line follows. The Rate Property has a range from 0.00-20.0 Hz.

Depth

The Depth Setting defines the size of the sine wave that each delay line follows. 100% makes it oscillate from the dry position to the delay value, 0% would make it remain constant on the delay value. The Depth setting has a range from 0-100%.

Mix

The Mix Setting blends the power of the original and duplicate signals. Its default of 50% produces an even mix of the two signals. Altering the value up or down changes the mix to either emphasize the

original signal or the duplicate signal. This can produce a broad range of results. The Mix Setting has a range from 0% Dry (only the original signal) – 100% Wet (only the duplicate signal).

Delay

The Delay Effect module plays the input signal and then repeats it a specific number of times depending on the Properties provided. This module includes three controllers that allow for the signal processing to be altered in a variety of ways.



Delay Effect

Delay Properties and Default Values

- Delay 500 ms
- Wet Level 0.00 dB
- Dry Level 0.00dB
- Feedback 50%

The default Properties when applied to a single ping input sound produces a series of repeating echoes spaced half a second apart. The first echo plays back at the same output level as the initial signal and each subsequent echo diminishes in amplitude until the sound is inaudible. The default setting produces approximately 16 audible echoes.

Properties

Delay

The Delay Setting defines the length of time between the playback of each echo. This setting is measured in milliseconds and abbreviated to ms. The default value of 500 ms is equal to one half a second delay. The Delay Setting has a range from 10ms-5 sec.

Wet Level

The Wet Level Setting defines the output level of the first echo. The default value of 0.00 means the first echo is equal to the output level of the initial source signal. Each subsequent echo is reduced in amplitude from the one before it; the Wet Level setting defines the initial amplitude of the echo drop-off curve. The Level Setting has a range from -80 dB-10 dB.

Dry Level

The Dry Level Setting defines the output level of the original signal without the Delay Effect applied. The default value of 0.00 means the original signal plays at the same level as when it entered the Delay Module. Reducing the Dry Level results in the affected signal being easier to hear. A blend of the Wet Level and the Dry Level can be used to achieve the desired output balance. The Dry Level Setting has a range from -80 dB-10 dB.

Feedback

The Feedback Setting defines the number of echoes that follow the initial signal. The echoes always diminish with each subsequent repeat, the less echo steps defined the more extreme the attenuation is between each echo step. Feedback has a range from 0.00-100%.

Flanger

The Flanger Effect produces a duplicate of the original signal that is played at the same time as the original signal but with a minute delay in its starting time. The delay value of this second signal also changes over time in a sinusoidal manner. The combination of these two processes creates a sweeping effect as the two signals blend. Their phase relationship with each other ebbs and flows creating the sweeping effect.



Flanger Effect

Flanger Properties and Default Values

- Rate .10 Hz
- Depth 1.00 ms
- Mix 50%

The Flanger Effect is more noticeable with sustained or looping input sounds. Single pulse notes do not produce much of an effect when flanged. A sustained note sounds like there is a slightly “out of tune” duplicate being played alongside. The relationship between the two sounds alters over time as they shift more and less out of phase with each other.

Properties

Rate

The Rate setting creates a pattern at the designated frequency. This pattern is what determines the rate of delay of the Flange Effect. The duplicate signal is delayed initially by the amount set in the Delay Setting value, but it alters over time by the amount defined in the Rate setting. The rate setting has a range from 0.00-20.0 Hz.

Depth

The Depth Setting defines the length of the initial delay between the original signal and the duplicate signal. At its minimum value of 0.01 there is such a small delay it may not be noticeable. The delay value may not seem very large but when the value is multiplied by the Rate setting it can cause quite extreme changes to the overall signal. The Depth setting has a range from 0.01 -1.00 ms.

Mix

The Mix Setting blends the power of the original and duplicate signals. Its default of 50% produces an even mix of the two signals. Altering the value up or down changes the mix to either emphasize the original signal or the duplicate signal. This can produce a broad range of results. The Mix Setting has a range from 0% Dry (only the original signal) – 100% Wet (only the duplicate signal).

High Pass

High Pass is a type of filter that allows signal material of a defined frequency range and above to be audible while filtering out frequency material below the defined range. This can produce a variety of results very much dependent on the frequency makeup of the input signal and the Properties defined within the Effects module.



Hi Pass Effect

High Pass Properties and Default Values

- Cutoff 2.00 KHz
- Resonance 1.00

The High Pass effect is noticeable for both sustained and single pulse or repeated pulse types of sounds. Because of the nature of this Effect module if the original signal is entirely made up of frequency material above the Cutoff frequency defined then there is minimal effect to the output signal from that setting, but the value of the Resonance setting may still produce an audible effect.

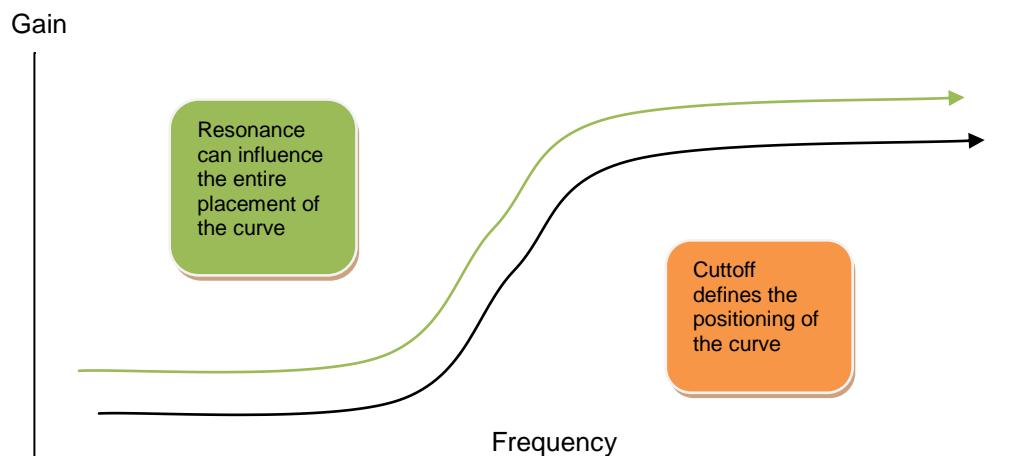
Properties

Cutoff

The Cutoff setting defines the frequency value at which the filter activates. Any material of a frequency value below the Cutoff frequency is filtered out of the signal. The extent to which this affects the input sound is entirely dependent on the frequency makeup of the original sound. If the entire sound consists of frequencies below the Cutoff value then the entire sound is filtered out, essentially leaving silence. The Cutoff Setting has a range from 10.0 Hz – 22.0 KHz

Resonance

The Resonance Setting acts as signal boost for frequency material that matches the Cutoff frequency. The nature of how frequency signal processing occurs means that this boost is not isolated to the exact frequency value defined in the Cutoff setting but that it lifts the power of the cut-off frequency and generates a curve in values that drops off to either side.



Note

Because of the nature of the resonance curve and how harmonics function, the Resonance Function can produce a wide range of effects. It is possible to set a cut-off frequency at a particular value and use the resonance function to affect the sound of frequency material far outside the range of the cut-off frequency.

Limiter

The Limiter Effect allows for control of signal peaks while also increasing the overall levels. By limiting the maximum level and boosting the lower levels the limiter can be used to produce a punchier overall output signal.



Limiter Effect

Limiter Properties and Default Values

- Input 0 dB
- Ceiling 0.00 dB
- Release 10.0 ms
- Channels Link deactivated

The Limiter Effect is particularly good in adding punch to short pulse sounds, but is also very useful for tightening up longer loop sounds.

Properties

Input

The Input defines the amount of gain prior to limiting. This setting works like an input gain control.

The input setting has a range from 0 dB-12 dB

Ceiling

Ceiling controls the maximum output signal that leaves the Limiter effect. The output amount cannot exceed this value. Louder input peaks are mostly limited, while lower signal levels are increased.

The Ceiling setting has a range from -12 dB (-∞ dB)- 0.00 dB

Release

The Release setting defines how quickly the Limiter recovers from signal peak reduction. The faster the release, the louder the limiter output level.

The Release setting has a range from 1.00 ms- 1.00 sec

Channels

Link applies absolute peak levels for all channels the same, this maintains relative values between channels while limiting. Without Link a flat volume is applied across all channels.

Note

The Limiter is a basic effects unit. It is fairly low quality but is very cheap on resource use. The Limiter Effect has no Attack time and so instantly clips signals at ceiling level and then allows a slow release.

Low Pass

Low Pass is a type of filter that allows signal material of a defined frequency range and below to be audible while filtering out frequency material above the defined range. This can produce a variety of results very much dependent on the frequency makeup of the input signal and the Properties defined within the Effects module.



Low Pass Effect

Low Pass Properties and Default Values

- Cutoff 2.00 KHz

- Resonance 1.00

The Low Pass effect is noticeable for both sustained and single pulse or repeated pulse types of sounds. Because of the nature of this Effect module if the original signal is entirely made up of frequency material below the Cutoff frequency defined then there is minimal effect to the output signal from that setting, but the value of the Resonance setting may still produce an audible effect.

Properties

Cutoff

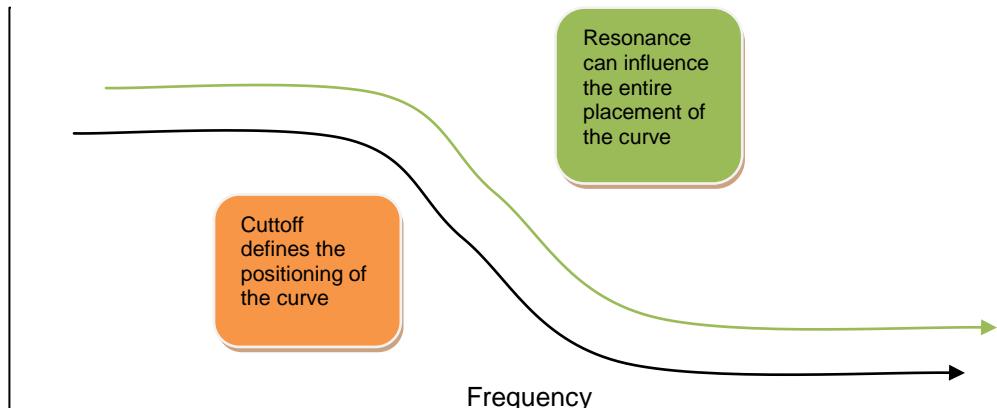
The Cutoff setting defines the frequency value at which the filter activates. Any material of a frequency value above the Cutoff frequency is filtered out of the signal. The extent to which this affects the input sound is entirely dependent on the frequency makeup of the original sound. If the entire sound consists of frequencies above the Cutoff value then the entire sound is filtered out, essentially leaving silence. The Cutoff Setting has a range from 10.0 Hz – 22.0 KHz

Resonance

The Resonance Setting acts as signal boost for frequency material that matches the Cutoff frequency. The nature of how frequency signal processing occurs means that this boost is not isolated to the exact frequency value defined in the Cutoff setting but that it lifts the power of the cut-off frequency and generates a curve in values that drops off to either side.



Gain



Note

Because of the nature of the resonance curve and how harmonics function, the Resonance Function can produce a wide range of effects. It is possible to set a cut-off frequency at a particular value and use the resonance function to affect the sound of frequency material far outside the range of the cut-off frequency.

Parametric EQ

The Parametric EQ effect allows for boosting of ranges of frequency material within a signal path. The Properties allow for specific frequency ranges to be targeted and the gain on that frequency material to be increased or decreased as required.



Parametric EQ Effect

Parametric EQ Properties and Default Values

- Frequency 2.00 KHz
- Gain 0.00 dB
- Bandwidth 1.00

The High Pass effect is noticeable for both sustained and single pulses and repeated pulse types of sounds. Because of the nature of this Effect module if the original signal is entirely made up of

frequency material above the Cutoff frequency defined then there is minimal effect to the output signal from that setting, but the value of the Resonance setting may still produce an audible effect.

Properties

Frequency

The Frequency setting defines the target frequency that is affected by the EQ Effect. Because of the nature of manipulating frequency material this setting always defines the center frequency of an overall range that is affected. The Frequency setting has a range from 20 Hz – 22 KHz

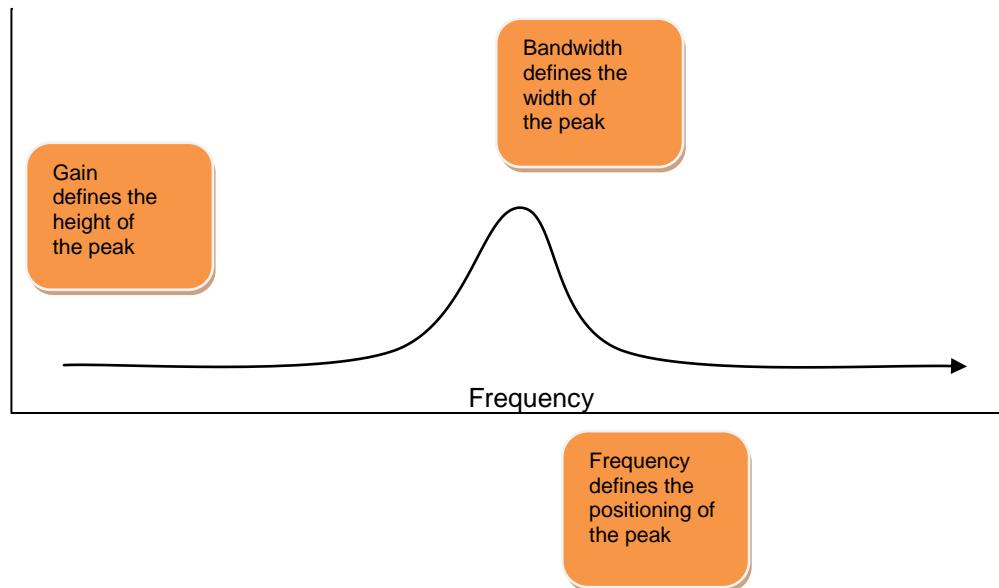
Gain

The Gain Setting allows for the attenuation of the target frequency range. The frequencies can be boosted or reduced as required over the range of this setting. The Gain Setting has a range from -30 dB (-∞) – 30dB

Bandwidth

Bandwidth defines the width of the range of frequencies around the target frequency defined in the Frequency Setting. The smaller the value of the Bandwidth setting, the fewer frequencies on either side of the target frequency are affected. The Bandwidth Setting has a range from 0.20-5.00

Gain



Reverb

Reverb is the effect of a sound being sustained beyond its initial source as a result of the environmental effects in the location where the sound is produced. This is most commonly caused by sound waves reflecting off surface material and generating subtle echoes. Reverb is a common effect used in many forms of media to reproduce the environmental behavior of different environments such as caves, large rooms, sport arenas or room with particular surface material.



Reverb Properties and Default Values

- Delay Time 1.50 sec
- Early 20.0 ms
- Late 40 ms
- HF Decay 50%
- HF 5.00 kHz
- Diffusion 100%
- Density 100%
- Low Gain 0.00 dB
- High Cut 5.00 kHz
- Low Freq 250 Hz
- Wet Level -6.00 dB
- Dry Level 0.00 dB
- Early/Late 50%

The Reverb Effect module has a large number of Properties that allow for a huge variety of reverb environmental effects to be produced. Reverberation is most noticeable on pulse type sounds where the reverb echoes continue after the primary sound has finished playing. With continual or looping sounds Reverb can produce muddy or feedback type results.

Properties

Reverb Time

The Reverb Time setting defines the length of the Reverb tail. This essentially controls how long a sound is once processed by the Reverb effect. Reverb Time has a range from 100 ms – 20.0 sec

Early Delay

Early Delay defines the amount of time before the early reflections. This is essentially defining the size of the space. A longer Early Delay time indicates the initial sound source must travel further before it is reflected. This is indicative of a larger space in which the sound is being heard. The Early Delay setting has a value from 0.00 ms – 300 ms

Late Delay

Late delay is the period of time between the early reflection and the reverb tail. This is also a function related to the size of the room space. The Late Delay setting has a value from 0.00 ms – 100 ms

HF Decay

HF Decay defines the amount of time that High Frequency material takes to decay. The behavior of high frequency sounds indicates the material makeup of a room. Low levels of HF decay dampens the room, making it sound like it has absorbent materials. High levels of HF Decay indicates lots of hard



surfaces from which high frequency sounds reflect. The HF Decay values are a percentage of overall reverb time. HF Decay has a range from 10%-100%

HF Reference

HF Reference defines the exact frequency that is used to measure the HF Decay setting. This setting can considerably alter the color of the sound that outputs from the Reverb effect. HF Reference has a range from 20 Hz – 22.0 kHz

Diffusion

Diffusion defines irregularities in a room space that break up standing waves. Low values result in a resonant space with lots of parallel surfaces; a high Diffusion value stop resonance so there are less ringing tones by scattering the reflections. Diffusion has a range from 0.00- 100%

Density

Density controls the number of extra reflections in a room space representing extra surfaces that bounce the sound wave. The higher the Density value the more complex a room space with more reflective surfaces. Density has a range from 0.00 - 100%

Low Gain

This setting controls the signal boost for the EQ controls of the Reverb Effect. The EQ is essentially a shelf filter so the Low Gain setting boosts all frequency material below the defined value of the Low Freq setting. The Low Gain setting has a range from -36 dB (-∞) -12 dB

Hi Cut

Hi Cut is a low pass filter that cuts all frequencies material above its defined value. This combines with the Low Gain and Low Freq to generate the overall EQ of the Reverb Effect. The High Cut setting has a range from 20 Hz – 22.0 kHz

Low Freq

The Low Freq setting defines the upper frequency of the Low Frequency range. All frequency material below this value is affected by the Low Gain setting value. The Low Freq setting has a range from 20 Hz – 22.0 kHz

Wet Level

Wet Level sets the overall output level of the Reverb affected signal. Wet Level contains only signal material that has been passed through the Reverb module. A Combination of Wet Level and Dry Level determines the overall output signal from the Reverb effect. The Wet Level setting has a range from -80 dB (-∞ dB) -12.00 dB.

Dry Level

Dry Level sets the overall output level of the original clean signal, unaffected by the Reverb effect. A Combination of Wet Level and Dry Level determines the overall output signal from the Reverb effect. The Dry Level setting has a range from -80 dB (-∞ dB) -12.00 dB.

Early/Late

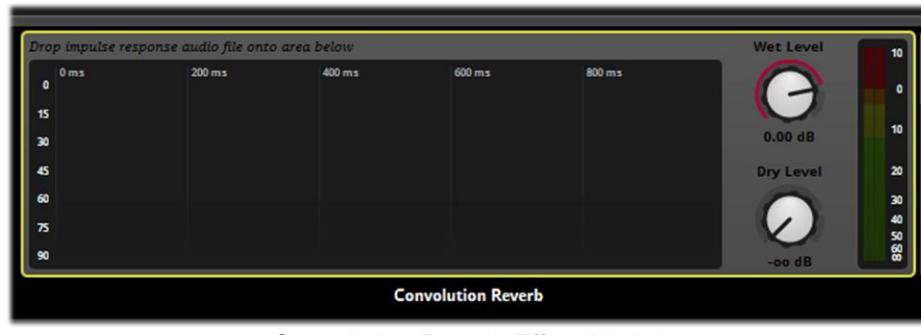
The Early/Late setting defines the balance between the Early Delay and Late Delay Properties. Its default 50% value defines an equal balance between the two delay Properties and it can be altered to weight the value towards either setting. The Early/Late setting has a range from 100% Early – 100% Late.

Convolution Reverb

The Convolution Reverb effect module can only be added to Buses in the Mixer Window. This effect applies convolution reverb to the signal based on the impulse response file dragged onto it. For best

results, the impulse response file should have a same number of channels as the target output format.

To avoid noise, the louder the impulse response, the lower either the wet level of the effect or the level of the signal fed into the effect must be.



Convolution Reverb Effect Module

Distortion



Distortion adds harmonics and texture to an existing signal by boosting the signal's Sound File until clipping occurs. Whilst distortion reduces the fidelity of a sound, at times the increased harmonics and reduced dynamic range can add an appealing effect.

Highpass Simple



This Effect is a second-order filter that leaves high frequencies unaffected but attenuates (or reduces) frequencies below a cutoff frequency point. This Effect is used to reduce the amount of bass

frequency in a signal. This property represents the frequency at which the filter stops attenuating. The cutoff frequency range is 10 to 22000Hz.

Lowpass Simple



This Effect is a first-order filter that leaves low frequencies unaffected but attenuates (or reduces) frequencies above a cutoff frequency point. This Effect is used to reduce the amount of treble in a signal.

FMOD's Lowpass Simple Effect was specifically designed to provide full bandwidth response (up to 22 kHz) at the cheapest CPU costs. It does not provide any resonance control.

This property represents the frequency at which the filter starts attenuating. The cutoff frequency range is 10 to 22000 Hz.

Pitch Shifter



This Effect allows a sound to be raised or lowered in pitch without changing its time duration.

This plugin is very CPU intensive. However, using mono files, selecting a smaller FFT size and reducing the sampling rate of the source file can reduce the CPU load, albeit at the expense of sound quality.

Pitch: This property represents the change to the original pitch ranging from 0.5 (one octave down) to 2.0 (one octave up). A value of 1.0 represents the original pitch.

Overlap: This property has been deprecated and is no longer used.

FFT Size: The property specifies the spectral resolution used in the FFT process. A higher number reduces the number of unwanted artefacts in the sound (but requires more CPU time). The FFT size ranges from 256 to 4096.

Max Channels: This property makes it is possible to set the number of channels (and memory allocation) to be used by the Pitch Shifter plugin. This property can be set from 0 (default) to 16. The value 0 represents FMOD's default polyphony, 1 = mono, 2 = stereo, etc.

Tremolo



This Effect produces a periodic variation in the amplitude of the signal.

Frequency: Specifies the speed at which the variation in amplitude occurs. Frequency ranges from 0.01 to 20Hz. The default is 5Hz.

Depth: Specifies the amount of variation in amplitude. Depth ranges from 0 (no change) to 1 (full signal). The default is 1.

Shape: Specifies the LFO shape, morphing between triangle and sine. Shape ranges from 0 (triangle) to 1 (sine). The default is 0.

Skew: Specifies time-skewing of LFO cycle. It ranges from -1 to 1. The default is 0.

Duty: Specifies the LFO 'on-time', ranging from 0 (always off) to 1 (always on). The default is 0.5.

Phase: Specifies an offset to the phase of the LFO. Ranges from 0 (LFO starts at its low point) to 1. Setting the 'Phase' to 0.5 means the LFO is at its high point, allowing the start of the signal to be heard. The default is 0.

Square: Specifies the flatness of the LFO shape, ranging from 0 to 1. The default is 0.

Spread: Creates rotation / auto-pan effects in multichannel audio files by spreading the LFO phrase evenly between channels (In 5.1 the LFE channel is synced to the center channel). Spread ranges from -1 to 1, with the default being 0.

Settings for creating basic LFO shapes:

Triangle: shape = 0, skew = 0, square = 0

Sine: shape = 1, skew = 0, square = 0

Rising saw: shape = 0, skew = 1, square = 0

Falling saw: shape = 0, skew = -1, square = 0

Square: skew = 0, square = 1

When using a pure square or saw tooth saw LFO shape, the presence of clicks should be expected as a natural by-product of instantaneous volume change. Clicks can be avoided by setting 'skew' and 'square' values within their extremes (not 1). This provides some time to ramp the volume, and avoid clicks.

Plugin Modules

FMOD Studio supports Plugin Modules and allows for both user created and third party created plugins to be used in projects. Plugins are different to built-in Effect Modules as they are not automatically supported when FMOD Studio is used in a project.

Plugins need to be set up for each platform your project runs on and be compiled into the build for that target platform. The Plugin then needs to be loaded within the FMOD Studio API. This aspect of production obviously needs to be undertaken by your programming team.

Gain Effect Module

The Gain Plugin Module provides two basic functions. The first of these is a Gain property that can be used to boost or reduce the signal that passes through it. The Gain property has a range from -80dB (negative infinity) to +10dB.

The Gain Module also has an Invert toggle button that inverts the wave form of the signal passing through it.



The Gain Plugin Module

Distance Filter Effect Module

The Distance Filter Plugin Module provides a series of Properties that can be useful in simulating the behavior of physical sound characteristics in a 3D world. The nature of this plugin is such that it needs to be used in combination with other Property settings in a specific manner to utilize the module effectively.



The Distance Filter Plugin Module

The Max Distance Property has a range of values from 0-10,000 (10K). This property defines the value at which the effects of this module are applied. Note the information below on how this value is influenced by other functions within FMOD Studio.

The Frequency Property defines the target frequency for a band-pass filter. This property has a range of values from 10Hz-22KHz. As the distance value of the event moves closer to the Max Distance Property value the filter affects all frequencies both above and below the defined Frequency value reducing all frequency content around the defined frequency. This essentially creates a high-pass and low-pass filter with converging cutoffs towards the Frequency value.

Note

The Distance Filter Plugin Module is dependent on various other FMOD Studio features. By definition it is designed to function with 3D sounds, however if a 3D Panner is included in the Event you need to switch off its Distance Attenuation or extend its Max Distance to be greater than that of the Distance Filter, as otherwise the sound fades out as a consequence of Distance Attenuation before the effect of the Distance Filter is audible.

There are a variety of different FMOD Studio Modules and Properties that can influence how a sound plays within a 3D space. Ensure all these Properties are set to complement and not conflict with each other.

Additional Modules Requiring Licenses

FMOD Studio also comes with the following two Modules:

- AudioGaming's AudioWeather sound module.
- McDSP's ML1 Limiter effect module.
- GenAudio's AstoundSound RIT effect module.

These modules require special licenses to use. Please contact FMOD at support@fmod.org for more information on licensing these Plugins.

Sidechaining

Sidechaining is a process where a channel has a compressor introduced to the Signal Chain but the control of this compressor is defined by a secondary channel. This is very useful for creating channel ducking behavior.

Channel Ducking

Channel Ducking, is when a project requires that the sound levels from one channel are reduced to allow the output from another channel to be clearly audible. The most basic example of this is many games duck non-dialogue channels when the dialogue channel is active to ensure that players can hear the dialogue clearly.

A Sidechain is a method of creating a compressor in a Signal Path that can be controlled from a different Signal Path. In the above example a Compressor would be created in the combined Signal Path of all non-dialogue channels, and the control channel would be the dialogue channel. This would allow you to tune the Compression behavior to ensure that dialogue playback was clearly audible above all other sounds.

Creating a Sidechain

Sidechains can be created in the Deck areas of both the Event Editor and the Mixer. They function slightly differently in both locations, but the creation process is the same in both areas.

Creating a new Compressor

The first step in creating a Sidechain is to add a Compressor to the channels you wish to duck.



Creating a Compressor

1. Select an event in the Event Editor or an event or Bus in the Mixer
2. Right click in the Deck area to access the context menu
3. Select FMOD Compressor through the Add Effect menu

FMOD Studio creates a new Compressor Effect Module in the Deck area of the target event or Bus. This Compressor currently functions as a regular Compressor Effect until the Sidechain is created.

Note

Sidechains can be created in either the Event Editor or the Mixer. A control channel within the Event Editor controls a Sidechained Event channel and a control channel within the Mixer controls a Sidechained Mixer channel. It is not possible to set up a Sidechain relationship from the Event Editor to the Mixer.

Creating the Sidechain

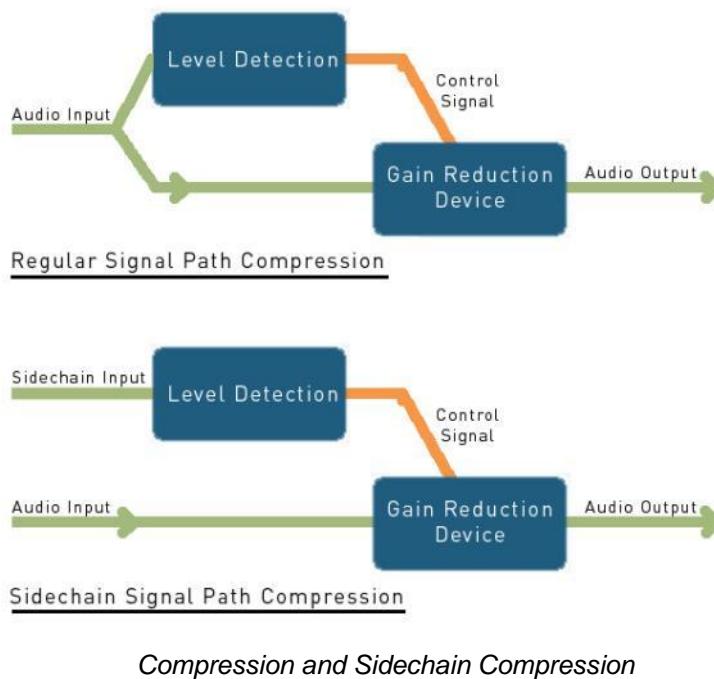


Creating a Sidechain

1. Select the event or Bus you wish to use as the control channel for the Sidechain
2. Right click in the Deck area to access the context menu
3. Within the context menu, via Add Sidechain select the Target channel to be controlled

FMOD Studio creates a Sidechain meter within the Deck area of the control Channel.

The Sidechain process is a very simple function. It works exactly as a Compressor Effect Module would work in any other circumstance with the one exception that where a Compressor uses the input signal from its own channel to define the level of compression a Sidechain compressor uses the signal from the control channel to define the level of Compression.



Note

A Sidechain compressor needs to be tweaked and tuned in exactly the same manner as a regular Signal Path compressor Effect; it simply has the function controlled from a different source. All the tip and techniques for successful compression should be equally applied to the use of Sidechain compression.

Localization

Overview

Localization in FMOD Studio is based on two complementary features: audio tables and programmer sounds.

Audio tables contain loose audio files that can be used by any programmer sound module.

Programmer sounds are used as placeholders for dialog lines within events. When played these sounds generate callbacks within the engine allowing the game to choose which dialog lines to play. The engine looks up the localized dialog line using an audio table.

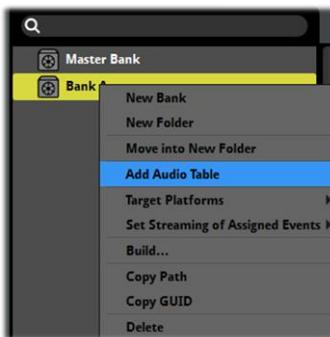
One of the benefits of this setup is reduced project complexity as one event with a programmer sound can stand in for many dialog lines in the game. Additionally audio tables are designed to scale to handle thousands of lines. There can only be one Audio Table per Bank.

Audio Tables

Audio tables are created in the Banks browser.

1. Right-click on the bank in the browser
2. Select 'Add Audio Table'

Selecting the audio display displays its contents in the editor.



Add Audio Table

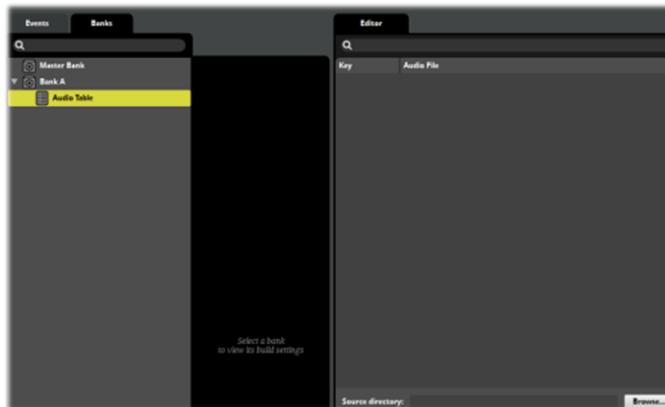
Audio tables pull their contents on-demand from a source directory every time the editor is shown and before every build. There are two ways to select a source directory.

1. Click on the 'Browse...' button to select a directory
2. Type a path into the source directory field and hit enter.

Note

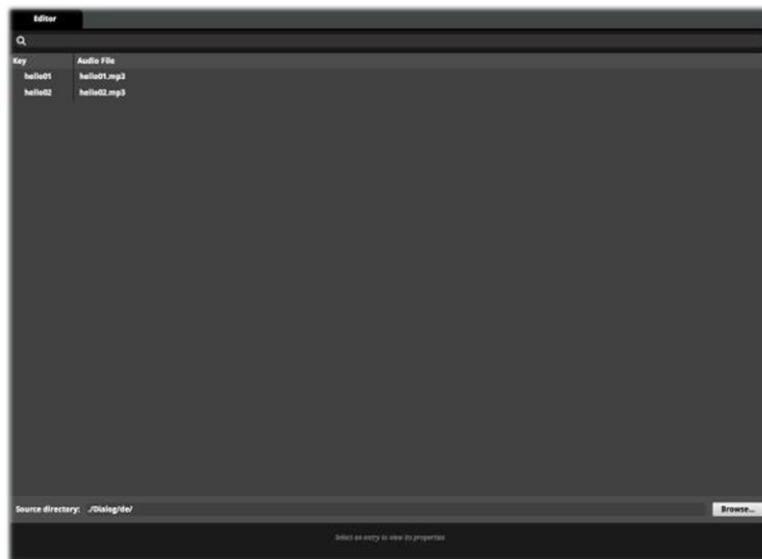
The path can include environment variables, for example %HOME% or \$HOME.

If the path is relative, i.e. ../source it is treated as relative to the project directory.



Audio Table Editor

By default, the audio table uses the filename as the key. This key is used by the engine to lookup the dialog line.

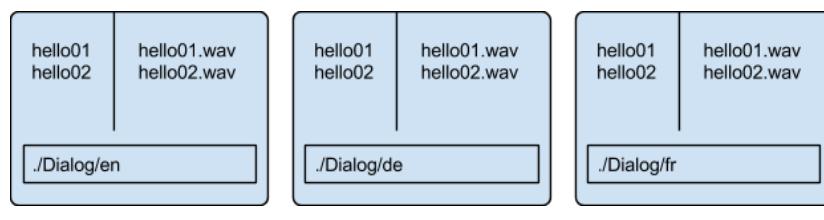


Alternatively, you can place a comma separated `keys.txt` file in the source directory to specify a custom key for each file.

For example:

```
DL0001,hello01.mp3
DL0002,hello02.mp3
```

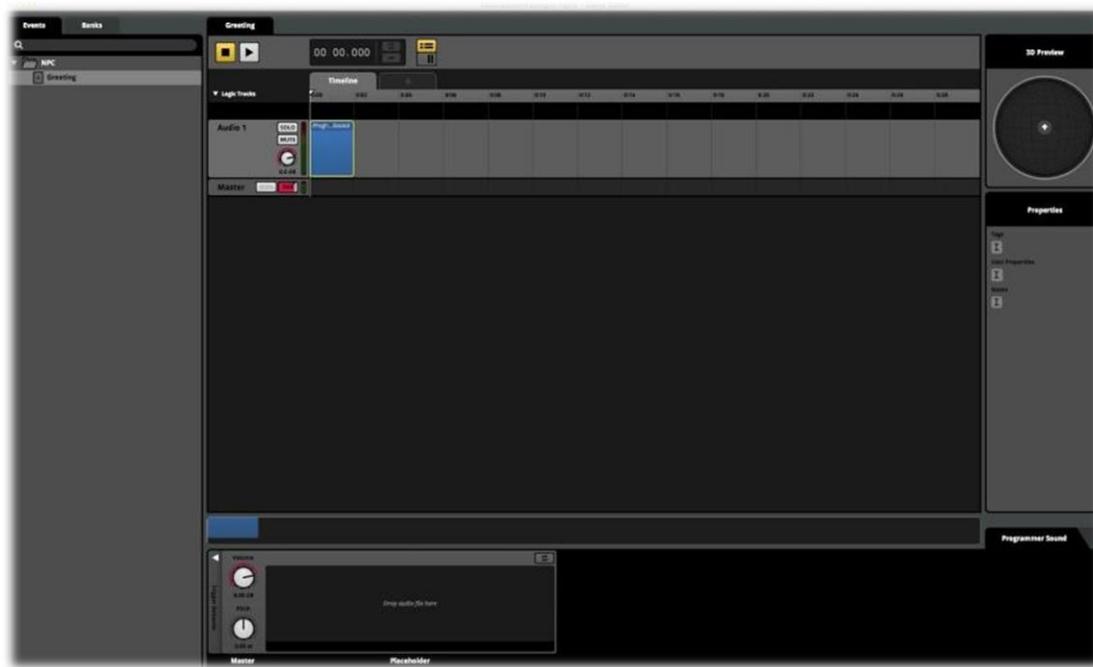
Each audio table holds the localized audio files for a language. By creating an audio table for each language, and using a language-independent key for each dialog line, the engine is able to lookup the localized audio file to play.



Localized audio files

Programmer Sounds

For general dialog, where the game is controlling which dialog line to play, simply create an event with a programmer sound.



Event with a Programmer Sound

You may want to create a generalized dialog event like this for different situations in the game. For example:

- An event with a 3D Panner for NPCs
- An event without a 3D Panner for voice overs

For localized emotes, where you'd like control over the line within FMOD Studio, you can drag and drop entries from an audio table into an event. There are two ways to do this.

Method 1

1. Select a single audio table entry
2. Drag and drop into an event to create a programmer sound

Method 2

1. Select multiple audio table entries
2. Drag and drop into an event to create a multi sound

These programmer sounds are created with names matching the keys from the audio table.

Localization API

To setup localization in a game you'll need to register a callback for programmer sounds.

```
// Declare callback for dialog
FMOD_RESULT F_CALLBACK dialogueEventCallback(
    FMOD_STUDIO_EVENT_CALLBACK_TYPE type,
    FMOD_STUDIO_EVENTINSTANCE *event_c_style,
    void *Parameters);
...
FMOD::Studio::EventDescription* greetingDescription = NULL;
```

```

ERRCHECK( system->getEvent("event:/NPC/Greeting",
    &greetingDescription) );

// Register callback used to look up localized asset
ERRCHECK( greetingDescription->setCallback(dialogueEventCallback) );

```

Within the callback you can look up the localized audio file using the dialog line key with `Studio::System::getSoundInfo`. This method searches all loaded audio tables for a matching key - starting with the most recently loaded.

```

// Start with Dialog_en.bank loaded
ERRCHECK( system->loadBankFile("Dialog_en.bank",
    FMOD_STUDIO_LOAD_BANK_NORMAL,
    &g_dialog_bank) );

```

For localized emotes, you can use the name of the programmer sound to lookup the localized audio file.

```

FMOD_STUDIO_PROGRAMMER_SOUND_PROPERTIES* properties =
    reinterpret_cast<FMOD_STUDIO_PROGRAMMER_SOUND_PROPERTIES*>
    (Parameters);

// Look up localized sound using programmer sound name
FMOD_STUDIO_SOUND_INFO info;
ERRCHECK( g_system->getSoundInfo(properties->name, &info) );
properties->subsoundIndex = info.subsoundIndex;

FMOD::Sound* sound = NULL;
ERRCHECK( g_low_level->createSound(info.name_or_data,
    info.mode | FMOD_NONBLOCKING,
    &info.exinfo,
    &sound) );
properties->sound = reinterpret_cast<FMOD_SOUND*>(sound);

```

Building and Implementing Projects

Introduction

The final stage in producing any game audio project is getting that content into the game. All the material developed in FMOD Studio needs to be built and then implemented into the relevant project to be usable.

There are several aspects to correctly exporting an FMOD Studio project that can impact its efficiency and overall functionality. Depending on the size and scope of your project some of these aspects may need to be planned early in development, and others can be applied when implementing your content.

Live Update

Live Update allows you to connect your project to your game and adjust the project settings while the game runs. When enabled, any change made to the project in FMOD Studio is immediately reflected in the game.

Note

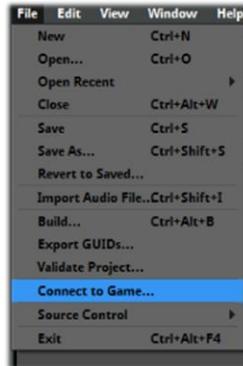
Audio assets cannot be added or removed from a project while live updating. If you do want to add or remove audio assets, you need to exit the game, rebuild the FMOD Studio project, put the new Banks into the game, and then connect live update again.

Enabling Live Updates

Your programmer needs to use the FMOD_STUDIO_INIT_LIVEUPDATE flag when calling `Studio::System::initialize`. This flag also automatically enables profiling.

Within FMOD Studio, all that needs to be done to connect to a running game is to load the FMOD Studio project, then select 'Connect to Game...' from the 'File' menu and specify the IP address of the machine running the game in the window that appears ('localhost' is acceptable).

Once connected, all meters in the mixer reflect the current state of the meters in the game.



Connect to Game...

Banks

Banks are collections of the assets, events and other content used in your project, to be built and used in your game. How you allocate events and set up your Banks can significantly affect the performance and efficiency of your project. There is no limit to the number of Banks your project can include.



The Banks Browser with the default Master Bank

By default all projects are created with a single Master Bank in the Banks Browser. FMOD Studio does not automatically assign any events to Banks (not even to the Master Bank) so if you run the build process without first assigning the Project events to a Bank, then your exported files contain no events.

The Master Bank

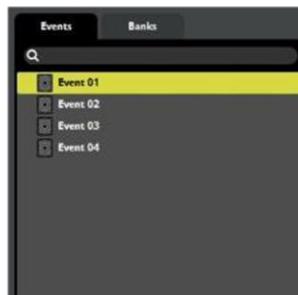
Each project in FMOD Studio requires a Master Bank. For a small project it may not be necessary to create additional Banks. The number of banks that's best for your project depends on the requirements of your project. In general if you require more than one Bank, then it's best not to place content onto the Master Bank unless your game project requires that content be loaded at all times.

In addition to being usable as a normal bank, the Master Bank contains metadata that is necessary for a game project to function, and must therefore be loaded at all times.

Assigning Events to a Bank

This section describes how to assign events to a specified Bank.

Accessing the Events within a Project



Accessing a Project's Events

1. Right click on the target event to access its context menu.

Assigning to a Bank

1. Select Assign to Bank from the context menu
2. From the sub menu select either one of the existing Banks or create a new Bank



Assigning an Event to a Bank

If you choose to create a new Bank a dialogue box allows you to define the name of the new Bank. Once defined, a new Bank is created in the Banks Browser and the targeted event is assigned to that Bank.

The exact same process can be used to assign an event to an additional Bank. Follow steps 1-3 and at step 4 simply select a different Bank from the list. Once an event has been assigned to a Bank a tick appears next to that Bank in the sub menu to indicate the association. A single event can be assigned to an unlimited number of Banks. When events are assigned to Banks, the Banks Browser updates to display which events are assigned to which Banks.

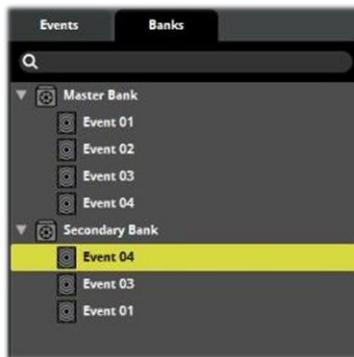
Note

When you assign an Event that is already in a Bank to another Bank it is added to the new Bank in addition to its association with the old Bank. If you do not wish the Event to be assigned to both Banks you need to remove it from one via the context menu.

Assigning Events within the Banks Browser

This section describes how to assign events to specified Banks within the Banks Browser.

Assigning Events within the Banks Browser



Accessing a Projects Events

1. Click on an event to select it.
2. Drag the target event onto a Bank to assign it to that Bank.

Events do not appear in the Banks Browser unless they have been assigned to at least one Bank. Once an event is visible within the Banks Browser it can be dragged within the Browser freely to assign it to other Banks.

Removing Events from Banks

There are two ways in which an event may be removed from a Bank.

Method 1



Removing an Event in the Events Browser

1. From the events Browser select the event you wish to remove from a Bank.
2. Right click on the event to access the context menu and select Assign to Bank.
3. From the submenu select the Bank you want to remove the event from (it should have a checkmark marking it)

Method 2



Remove an Event from the Banks Browser

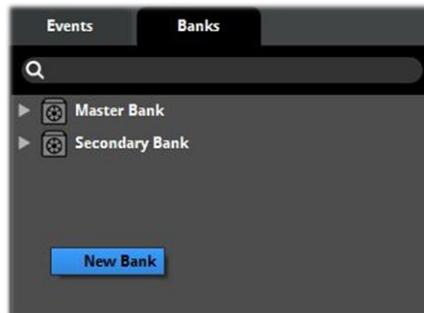
1. Within the Bank Browser select the event you wish to remove from a Bank
2. Right click to access the context menu and select Remove

Remember that if an event is not assigned to any Banks then that event is not included within the build process for export.

Creating New Banks

This section describes how to create a new Bank within the Banks Browser.

Creating a New Bank



Creating a new Bank

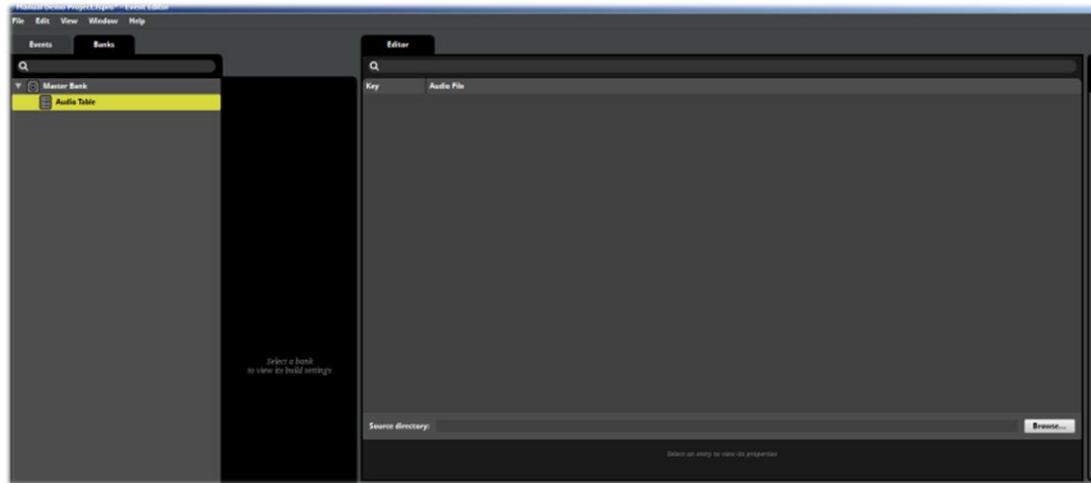
1. Right click within the Banks Browser and select New Bank from the context menu.
2. Type the name of the Bank you are creating.

Once a Bank has been created it can have events assigned to it either via the Events Browser or within the Banks Browser.

Bank Audio Tables

You can create an Audio Table for a Bank, which lets you select a path to a directory of loose assets that can be used when building the Bank.

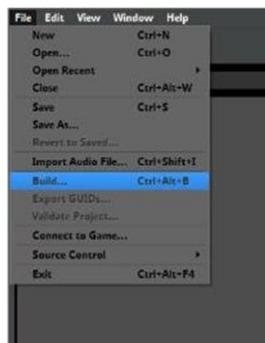
This allows you to build the audio files into a normal Bank without having to include them in a specific event. This can be useful if you're using programmer sound modules but still want the simplified asset loading and project management that Banks provide.



Bank Audio Table

The Build Process

Once all the events have been assigned to the relevant Banks and all Bank Properties have been defined as needed, the Build process can be run to export all the files. This is a single step process that can be started by selecting Build from the File menu.



Building a Project

The time a project takes to build is dependent on the size of the project, how many sound files there are, how big those sound files are, what formats it needs to be built for, what formats the banks are in, and how many Banks are included. Smaller projects may only take a few seconds, while a very large project may take hours.

When the Build process is run, FMOD Studio packs up all the assets and relevant project files into a series of new files. Every Bank and formal combination within an FMOD Studio project results in a '.bank' file, and there is also a '.strings.bank' file that contains both the name of each event and its relevant GUID, which allows an audio programmer to work with the project elements using either the name or the GUID depending on what works best for the project.

If multiple platforms have been selected under the Banks Browser then the Build process creates multiple folders within the project folder and produce a set of banks formatted correctly for the selected platforms.

Note

While FMOD Studio allows Banks to be created for all of the supported delivery platforms, it is important to know that the build process cannot complete successfully for certain platforms unless the appropriate SDK is installed. This limitation is to fulfil the requirements of the various platform manufacturers' licence agreements.

Setting up Control Surfaces

Configuring the Device

FMOD Studio supports the Mackie MIDI control surface protocol. The following section provides instructions for how to enable the Mackie protocol on some common devices.

MCU Pro

To run the MCU Pro in Mackie mode, hold down the first two SELECT buttons when booting the device. Select the Mackie option using the dials when prompted via the LCD interface.



Mackie MCU Pro

Adding the Device

To configure a new control surface, go to the Preferences dialog using the **[Edit>Preferences]** menu. In the Control Surfaces tab, select a control surface type from the context-sensitive menu. Select the MIDI port(s) required for the surface. FMOD Studio connects to the device as soon as the MIDI ports have been selected.

Ensure that both input and output ports are set to the same value. Thoroughly read the setup information for your control surface to ensure you have setup drivers and software support correctly prior to attempting connection to FMOD Studio.



Studio Preferences Pop-up Window

Note

Once the control surface is set up correctly and the FMOD Studio preferences are also set up correctly the controller should display a message that says **FMOD Studio**. If it does not display this message then there is likely something that has not been set up correctly. Review the steps provided in this section to help you correctly set up the connection between the Nucleus and FMOD Studio.

SSL Nucleus



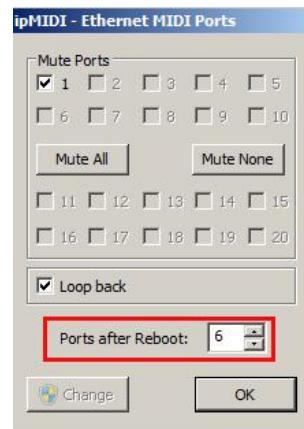
FMOD Studio integrates directly with the SSL Nucleus control desk

Software Set-up

Ensure that all setup programs that are provided on the install disk for the Nucleus are installed on your computer. There are several steps involved in getting the Nucleus to connect properly with FMOD Studio.

ipMIDI

The first step involves adjusting the setting for the ipMIDI utility before starting the Nucleus setup process. By default ipMIDI is set to 2 channels. This needs to be adjusted to 6 channels for optimal performance.



Optimal ipMIDI setup

Note

By default the Loop Back box is ticked. We recommend you un-tick this option.

Adding the FMOD Studio Profile to the Nucleus

First ensure that the SSL Nucleus is connected to your computer via the Ethernet port. The USB connection is to use the Nucleus as a sound card, the Ethernet connection is used for Digital Audio Workstation (DAW) functionality.

Open the Nucleus Remote software and connect to the device.



Edit Profiles setting

Select the **Edit Profiles** button and select the **Restore** option.



Edit Profiles pop-up window

In the Edit Profiles pop-up window select the Restore button, a standard file browser window opens. Navigate to the FMOD Studio folder on your computer. The FMOD Studio SSL Nucleus Profile.zip file can be found in the root of the FMOD Studio folder. You do not need to unzip this file, just select it directly through the Nucleus set-up program. The program imports and sets up these files as the next step.



Nucleus profile zip file

The FMOD Studio profile is displayed in the profile editor and can be selected



The FMOD Studio profile

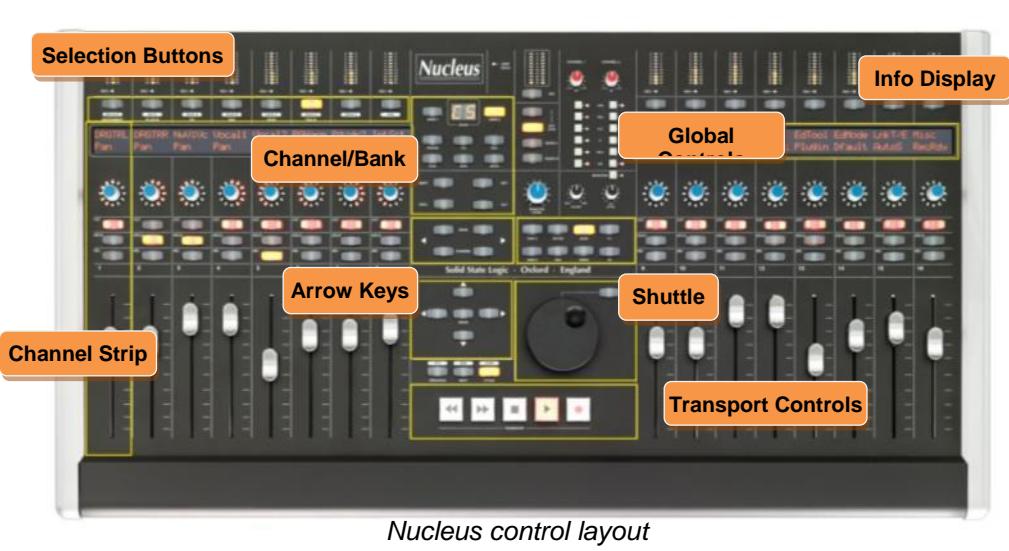
Select **OK** and return to the main Nucleus Remote window. Select the **FMOD Studio** profile from the **Selected Profile** dropdown and reboot the Nucleus if prompted.



Select the profile from the context-sensitive menu

Nucleus Control with Studio

While the SSL Nucleus is fully programmable there is a basic default setup in regards to FMOD Studio function. By default the Nucleus links its channels directly to the Audio Tracks within Studio Events as well as some standard global.



Nucleus Profiles

The information for profiles and user set-ups are saved globally on each relevant computer and the Nucleus itself has an SD memory card that saves Profiles that are imported to the hardware.

Common Controls

The Nucleus includes control strips for 16 tracks. The left hand side set of 8 are routed through midi channel 1 the right hand side 8 are routed via midi channel 2. If an FMOD Studio event is created with more than 16 tracks the Nucleus defaults to controlling the first 16 tracks created. You can change which tracks are controlled if required.

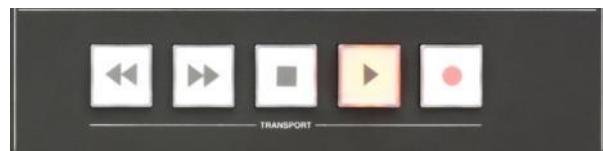


Bank/Channel Selector

The Bank / Channel selector buttons allow for the Nucleus to change which audio tracks are assigned to the available channel strips. The channel buttons move the selection up or down by a single track. The Bank button shifts the selection by an entire group of 16 Tracks.

Transport Controls

The Nucleus control surface has the standard transport controls used on any DAW. The only exceptions are the Forward and Rewind functions, as FMOD Studio does not have traditional linear functionality at its core. These options are currently not linked to any settings.



Transport Controls

- Reverse Button: No Setting currently
- Forward Button: No Setting currently
- Stop Button: This stops playback when pressed once, but does not reset the cursor position. When pressed twice this returns the cursor to the beginning of the timeline
- Play Button: The play button always plays the event from the beginning of the timeline

Selection buttons

The Nucleus has a range of selection buttons along the top of the unit. These buttons have specific settings when working with FMOD Studio. Activating the different buttons alters which settings are mapped out to the Channel Strip controls and the information in the display screen. The Value Dials can be used to manipulate displayed values, but pressing the value dial as a button can also access extra settings.



Nucleus Selection Buttons

Instrument

This button accesses the values of Parameters. Holding down the Instrument button allows you to select between all available Parameters using the Dial controller. Once a Parameter has been selected and the Instrument button is released the Dial controller is used to alter the Parameter values. Events with only the timeline view are not affected.

Plug In

The Plug In button displays a list of all plugins added to the Deck for the active event Audio track.

By default an audio track contains only a Fader control. As more plugins are added to the Deck the Nucleus display alters to indicate all plugins present. In this view any of the Dial controls associated with a Plug In can be pushed to gain access to that plugin. The Nucleus display alters to indicate all the relevant controls for that plugin and its values can now be accessed through either the control Dial or the Fader.

To return to the plugin selection display, simply press the plugin button again.

Note

The range of controllers and they type of display and values is entirely dependent on the Plugin being controlled. The selection of Plugins provided with FMOD Studio covers a wide range of effects and controllers and each one has a unique set of controllers. There is a logical link between the active controllers on the Nucleus and the visual display of each Plugin in Studio.

EQ

EQ directly links to the EQ plugin for the active audio track. It displays the relevant property values and allows direct control of those properties via the control dials.

Pan

The Pan button accesses a range of pan controls for the active audio track. FMOD Studio has a range of panning options and this setting can access all of them.

For Surround Pan events, the pan control, when first activated, provides control Properties to alter the listening position within the circular field of the 3D pan environment. The 3D Pan Preview in FMOD Studio displays the relevant graphical representation of the pan position as values are altered on the Nucleus.

Pressing the Pan button a second time for a 3D sound opens the Extent setting. This represents the range from center to the outer boundary of the 3D Pan Preview. Altering this value moves the signal from center point where the signal is strongest to the outer edge where it is weakest.

For audio tracks set to stereo the pan control allows for value adjustment from center to either 100% left or 100% right hand side channel assignment.

Audio tracks set to mono default to equal left and right channel values and cannot be changed.

Send

The Send button displays all Sends created within the Deck for the active audio track. It allows individual control of the values of each Send displayed. Audio tracks that do not include Sends display nothing when the button is pressed.

Track

Activating the Track button displays all the included tracks within an event including the Master Track. Each Track is assigned to a control strip and the Nucleus display indicates the volume levels of all audio tracks. The Faders can be used to directly control the Volume values of each audio track. This setting can be used to readjust the tracks that are being displayed by the Nucleus for control by using the Channel/Bank controls in the center of the Nucleus surface.

While in Track mode the arrow keys can be used to scroll the display in FMOD Studio up and down within the event as well as to the left and the right. If the entire event is displayed in the default screen view these Properties have no effect.

Flashing display lights on the Nucleus Surface indicate which track has been selected as active.

Send E

Pressing this button automatically overrides all Mute and Solo Properties active in the event. This allows you to instantly monitor the event with all tracks present and active. Releasing this button returns the event to its actual state with any active Solo or Mute Properties reactivating.

Pan

The Pan selection button above track 8 currently has no function.

Track 9 Selection Button

This button is a shortcut to display the events view in FMOD Studio. Pressing this button multiple times cycles through all existing event windows within the project in order.

Track 10 Selection Button

This button is a shortcut to display the Mixer View in FMOD Studio. Pressing this button multiple times cycles through all existing event windows within the project in order.

Tracks 11-16 Selection Buttons

These buttons currently have no function.

The Shuttle Wheel



The Shuttle Wheel

The Shuttle Wheel control on the Nucleus directly controls the cursor in FMOD Studio. As its name suggests, it shuttles the cursor backwards and forwards in direct relation to how it is turned.

Holding the toggle button with Shuttle control changes its function to a visual display zoom within FMOD Studio. This allows quick horizontal zooming in and out of the audio tracks in an event.

Global Controls

There are a series of global control buttons on the Nucleus with a range of functions.



Global control buttons

DAW 2 and DAW 3

The SSL Nucleus is capable of controlling up to three Digital Audio Workstation setups and swapping between them.

The DAW2 and DAW 3 buttons change the control surface to control secondary and tertiary applications. This allows Sound Designers to quickly jump between different aspects of a project across multiple applications and even multiple computers as long as all the relevant setup Parameters have defined correctly.

Note

You can control projects over multiple computers. The SSL Nucleus set-up programs need to be present and set-up correctly on all machines. ipMidi also needs to be set up.

For a second computer use midi port 3 and 4 and a third, use 5 and 6.

Enter

This button simply allows selection to be confirmed from the control surface. Any setting that would require the Enter key to be used can be triggered with this button.

Save

This button saves the current FMOD Studio project directly from the control surface.

ESC

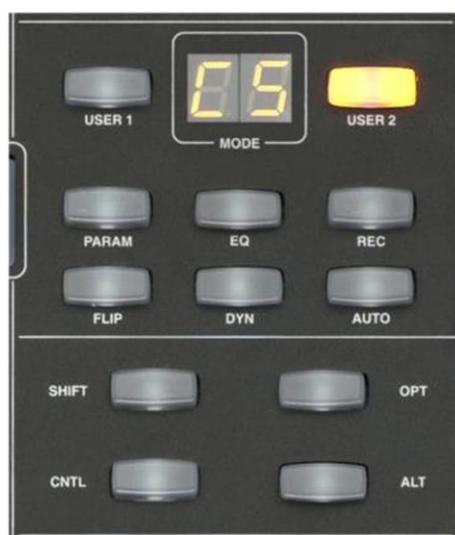
This button replicates any of the keyboard ESC functions in FMOD Studio.

Undo

Undo allows the undo or keyboard shortcut CTRL+Z to function directly from the control surface.

F2

The F2 button maps to the Open Mixer Window command.

Miscellaneous Controls

Miscellaneous controls

Flip

The Flip button swaps the Properties assigned to the Dial controller and the Fader controller. This allows the default volume control of the fader to be swapped for Parameter control, effects value controls and all other controls usually assigned to the control Dial. This can be convenient for easy manipulation of multiple Parameter controls for auditioning and is very useful for teaching and other presentation purposes.

User 1 & User 2

The SSL Nucleus is a fully programmable control surface and can support up to additional 2 user defined profiles for each loaded profile setup. These buttons switch between the default profile and User Profile 1 and User Profile 2.

Channel Strips



SSL Nucleus Channel Strip

Default Channel Strip Behavior

The Default behaviors of the Channel Strips on the Nucleus are as one would expect of any standard DAW control surface.

- **Display:** IDs the assigned Audio Track in Studio and displays the volume level
- **Value Dial:** Controls the Volume level of the assigned Audio Track in Studio
- **Cut:** Mutes the assigned Audio Track in Studio
- **Solo:** Solos the assigned Audio Track in Studio
- **Select:** Selects the assigned Audio Track in Studio and makes it the active track
- **Fader:** Controls the Volume level of the assigned Audio Track in Studio

The Fader controls are motorized and respond to automation value changes in FMOD Studio events.

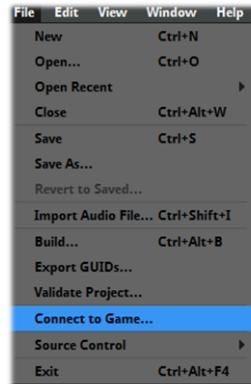
Profiler

FMOD Studio includes a Profiler tool that allows you to assess the performance of your projects. The Profiler can be used to check performance of events and Projects directly in FMOD Studio, and can also work when a project is linked to a game running on a computer or console development kit to assess performance at runtime.



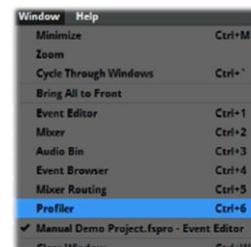
The FMOD Studio Profiler

To start using the Profiler, you need to connect Studio to a game project. Open your project, run your game, and go to “**File > Connect to Game**”. The editor border changes to red when the game is connected.



Connect to a Game

Once you have connected to your game, you need to open the Profiler window. Go to “**Window>Profiler**” or use shortcut **Ctrl+6**.



Open the Profiler

The Profiler has many of the same interface areas as other FMOD Studio windows.

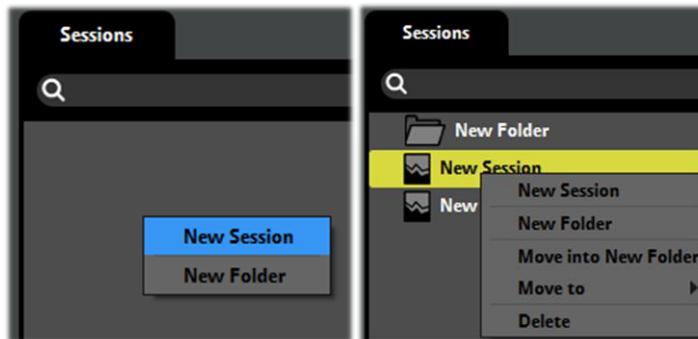
The Session Browser

The Session Browser allows you to create and administer Sessions. Its context menu contains the following items:

- New Session: Creates a new session with all Buses scoped in.
- New Folder: Creates an empty Folder for session grouping and administration.

Right Clicking on an object in the Session Browser provides additional context menu items:

- Move into New Folder: Creates a new folder and allocates the context to the targeted Folder.
- Move to: Allocates the current context to the targeted Folder.
- Delete: Deletes the object from the project.



Session Browser context menus

Adding a Bus to a Session

Buses can be scoped into a Session simply by dragging them from the Mixer Routing Window onto the Editor area. Any type of Bus, as well as events, can be added to a Session.

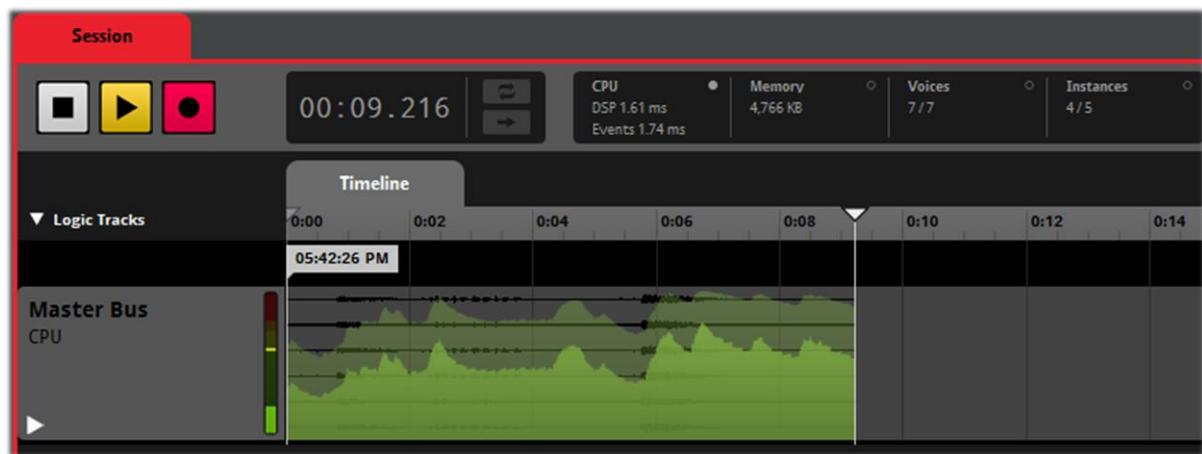
1. Activate a Session by clicking on it in the Session Browser area.
2. Display the Mixer Router window.
3. Drag the desired Bus from the Mixer Router Window into the Session Editor to create a new Track.



Close up of a Session Track

Recording a Session

Once you have created a blank profiler session, you can hit the record button in the transport bar. Waveform data appears on the master track, as well as a graph overlay.



Profiler Session

Once you're finished capturing data, hit the stop button in the transport bar. You can then explore the recorded session.

Note

By default, a track for each recorded event type is automatically added to the session once the recording is completed. You can optionally scope in mixer strips (or events) to see information pertaining to those items by dragging and dropping them into the profiler session.

The Profiler records several pieces of data as it is recording.

Master Bus

The Master Bus Track captures Peak and RMS output data. These indicate the output levels of all signals playing through the Master Bus in FMOD Studio.

The Disclosure Triangle on the Master bus expands the tracks to indicate both the output levels and the CPU usage for the events being played. Moving the cursor over the output curve displays the exact value for the point at which the cursor is positioned.

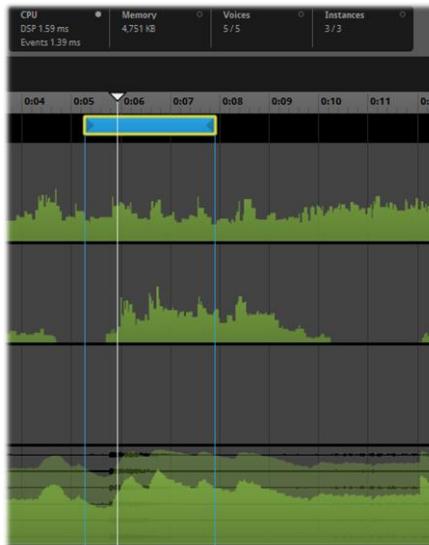
Note

The data displayed in the Profiler is vertically scaled to maximize space and visibility, so the maximum peak of a data stream is at the top of the display area, but this peak may only represent one or two percent of capacity. A subsequent greater peak results in the entire data display being scaled appropriately. For this reason it is important to use mouse hover to gather accurate data values.

Each Track displays the relevant data for its associated Bus. In this way it is easy to scan though Track data looking for anomalies and performance issues. Session data is useful for debugging and optimization purposes.

Playback

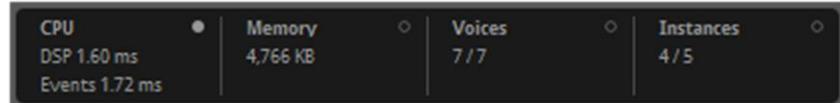
Once you have recorded a session in the Profiler, you can play back the recorded session, as you would with an event, and hear the recorded output from your game. Areas that show instantaneous values (such as the transport bar and 3D View) corresponds to the playhead position. You can also use loop regions to focus in on parts of the recording.



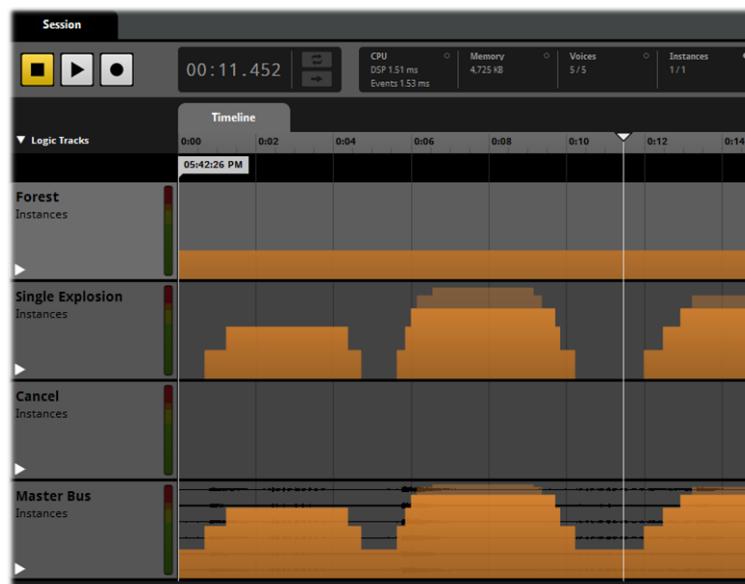
Profiler Loop Region

Graphs

Each track in the profiler can be used to view information about that item. You can switch data points using the master profiler statistics in the transport bar. For example, clicking on the **Instances** section switches to the corresponding graphs.



Data Point Options for Graphs



Profiler Graphs

Moving your mouse over graphs displays the value at a particular point.

The current graphs available are:

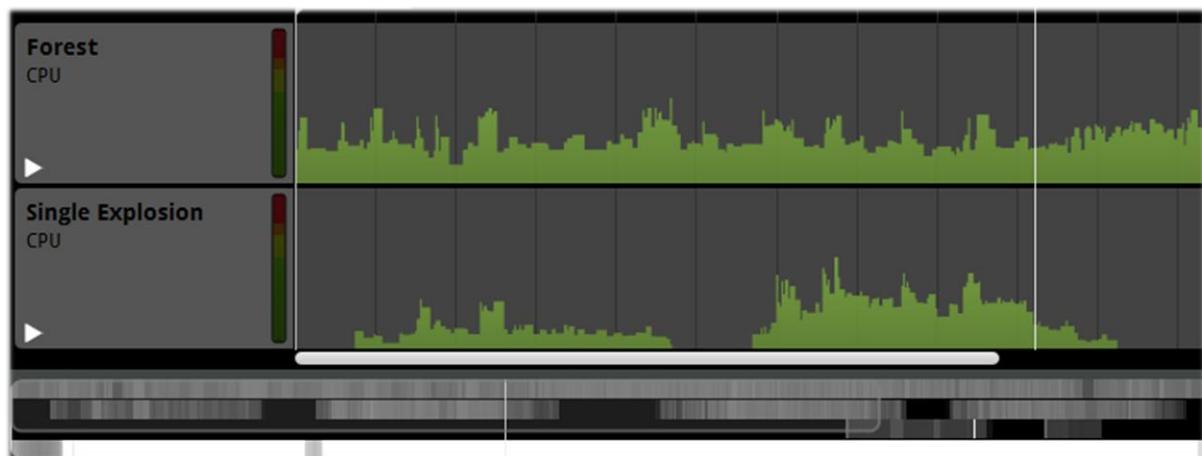
- CPU (ms): Available for all buses, including the Master Bus. This represents the CPU time used by the bus. On the Master Bus, this represents total CPU time used, and the additional background graph represents the time spent updating events and mixer DSP.
- Memory (KB): Available only for the master bus, where it represents global memory use.
- Voices: Available for all buses, including the Master Bus. The opaque graph represents the number of real voices used by the bus, while the transparent background graph includes both real and virtual voices. On the Master Bus, both graphs represent voices used globally.
- Instances: Available for events and the Master Bus only. The opaque graph represents the number of real event instances, while the transparent background graph includes both real and virtual event instances. On the Master Bus, both graphs represent global event instances.

The Birdseye view displays a summary of the active graph type for the entire session.

Note

Virtual elements are included because they take up almost as much system resources as non-virtual elements. Even if they're not included in the mix currently, they could potentially become audible again in future, so FMOD has to process them just as if they were still playing audibly.

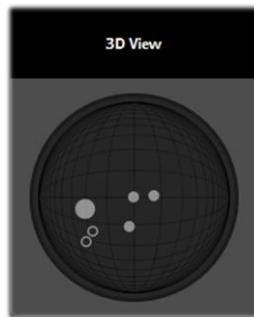
The Birdseye view displays a summary of the active graph type for the entire session.



Birdseye View Summary

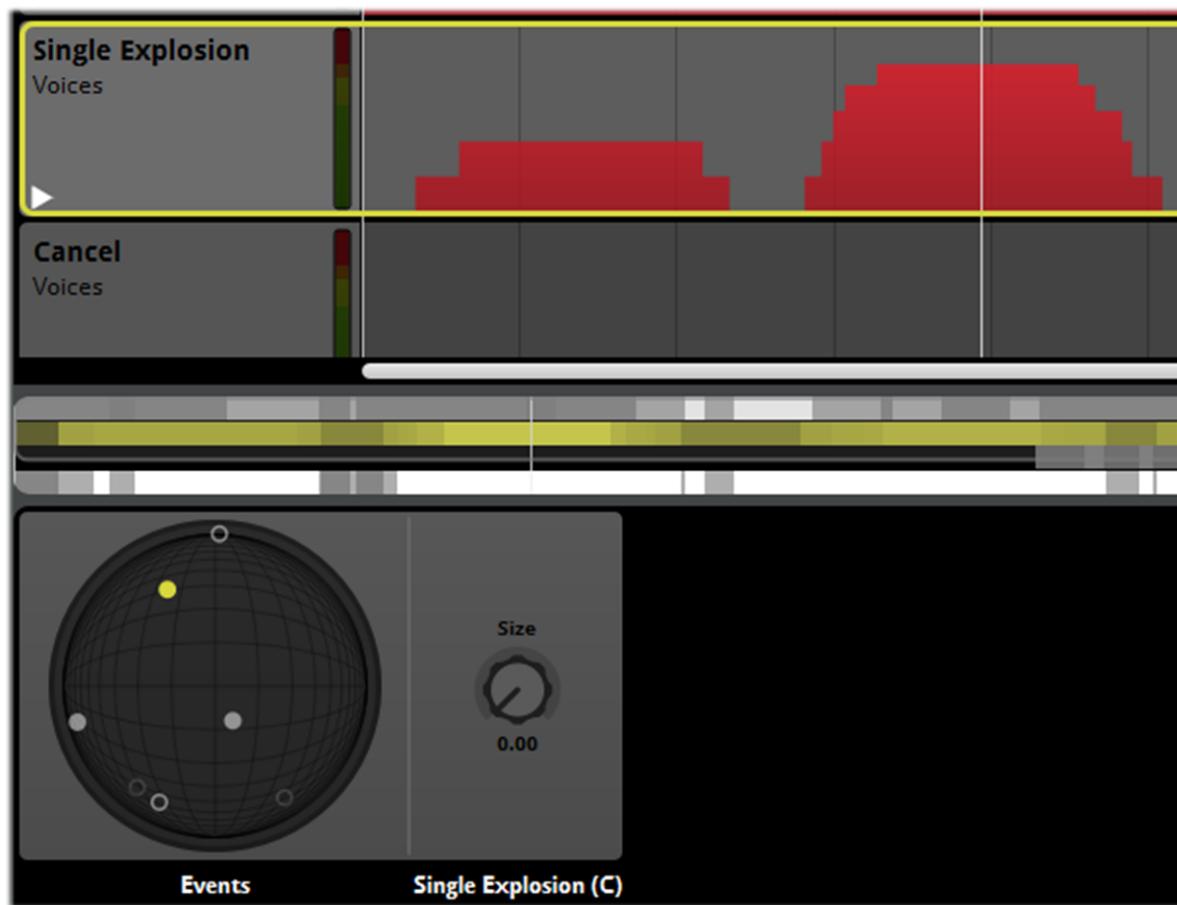
Event Instances

The profiler provides additional information about each event instance. A 3D view is available of the entire scene in the top right of the profiler window.



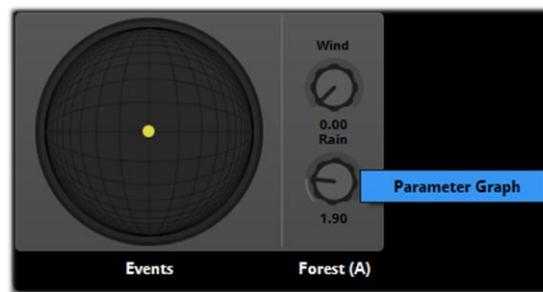
3D View for Event Instances

Playing events are shown as filled circles, while virtual events are hollow circles. Stopped events become dimmer and fade over time, which assists with finding short-lived events. Clicking on an event in this view highlights its associated track. This also presents a secondary 3D view in the deck which shows only events of a given type.



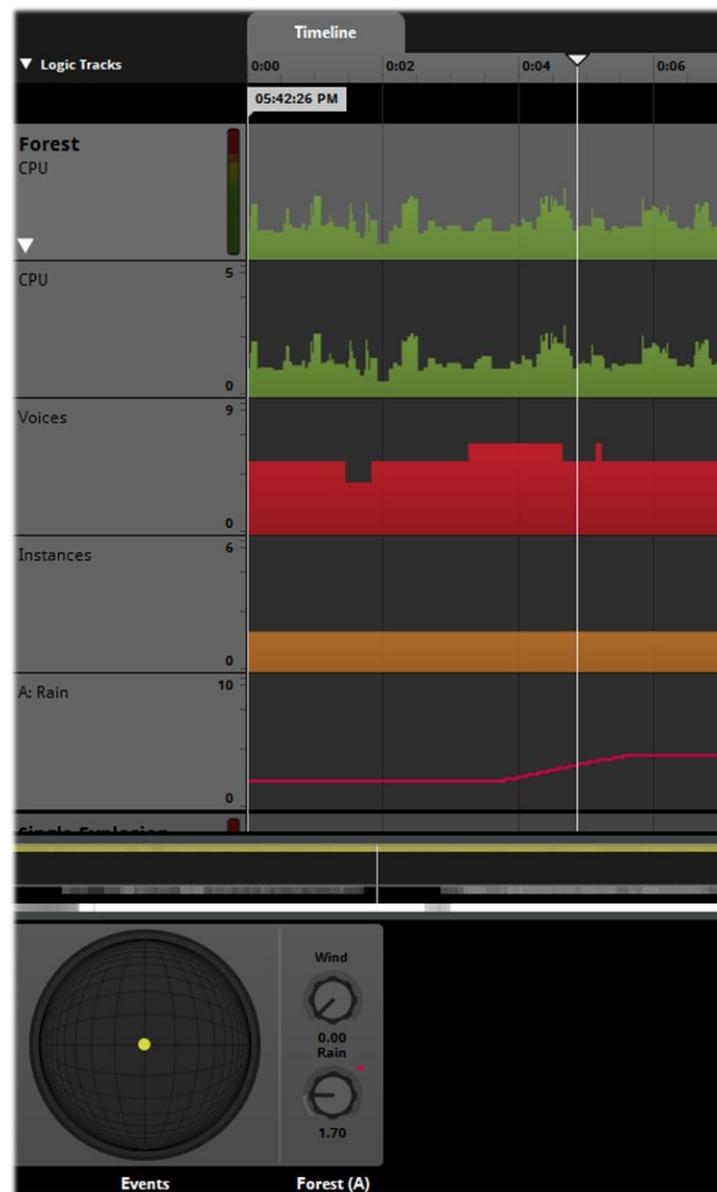
Highlighting Events in the Profiler

From here, you can see an event's Parameters over time. Right click a Parameter dial to add a graph for that particular event's Parameter.



Right click on the Parameter Dial to select Parameter Graph

This is added as a subgraph in the multitrack view.



Multitrack View with Parameter Graph

Loudness Meter

A loudness meter module measures the integrated loudness, loudness range and true peak of the game's output at its position in any signal chain it is added to.

Loudness Meter Module

FMOD Studio provides a Loudness Meter Module that can be added to the Deck region of event tracks and mixer buses. The loudness meter tracks levels and displays a range of data to assist in balancing your project.

Standards of loudness are a complex aspect of audio production for media and the games industry has many people involved in attempts to provide useful information to developers, allowing them to produce contact that is consistent with other forms of media.

The loudness meter included in FMOD Studio provides data that is consistent with the EBU recommendation R128. <http://tech.ebu.ch/loudness> and the ITU's BS. 1770-3 recommendation <http://www.itu.int/rec/R-REC-BS.1770-3-201208-I>. This is a series of standards agreed upon within Europe and the US to guide broadcasters in the production of media.

The general concept is that measurement of sound pressure levels and values defined by decibels does not provide sufficient information for developers to be able to accurately balance the audio for a project. The way humans perceive sound is more of a psychoacoustic phenomenon than something that can be perfectly mathematically measured. There are many reasons for this, but the following describes a simple issue with audio measures.

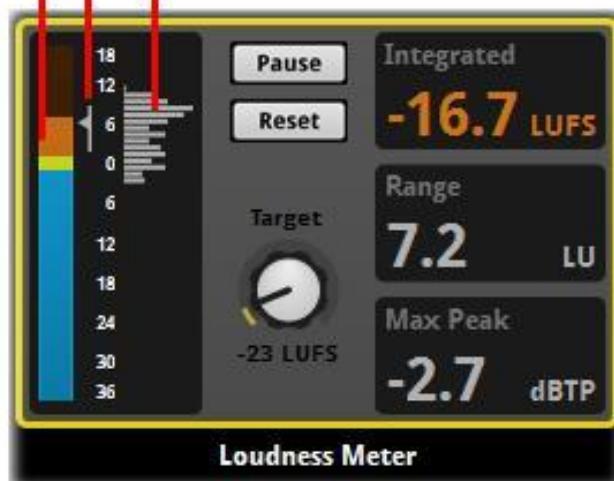
Humans perceive the range of frequencies differently. A very low frequency sound at 60dB sounds softer than a high frequency sound. For this reason a flat measurement of dB output is very limited in its usefulness. Loudness Units or LU are a measurement standard that provides a useful way of measuring the overall perceived loudness of a signal.

This article provides further information on this method of measuring audio output:
<http://auphonic.com/blog/2012/08/02/loudness-measurement-and-normalization-ebu-r128-calm-act/>

Integrated Loudness

Loudness Range

Loudness Histogram



The Loudness Meter

The Loudness meter is useful for capturing an overview of a projects output. While it does update its measured values constantly it is designed less for moment by moment monitoring and more to provide a summary of the overall output performance of a project.

Integrated Loudness

The *Integrated* display provides an overall summary of the output measured in Loudness Units with reference to Full Scale (LUFS). This shows a running total of the measured loudness since playback was started.

Note

You could essentially leave the Loudness meter to track an entire piece and return afterwards to examine the data. If the Integrated display values are displayed in blue it means the overall levels are under the defined Target level, a display in Yellow indicates the levels are within +/-1LU of the Target level and a value displayed in Orange indicates the levels exceed the Target loudness. When used as part of the Profiler the profiled output would indicate exactly where these levels fell below, matched or exceeded the defined target levels.

Loudness Range

The Range display indicates the overall dynamic range in LU that the project has produced while being tracked by the Loudness meter. This information can be useful for a variety of reasons. Mobile platforms may need a smaller dynamic range to avoid signals being obscured on smaller playback devices while projects developed for high end system may wish to maximize their use of dynamic range.

Max Peak

Audio software applications have a level to indicate where digital audio signals Peak and potentially distort the audio signal. This usually occurs at 0dB and signals that are close to 0dB are considered undesirable. An issue with digital audio signals is that generally it measures signals by plotting the position of each sample location. While this is generally accurate it is possible for a signal to exceed 0dB between samples. The unit dBTP measures Decibels True Peak and so more accurately estimates the peak level when the signal is converted to analogue during playback. A headroom of 1dB is prescribed in loudness standards so this meter turns red when the signal exceeds -1dBTP.

Target

This property allows you to define the desirable overall loudness. This can be set to any level between -24 LUFS and -12 LUFS and defines what information the Integrated Display and bar meters show. Double Clicking on the Dial sets the Property to a default value of -23 LUFS. All levels in the bar meters and Profiler are shown relative to the target value.

Pause

When Paused the Loudness Meter ignores output level data and does not record it for inclusion in its summary.

Because the Loudness Meter tracks data over time and provides a summary of the output it may not always be desirable to include all data. Pausing allows you to temporarily cease recording, alter aspects of their project that may be causing extreme output values, and then resume recording to resume collecting the data.

Reset

The reset button clears the Loudness Meter of all data to allow it to start fresh with a new set of data.

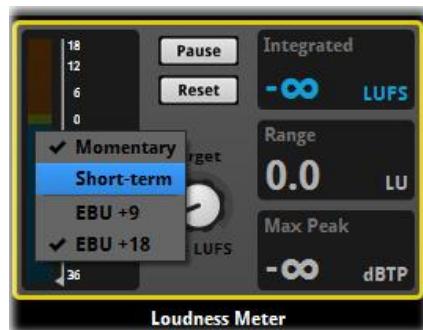
Bar Meters

The Bar Meters track and measure the output of the Project. By default the Meter measures Momentary information and displays a range of -36 to +18 LU. All values shown are relative to the set Target loudness.

Information on EBU +9 and +18 meter ranges:

<http://tech.ebu.ch/docs/tech/tech3341.pdf>

The Context menu for the Meter allows the Meter to switch from Momentary to Short term signal tracking. Momentary is a 400ms response while Short Term is a 3 second response. The Context menu can also allow the meter to display a range from -36 to +18 LU or -18 to +9 LU. This context menu also switches what is displayed in the Profiler.



Bar Meters Context menu

Meter Signal Peak and Range Indicator

The Output Meter includes an integrated loudness indicator as well as a range indicator line. The Arrow shows the integrated loudness relative to the Target loudness while a line extends from the arrow both up and down, which tracks the overall loudness range.

Loudness Histogram

As part of the Output Meter there is a graphic display that provides a quick reference for the spread of loudness during playback. Each bar indicates a range of 1LU and the overall display is scaled to show the playback duration of each loudness range relative to each other.

Scripting

FMOD Studio's scripting feature lets you control Studio and edit projects by using javascript. Commands can be entered into Studio's built-in scripting interface, written into files that can be loaded either when FMOD Studio is launched or while it is running, or via TCP/IP connection.

The main use of scripts is to automate and simplify processes frequently required by your workflow, such as creating events with certain specific kinds of content required by your project, or making the same specific change to a large number of events.

Script files are automatically evaluated every time you load a project. You can also force FMOD Studio to reload script files by selecting 'Reload' from the 'Scripts' menu. In either case, FMOD Studio reads scripts from any files with the .js extension in the following locations:

- User scripts directory:
 - %localappdata%/FMOD Studio/Scripts on Windows
 - ~/Library/Application Support/FMOD Studio/Scripts on Mac OS
- Built-in scripts directory:
 - %fmod_install_directory%/Scripts on Windows
 - %fmod_bundle%/Scripts on Mac OS
- Project scripts directory:
 - *fspro_root_directory*/Scripts

Some example scripts are included with FMOD Studio, and can be found in the built-in scripts directory.

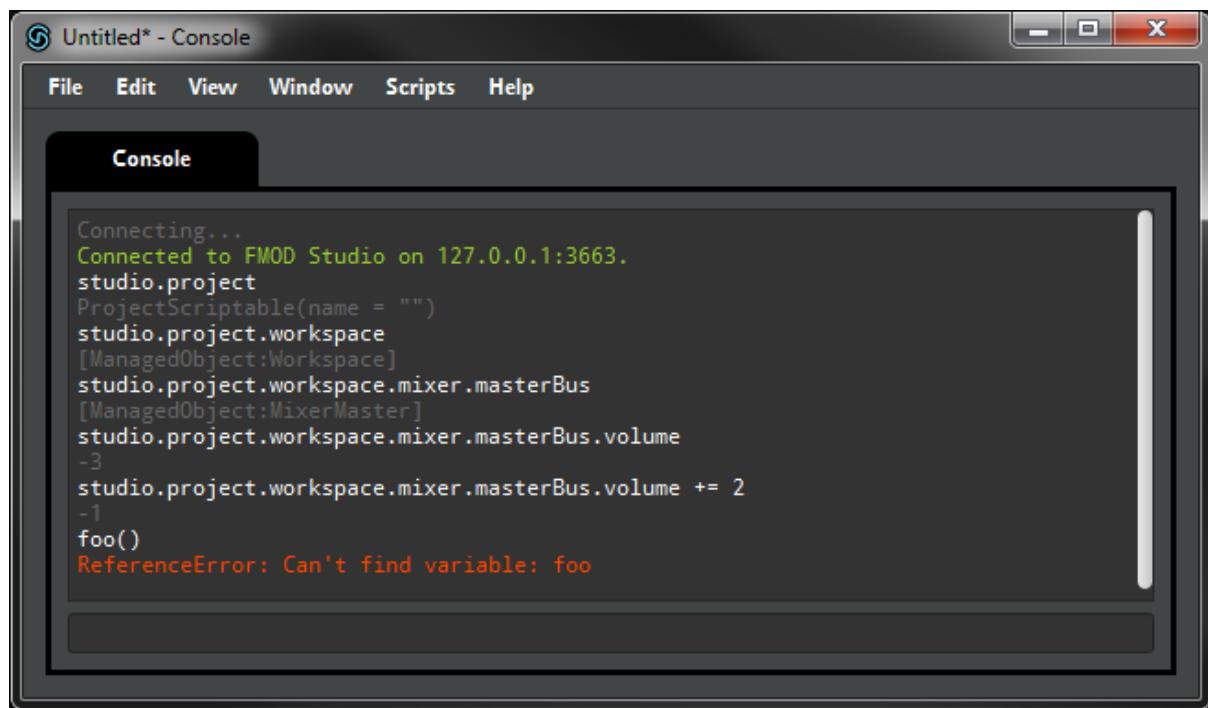
Note that this section of the manual assumes some basic knowledge of programming concepts and JavaScript.

The Shell Interface

Studio also supports evaluating script code in a REPL style interface. This can be used to control Studio remotely from an external application, such as a level editor.

Communicating with Studio is done via a TCP/IP connection over port 3663. Any data transmitted to FMOD Studio will be interpreted as JavaScript encoded as UTF-8, and any data received by clients from FMOD Studio should be interpreted as UTF-8 strings, representing the evaluated code.

As an alternative to sending commands from an external application, you can also type them directly into FMOD Studio's Console window. To access this window, select 'Console' from the 'Window' menu.



The Console Window

Scripting Reference

Much useful FMOD Studio scripting deals with ManagedObjects. The Project Objects section, below, describes these in detail.

Global objects

The following APIs are all available within the global object context (e.g. `console.log("hello");`).

API	Description
<code>console.log(msg)</code>	Logs a message. Alias for <code>studio.system.print</code> .
<code>console.warn(msg)</code>	Logs a warning. Alias for <code>studio.system.warn</code> .
<code>console.error(msg)</code>	Logs an error. Alias for <code>studio.system.error</code> .
<code>console.dump(object)</code>	Logs all members of object.
<code>alert(msg)</code>	Logs a message in a popup dialog (blocking). Alias for <code>studio.system.message</code> .

Studio objects

The following APIs are available within the `studio` object context (e.g. `studio.project.save();`).

API	Description
<pre>version</pre>	<p>An object containing version information for the Studio tool:</p> <pre>{ productVersion: [number], majorVersion: number], minorVersion: [number], changelist: [number], }</pre> <p>This object also implements <code>toString()</code> which returns the version in a human-readable format (e.g. "Version 1.00.00 (100000)").</p>
<pre>menu.addMenuItem(description)</pre>	<p>Registers a menu item under the 'Scripts' menu, based on the description object:</p> <pre>{ name: [string] (required), execute: [function] (required), keySequence: [string], isEnabled: [bool] (dynamic), isChecked: [bool] (dynamic), subMenuItems: [array(description)] (dynamic), }</pre> <p>The name and keySequence properties must each be a string, but other properties can specify a function to be called dynamically, which returns a value of the expected type (for example, you may want to reevaluate isEnabled based on what's selected). Using a '\' character in the item name will place it in a submenu. Adding an item with a name containing all dashes ('-' character) will result in a separator being added.</p> <p>name is the name of the menu item that appears in the menu.</p>

	<p><code>execute</code> is the function to be called by the menu item.</p> <p><code>keySequence</code> is the keyboard shortcut for the menu item.</p> <p><code>isEnabled</code> determines whether the menu item appears normally or grayed out.</p> <p><code>isChecked</code> determines whether a checkmark appears next to the menu item.</p> <p><code>subMenuItems</code> specifies the items that appear in the menu item's associated submenu, if any.</p>
<code>menu.removeMenuItem(name)</code>	<p>Removes a menu item created by <code>menu.addMenuItem()</code>. Name is the same name specified in the description passed to <code>menu.addMenuItem()</code>.</p>
<code>ui.showModalDialog(description)</code> <code>ui.showModelessDialog(description)</code>	<p>Displays a modal dialog. In the case of <code>showModalDialog()</code>, the function returns once the dialog is closed. In the case of <code>showModelessDialog()</code>, the function returns once the dialog is shown. In either case, the layout of the dialog is based on a widget description object, which contains the following attributes:</p> <ul style="list-style-type: none"> • <code>widgetType: [ui.widgetType]</code> (required, immutable) • <code>widgetId: [string]</code> (immutable) • <code>isVisible: [bool]</code> • <code>isEnabled: [bool]</code> • <code>minimumWidth: [int]</code> • <code>minimumHeight: [int]</code> • <code>sizePolicy: [ui.sizePolicy]</code> • <code>onConstructed: [callbackFunction]</code> • <code>onClose: [callbackFunction]</code> <p>Applicable to the root widget description in a dialog:</p> <ul style="list-style-type: none"> • <code>windowTitle: [string]</code> • <code>windowWidth: [int]</code> • <code>windowHeight: [int]</code> <p>Applicable to <code>widgetType.Layout</code>:</p> <ul style="list-style-type: none"> • <code>layout: [ui.layoutType]</code> (required, immutable) • <code>items: [array of widget]</code>

- descriptions] (immutable)
- contentsMargins: [object with format { left: [int], top: [int], right:[int], bottom:[int] }]
- spacing: [int]

Applicable items within
layoutType.HBoxLayout and
layoutType.VBoxLayout:

- stretchFactor: [int]
(immutable)
- alignment: [ui.alignment]
(immutable)

Applicable items within
layoutType.GridLayout:

- row: [int] (immutable)
- column: [int] (immutable)
- rowSpan: [int] (immutable)
- columnSpan: [int] (immutable)
- alignment: [ui.alignment]
(immutable)

Applicable to widgetType.Label:

- text: [string]
- wordWrap: [bool]

Applicable to widgetType.PushButton:

- text: [string]
- onClicked: [callbackFunction]

Applicable to widgetType.LineEdit:

- text: [string]
- isReadOnly: [bool]
- echoMode: [ui.echoMode]
onTextEdited:
[callbackFunction]
- onEditingFinished:
[callbackFunction]

Applicable to widgetType.TextEdit:

- text: [string]
- html: [string]

- `isReadOnly: [bool]`

Applicable to `widgetType.ComboBox`:

- `items: [array of objects with format { text: [string], userData: [variant]}] (write-only)`
- `currentIndex: [int]`
- `currentText: [string] (read-only)`
- `currentUserData: [variant] (read-only)`
- `onCurrentIndexChanged: [callbackFunction]`
- `isEditable: [bool]`
- `editText: [string] (write-only)`
- `onEditingFinished: [callbackFunction]`

Applicable to `widgetType.CheckBox`:

- `text: [string]`
- `isChecked: [bool]`
- `onToggled: [callbackFunction]`

Applicable to `widgetType.Slider`:

- `orientation: [ui.orientation]`
- `value: [int]`
- `range: [object with format { minimum: [int], maximum: [int] }]`
- `onValueChanged: [callbackFunction]`

Applicable to `widgetType.SpinBox`:

- `value: [int]`
- `range: [object with format { minimum: [int], maximum: [int] }]`
- `onValueChanged: [callbackFunction]`

Note that all attributes are optional except those marked as required. Also note that all attributes not marked as immutable can be set dynamically, with the corresponding 'setter' (e.g. `call widget.setVisible()`) to

	<p>modify the <code>widget.isVisible()</code> attribute).</p> <p>For all callbacks (attributes starting with the <code>onXYZ</code> pattern), functions are called with the <code>this</code> value set to the corresponding widget. For example:</p> <pre>{ widgetType: ui.widgetType.PushButton, text: "Show Alert", onClicked: function() { alert("Button text is " + this.text()); } }</pre> <p>This would alert the user by announcing what text just been entered, when text has just been entered.</p> <p>Widgets can also define timers. This is particularly useful for modeless dialogs. Timers can be used via the following functions:</p> <ul style="list-style-type: none"> • <code>startTimer(intervalInMs [int])</code>, returns a <code>timerId [int]</code> • <code>stopTimer(timerId [int])</code> • <code>onTimerEvent(timerId)</code>: [callbackFunction] <p>See the <i>Engine Designer</i> example for a demonstration of how this can be used.</p>
<code>ui.widgetType</code>	<p>An enum corresponding to different widget types available within a UI widget description. Possible values are:</p> <ul style="list-style-type: none"> • <code>Spacer</code>: A blank widget. • <code>Layout</code>: A container widget. Must specify a corresponding layout type. • <code>Label</code>: A text label. • <code>PushButton</code>: A clickable button. • <code>LineEdit</code>: A single-line text entry widget. • <code>TextEdit</code>: A multi-line text entry widget. Supports plain text and

	<p>HTML.</p> <ul style="list-style-type: none"> • ComboBox: A drop down. • CheckBox: A boolean check box with a text label.
<code>ui.layoutType</code>	<p>An enum corresponding to different layout types available within a UI widget description of <code>widgetType.Layout</code>. Possible values are:</p> <ul style="list-style-type: none"> • HBoxLayout: A horizontal layout. Items are spaced evenly, unless a <code>stretchFactor</code> is specified. • VBoxLayout: A vertical layout. Items are spaced evenly, unless a <code>stretchFactor</code> is specified. • GridLayout: A grid layout. Items can specify a row, column, <code>rowSpan</code> and <code>columnSpan</code>.
<code>ui.alignment</code>	<p>An enum corresponding to the alignment of an item within a layout widget. Possible values are:</p> <ul style="list-style-type: none"> • AlignLeft • AlignRight • AlignHCenter • AlignJustify • AlignAbsolute • AlignTop • AlignBottom • AlignVCenter • AlignBaseline • AlignCenter
<code>ui.sizePolicy</code>	<p>An enum corresponding to the size policy of a widget. This determines how much space it will try to consume within its layout, and how it will react when more or less space is made available. Possible values are:</p> <ul style="list-style-type: none"> • Fixed • Minimum • Maximum • Preferred • MinimumExpanding • Expanding • Ignored
<code>ui.orientation</code>	<p>An enum representing orientation. Possible values are:</p> <ul style="list-style-type: none"> • Horizontal • Vertical

<code>ui.echoMode</code>	An enum corresponding to the echo mode of a LineEdit widget. This can be used to display asterisks instead of characters when entering a password. Possible values are: <ul style="list-style-type: none"> • Normal • NoEcho • Password • PasswordEchoOnEdit
<code>project.save()</code>	Saves the project.
<code>project.build()</code>	Builds the project.
<code>project.exportGUIDs()</code>	Exports project guids.txt.
<code>project.lookup(guid)</code>	Returns the ManagedObject with the specified guid. See the <i>Project objects</i> section for further information about ManagedObjects.
<code>project.create(entityName)</code>	Returns a new instance of a ManagedObject of the given entity name. This will create any required child objects for the object (e.g. creating an Event will create and attach its EventMixer). See the <i>Project objects</i> section for further information about ManagedObjects.
<code>project.workspace</code>	The root Workspace ManagedObject of the project. See the <i>Project objects</i> section for further information about ManagedObjects.
<code>project.filePath</code>	The absolute path to the project's .fspro file on disk.
<code>project.importAudioFile(filePath)</code>	Imports an audio asset into the project and returns an AudioFile ManagedObject, or null if importing failed. The file is always imported into the root audio bin folder. Use the AudioFile.container relationship to reassign it to another folder.
<code>project.audioFileImported.connect(callback)</code>	Sets a function to be called when an audio file is imported, with the signature <code>function(audioFile)</code> .
<code>system.print(msg)</code>	Logs a message.

<code>system.warn(msg)</code>	Logs a warning.
<code>system.error(msg)</code>	Logs an error.
<code>system.message(msg)</code>	Displays a message in a modal popup dialog (blocking).
<code>system.question(msg)</code>	Displays a message in a modal popup dialog that also prompts the user for a 'yes'/'no' response, and returns a boolean value (blocking).
<code>system.getText(msg[, defaultText])</code>	Displays a message in a modal popup dialog that also prompts the user to enter a text string. Returns the string, or null if the user cancels (blocking).
<code>system.getNumber(msg[, defaultText])</code>	Displays a message in a modal popup dialog that also prompts the user for numeric input. Returns the value, or null if the user cancels (blocking).
<code>system.start(path_to_executable, process_options)</code>	<p>Runs an external process with options object { workingDir, args, timeout (milliseconds) } (blocking).</p> <p>Returns a result object { exitCode, standardOutput, standardError }.</p>
<code>system.startAsync(path_to_executable, process_options)</code>	<p>Runs an external asynchronous process with options object { workingDir, args }.</p> <p>Returns a ScriptProcess object representing the asynchronous process</p>
<code>system.getFile(filePath)</code>	Returns a studio.system.File object representing the file at filePath.
<code>system.openMode</code>	<p>An enum used with studio.system.File.open() to describe the mode in which a file is opened. Possible values are:</p> <ul style="list-style-type: none"> • <code>ReadOnly</code>: The file is open for reading. • <code>WriteOnly</code>: The file is open for writing. Note that this mode implies Truncate. • <code>ReadWrite</code>: The file is open for

	<p>reading and writing.</p> <ul style="list-style-type: none"> Append: The file is opened in append mode so that all data is written to the end of the file. Truncate: Is possible, the file is truncated before it is opened. All earlier contents of the file are lost.
<code>window.open(windowType)</code>	Opens a new window of the given window type (e.g. "Event Editor").
<code>window.navigateTo(managedObject)</code>	Attempts to navigate to a ManagedObject in the UI. Will open a new window if required. Returns true if successful. See the <i>Project objects</i> section for further information about ManagedObjects.
<code>window.browserCurrent()</code>	Returns the currently selected ManagedObject in the last active browser. See the <i>Project objects</i> section for further information about ManagedObjects.
<code>window.browserSelection()</code>	Returns an array of selected ManagedObjects in the last active browser (e.g.: the Banks browser). See the <i>Project objects</i> section for further information about ManagedObjects.
<code>window.editorCurrent()</code>	Returns the currently selected ManagedObject in the last active editor (e.g.: the Event Editor). See the <i>Project objects</i> section for further information about ManagedObjects.
<code>window.editorSelection()</code>	Returns an array of selected ManagedObjects in the last active editor (e.g. in the multitrack view). See the <i>Project objects</i> section for further information about ManagedObjects.
<code>sourceControl.registerProvider(name, provider)</code>	Registers a source control provided. (Experimental)
<code>os.platform</code>	The current operating system of the computer. Returns either "mac" for Mac OSX or "win" for all Windows versions.

The Javascript API for the `studio.system.File` type is as follows.

API	Description
<code>File.exists()</code>	Returns <code>true</code> if the file the object represents exists on disk and <code>false</code> otherwise.
<code>File.open(openModeFlag)</code>	Opens the file in the mode specified by <code>openModeFlag</code> . The available options are described by the <code>system.openMode</code> enum. Returns <code>true</code> if the operation succeeds and <code>false</code> otherwise.
<code>File.writeText(text)</code>	Writes the content of text to the file. Returns the number of bytes that were actually written, or <code>-1</code> if an error occurred.
<code>File.readText(maxSize)</code>	Reads at most <code>maxSize</code> bytes from the file.
<code>File.close()</code>	Closes the file and flushes any changes to disk.

Project objects

Project level data in Studio is represented as a graph of *ManagedObjects*. The scripting interface provides access to project data via the `studio.project.workspace` object. This returns a `studio.project.ManagedObject` type object that interfaces with the underlying project data.

A `ManagedObject` is composed of:

- **Id:** Unique identifier (GUID).
- **Properties:** Each property is a simple data field (e.g. float, bool, string) with a *value*.
- **Relationships:** Links to other `ManagedObject`s, known as *destinations*. These can be *ToOne*, *ToMany*, and may or may not be ordered.

Note

Only high level project validation is currently available at this time. It is therefore possible for a script to create types of invalid content that FMOD Studio cannot easily detect and fix. To prevent this, look at objects created without using scripting for examples of safe property values and object relationships. Feel free to contact us at support@fmod.com if you would like us to evaluate how safe your scripts are.

The Javascript API for the `studio.project.ManagedObject` type is as follows.

API	Description
<code>ManagedObject.entity</code>	A string representing the object's C++ type (e.g. "Event"). Immutable.
<code>ManagedObject.id</code>	The object's unique ID. Immutable.
<code>ManagedObject.isValid</code>	Returns whether an object is in a valid state. Immutable.

<code>ManagedObject.properties</code>	Extended API. Returns an object of type <code>studio.project.ManagedPropertyMap</code> . Provides access to an object's properties. Immutable.
<code>ManagedObject.relationships</code>	Extended API. Returns an object of type <code>studio.project.ManagedRelationshipMap</code> . Provides access to an object's relationships. Immutable.
<code>ManagedObject.isOfType(entityName)</code>	Returns <i>true</i> if the object has an entity that matches <code>entityName</code> , or is of a derived type.
<code>ManagedObject.isOfExactType(entityName)</code>	Returns <i>true</i> if the object has an entity that exactly matches <code>entityName</code> . This does not include derived types.
<code>ManagedObject.nullify()</code>	Detaches the object (and any of its child objects) from the graph. Equivalent to 'Delete' in the UI.
<code>ManagedObject.%property_name%</code>	Convenience API. Provides access to a managed object property's value. Setting the property (with the <code>= operator</code>) will set the property value.
<code>ManagedObject.%relationship_name%</code>	Convenience API. Provides access to a managed object relationship's destinations. Getting a <i>ToOne</i> relationship will return the single destination object (or null if it is unassigned). Getting a <i>ToMany</i> relationship returns an array of destinations (or an empty array if it has no destinations). Setting a <i>ToOne</i> relationship (with the <code>= operator</code>) will set replace the current destinations. <i>ToMany</i> relationships cannot be modified with the <code>= operator</code> , and must be edited via the Extended API.
<code>ManagedObject.dump()</code>	Logs all members.

As an example of using a `ManagedObject`, calling `studio.project.workspace.mixer.masterBus.volume = 2` would walk through the Workspace's relationship to the global Mixer, then the Mixer's relationship to the MasterBus, and finally set the MasterBus' volume property to 2dB.

Note that this example uses the Convenience API to walk the graph in a more readable fashion. Using the Extended API, the equivalent would be `studio.project.workspace.relationships.mixer.destinations[0].relationships.masterBus.destinations[0].properties.volume.setValue(2)`. As you can see, the Convenience API is suitable for quick traversal of the graph, while the Extended API provides access to advanced attributes of properties and relationships.

The Javascript API for Extended API objects (within the `studio.project` namespace) is as follows.

<i>API</i>	<i>Description</i>
<code>ManagedPropertyMap.parent</code>	The owning managed object of the property map. Immutable.
<code>ManagedPropertyMap.%property_name%</code>	Returns an object of type <code>studio.project.ManagedProperty</code> for the given property name. Immutable.
<code>ManagedPropertyMap.dump()</code>	Logs all members.
<code>ManagedRelationshipMap.parent</code>	The owning managed object of the relationship map. Immutable.
<code>ManagedRelationshipMap.%relationship_name%</code>	Returns an object of type <code>studio.project.ManagedRelationship</code> for the given relationship name. Immutable.
<code>ManagedRelationshipMap.dump()</code>	Logs all members.
<code>ManagedProperty.parent</code>	The owning managed object of the property. Immutable.
<code>ManagedProperty.name</code>	Name of the property. Immutable.
<code>ManagedProperty.dataType</code>	The type of data stored by this property's value. Immutable.
<code>ManagedProperty.value</code>	The value stored by the property. Immutable.
<code>ManagedProperty.defaultValue</code>	Default value the property takes when created. Immutable.
<code>ManagedProperty.setValue(value)</code>	Sets the value stored by the property.
<code>ManagedProperty.dump()</code>	Logs all members.
<code>ManagedRelationship.parent</code>	The owning managed object of the relationship. Immutable.
<code>ManagedRelationship.name</code>	Name of the relationship. Immutable.

ManagedRelationship.cardinality	A string representing the object's cardinality. Either 'ToOne' or 'ToMany'. Immutable.
ManagedRelationship.isOrdered	A boolean that is true if ordering is stored for this relationship, or false if it is not. The significance of ordering depends on the kind of relationship.
ManagedRelationship.destinations	An array of managed object destinations. Immutable.
ManagedRelationship.add(managedObject)	Appends a new destination to a relationship. Replaces the current destination for a <i>ToOne</i> relationship.
ManagedRelationship.insert(index, managedObject)	Inserts a new destination to a relationship at a given index. The index must be in range [0, destinations.length].
ManagedRelationship.remove(managedObject)	Removes a destination from a relationship.
ManagedRelationship.dump()	Logs all members.

Scripting Examples

Creating and deleting an event

Using the managed object API, along with the `studio.project.create(entityName)` function, you can create objects within a project. For example, to create a new event in the root folder of the currently open project:

```
myEvent = studio.project.create("Event");
myEvent.name = "My New Event";
myEvent.folder = studio.project.workspace.masterEventFolder;
```

And to delete the newly created event:

```
myEvent=nullify();
```

Checking the tool version

Studio provides the `studio.version` script object which allows you to check the tool version. It is good practice to check the tool version before installing script functionality. For example:

```
if(studio.version.majorVersion >= 7) {
```

```
// add menu bar items
}
```

Modifying the global context

When Studio loads script within your Javascript files, it executes within its own scope. This means that if you declare a function called `foo()`, you will still not be able to call `foo()` directly from the shell interface. To register functions that can be called globally, you must attach them to the global context. For example:

```
var global = this;
global.foo = function() { alert("Hello!"); }
```

Custom Menus

```
studio.menu.addMenuItem({
    name: "Greetings\\Say Hello",
    execute: function() { alert("Hello"); },
    keySequence: "Ctrl+H",
});

studio.menu.addMenuItem({
    name: "Greetings\\Say Bye",
    execute: function() { alert("Bye"); },
    isEnabled: false,
});

studio.menu.addMenuItem({ name: "Greetings\\----" }); // use all '-' characters to create a separator

studio.menu.addMenuItem({
    name: "Greetings\\Say Checked",
    execute: function() { alert("Checked toggled"); this.isChecked = !this.isChecked; },
    isChecked: true,
});

studio.menu.addMenuItem({
    name: "Greetings\\Advanced\\Say what's Selected",
    execute: function() { alert(studio.window.browserCurrent()); },
    isVisible: function() { return studio.window.browserCurrent(); }
});

studio.menu.addMenuItem({
    name: "Greetings\\Dynamic Submenu",
    subMenuItems: function() {
        var items = [];
        for(var i=0; i < 4; i++) {
            items.push({
                name: "#" + i,
                execute: function() { alert("Well hello there " + this.name); },
            });
        }
        return items;
    },
});
```

Custom Save & Build menu item

```
studio.menu.addMenuItem({ name: "Save and Build", execute: function buildAndCopy() {
    studio.project.save();
```

```
        studio.project.build();
        alert("Save and Build complete!");
    });
}
```

Setting master volume

```
var masterBus = studio.project.workspace.mixer.masterBus;
masterBus.volume = -2;
```

Selecting an event and setting timeline position

```
var eventId = "{aabe5118-c144-4dc3-839a-ff52a2b49162}";
var timelinePos = 2.3;

var event = studio.project.lookup(eventId);
if(event) {
    studio.project.open(event);
    event.timeline.cursorPosition = timelinePos;
    alert("Opened and scrubbed: " + event.name);
}
else {
    alert("Could not find event: " + eventId);
}
```

Single Sound Modules vs. Multi Sound Modules

Introduction

The Single Sound Module and Multi Sound Module objects are probably the two most commonly used Module objects within FMOD Studio. Between them they provide a range of Properties that suit many applications and they are also efficient in their use of resources. This example demonstrates why using a Multi Sound Module can result in better sounding audio for your project.

The Basics

Essential both the Single Sound Module and Multi Sound Module objects are an easy method of referencing sound files for playback within FMOD Studio. Both types of Module objects include Volume and Pitch controls the main difference is that a Multi Sound Module can contain more than one sound file. Different options are available when using multiple sound files within a module object, but the simplest and most direct is the reduction or removal of repetition of audio within a game project.

The Example



Event with Audio Track 2 Muted

Look at the example event within FMOD Studio called Single vs. Multi by clicking on the event name to make it active. The event contains two Audio Tracks to allow for a comparison between the two types of Module. Audio Track 1 contains a Single sound module object and Audio Track 2 contains a Multi Sound Module object. To start with Mute Track 2 so that only Track 1 is heard for Playback

If you press the Play control under the default setup, Studio triggers the Single sound module in Audio Track 1. This is the sound of a double barrel shotgun being fired. There is a single sound file with no alteration being applied to the sound in any way. Essentially FMOD Studio simply plays this sound over and over each time it is triggered. Play the event now a few times and listen to the sound of the shotgun.

As you would expect you hear the exact same sound each time you press play. This is how a Single sound module object works. Now, switch the active state of the two Audio Tracks by un-muting Audio Track 2 and Muting Audio Track 1. This makes Audio Track 2 live.



The Multi Sound Module object contains 3 sound files

Audio Track 2 contains a Multi Sound Module object that references three sound files. All three sound files are also of a Shotgun being discharged, but each sound is very slightly different. When the event is triggered with the Multi Sound Module object active there is a subtle difference each time it plays,

but even a subtle difference is enough for the human ear to register that it is not hearing the exact same sound. Obviously the more sound files that are included in a Multi Sound Module the greater the variation at playback.

Taking Things Further

Variation of a sounds playback is a powerful way to avoid repetitive sounds. Varying Pitch and Volume each time a sound is triggered can further add to the effect of having more than a single sound file. Certain Properties within FMOD Studio allow for variable Pitch and Volume to be applied to a sound on Playback. The range of values applied to the sound can subtly or drastically alter the sound at playback; experimentation with Property values is the best way to establish how far a sound can be manipulated while maintaining an appropriate effect.

Note

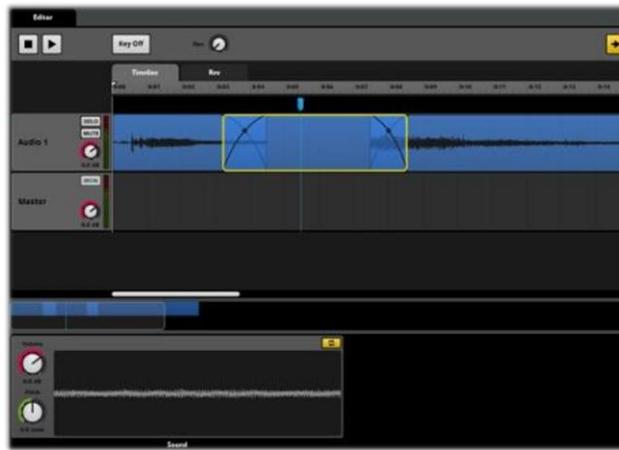
Do not underestimate the importance of subtle changes. When a sound is created to represent a particular object or action it needs to be relatively consistent with the event it represents. Subtle differences in playback avoid the feeling of repetition but more extreme changes in playback can make the audio sound like it belongs to a different object or event. In the case of the shotgun, slight alterations of Pitch can reinforce the subtle differences and add even more variation, but if the pitch alterations are too extreme either up or down in frequency then the sound of the shotgun may end up sounding like an explosion or a smaller handgun.

A Variable Duration Cycle

A common requirement in game projects is to create a sound for an event that has a definable start and finish point but that can have variable duration. This is very common for machinery sounds and can often create tricky implementation issues. FMOD provides a simple solution to this problem, and with the advanced functionality within FMOD Studio the process can be developed into complex audio events.

The Event

Examine the event named Variable Duration Cycle. It includes a single audio track and three Single sound module objects. The Modules have been positioned to create a smooth cross fade between them and this could easily be recreated in any audio editing software to create the sound of a car starting up, idling briefly and then being shut down. This event however includes a Sustain Point aligned with the middle Module that represents the idling sound and this allows the sound to be played for as long as required.



A Timeline with Sustain Point

The Mechanics

With regular playback an event created on a Timeline proceeds through time from zero until the last sound triggered has finished. FMOD Sustain Points instruct FMOD to pause in the advancement of the cursor along the Timeline. This means FMOD effectively continues to play that point in time until it is instructed to advance again.

In this example the Sustain Point coincides with the idle sound of the car. The Single sound module object of the idle sound has been set to Loop (note the small yellow square in the Property display in the Deck). This means that while the cursor is touching the Module it continues to play endlessly. Once Keyoff is triggered in the event, the cursor continues to move along the Timeline and ultimately trigger the engine shutdown sound and the event stops playing.

Going Further

FMOD Studio can add extra layers of complexity to a cycle event. The dial in the Play Region to the right of the Key off button adds some extra functionality to this event. Play the event again, and when the sound reaches the loop point, turn the dial and listen to what happens.



Parameter controller dial

The dial is a controller for a Parameter that has been added to this event. The Parameter has been called Rev, but it could be called anything appropriate. By clicking on the Parameter tab called Rev you can

change to the Parameter view and observe what is occurring. Explore this event to understand some of its possibilities.

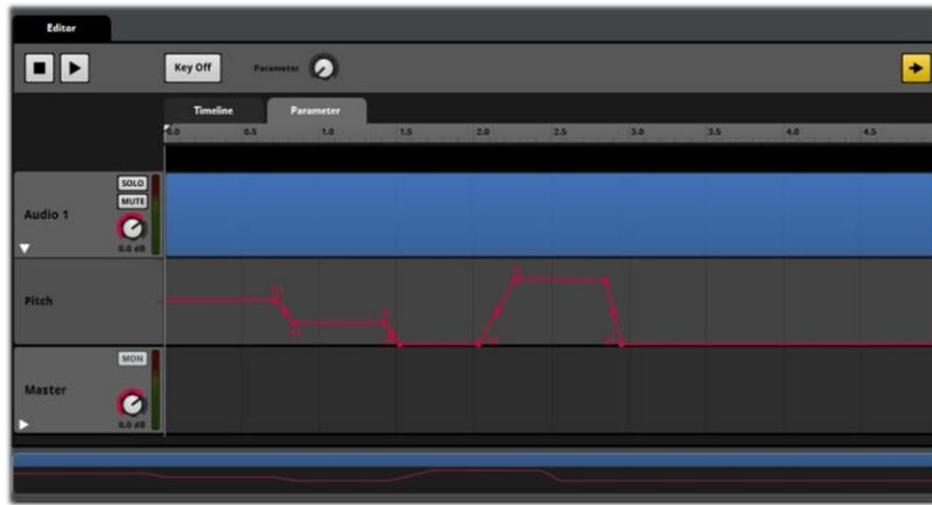
Pitch Shifting Magic

There is a previous mention in this document of the usefulness of using a single sound file for a variety of purposes. The fact the FMOD Studio only references a sound files and never alters them makes this function extremely powerful. The event named Pitch Shifting Magic has been created to more fully demonstrate this potential.

The Event

Pitch Shifting Magic consists of a Single sound module that contains a recording of a metal box about the size of a shoe box being kicked. The sound itself is somewhat plain and uninspiring and for this reason is perfect for this demonstration. Firstly you may notice that the event looks empty on first inspection. This is because in this event the Timeline does not include any Module objects. If however you click on the Parameter tab FMOD displays a different view that contains the Module for this example.

The Game Parameter for this event has a range of values and an Automation curve controlling Pitch has been added to the event. At various values of the Game Parameter different Pitches have been defined to affect the playback sound of the event. The initial value is set to the default pitch of the sound file being used for this event.



Automation of Pitch Property provides a variety of output results

When you press the Play control the event is triggered once and you can hear the sound of the metal box. As has been mentioned it is a fairly ordinary sound. It might perhaps be appropriate to represent a metal mailbox being struck or shut forcefully. This single sound file however can be altered in its playback behavior to provide a wider range of uses.

Click on the Parameter ruler at the value of 1.5 and the cursor repositions to that value. The Pitch of the sound has now been altered approximately one octave below the original sound file. Pressing Play now triggers the sound, but it now sounds like a much larger metal object, similar to a 40 gallon drum being impacted. History would suggest that games have a far greater use for the sound of a 40 gallon drum than that of a mailbox.



Clicking on the Parameter ruler again at a value of 2.5 lowers the pitch still further. This time the event sounds more like a large metal garage or hanger door being opened. The analogue nature of the pitch shifting means that as a sound is pitched down it becomes longer than the original file length, and as it is pitched up it becomes shorter.

Moving the Parameter to 3.5 now results in the event pitching the sound file up above the original. The sound is now higher in pitch and shorter in length; it sounds more like a tin can being kicked. A final selection of 4.5 on the Parameter ruler pitches the event back down to an extremely low level. The sound now takes on a far more ominous effect as it could represent the hull of a ship or spacecraft being breeched.

Going Further

All the previous effects were created using a single sound file and some pitch alteration, but remember the sound file can also be used as part of a more complex event. The low pitch effect could be combined with other sounds to simulate an impact crash of a vehicle (with the previously mentioned shotgun sound for instance), the high pitch clank could represent debris of such an event or perhaps the small component of a larger mechanical process.

This example was created simply to demonstrate how useful a single sound file can be and how powerful even simple process Properties within FMOD Studio are. Thinking outside the box of traditional sound design can produce effective results.

Appendix A

Keyboard Shortcuts

- B Hide / Reveal The Event Browser
- D Hide / Reveal the Deck Area
- P Hide / Reveal the Properties area
- CTRL + 1 Event Editor
- CTRL + 2 Mixer
- CTRL + 3 Audio Bin
- CTRL + 4 Event Editor
- CTRL + 5 Mixer Router
- Delete key (currently supported in the multi-track, hopefully everywhere in the release)
- Ctrl +N (Command+N) New
- Ctrl +O (Command+O) Open
- Ctrl +S (Command+S) Save
- Ctrl +Alt +W (Command+Alt +W) Close
- Ctrl X (Command+X) Cut
- Ctrl C (Command+C) Copy
- Ctrl V (Command+V) Paste
- Ctrl+Z (Command+Z) undo
- Ctrl+Y (Command+Shift+Z) redo
- Ctrl+` (Command+`) cycle through open windows
- Double-click on labels to edit (works for values, track names, Parameter tabs, mixer view tabs)
- Ctrl+Drag (Command+Drag) on dials/faders for fine control
- Ctrl Shift +L (Command+Shift+L) Loop Playback
- Alt+Drag on dials to set randomization
- Alt+Wheel to zoom in/out in multi-track
- Ctrl +] (Command+]) zoom in
- Ctrl +[(Command+[) zoom out
- Ctrl +M (Command+M) minimize
- Ctrl +W (Command+W) close window
- Shift+Wheel to scroll left/right in multi-track
- Alt+Shift+B send module to back
- Alt+Shift+F bring module to front
- Alt +F4 exit the program
- Double-click on curve control points (diamonds) to change curve shape
- Shift+Drag while bulk editing in the mixer to lock all strips to the same value
- F1 Display the manual
- Start+Ctrl+[(or Ctrl+Command+[on Mac) is the shortcut for zoom-to-fit.

Support

Website

www.fmod.org

Technical support

support@fmod.org

Sales / licensing enquiries

sales@fmod.org

Training and tutorials

http://www.youtube.com/user/FMODTV?ob=0&feature=results_main

Credits

Manual written and produced by:

Sound Librarian

www.soundlibrarian.com

and

Joseph Harvey, Thuan Seah Tan, and Patrick Wadsworth