

Koreographer PlayMaker Integration Overview

for v1.4.0



Table of Contents

[Table of Contents](#)

[Overview](#)

[PlayMaker Version Support](#)

[Built-in Help](#)

[Included PlayMaker Actions](#)

[Koreography Event Actions](#)

[Koreography Event Trigger Action](#)

[Payload Accessor Actions](#)

[Koreographer Music Time API Actions](#)

[Music Player Control Actions](#)

[Simple Music Player Actions](#)

[Multi Music Player Actions](#)

[The Trigger Proxy Component](#)

[Configurable Fields](#)

[Using the Koreography Event Trigger Proxy](#)

[Integration Support](#)

Overview

The PlayMaker Integration for Koreographer enables you to add Koreographer control and functionality to Finite State Machines (FSMs) built with Hutong Games' [PlayMaker](#). The integration provides PlayMaker [Actions](#) for handling Koreography Events, Koreographer Music Time APIs, and the two main Music Players included with Koreographer. Also included is a special proxy component that simplifies the handling of Koreography Events in scenes that involve lots of FSMs. Please see the following sections for more information!

PlayMaker Version Support

The PlayMaker Integration was built against PlayMaker 1.7.4. It has also been verified to work with 1.8.0. Previous versions have not been tested but we will try our best to support older versions. If you run into an issue with the integration, please [reach out to us for support](#)!

Built-in Help

All PlayMaker actions included in the integration provide a helpful description in the [Actions Browser](#) when selected. Furthermore, every configurable field of every action is equipped with a descriptive Tooltip. If you encounter a situation in which the information in this documentation conflicts with the information in action descriptions or tooltips, please [let us know](#)!

Included PlayMaker Actions

Three classes of PlayMaker Actions are included in the integration: Koreography Event actions, Koreographer Music Time API actions, and Music Player Control actions. Please see the following sections for more information on each.

Koreography Event Actions

At its core, the Koreographer runtime system is an event system. When a Koreography Event is encountered in Koreographed audio, any callbacks registered¹ to listen for that event will be notified and sent a reference to that Koreography Event. The Koreography Event (and its Payload) is only available during the callback phase. The Koreography Event Trigger action makes these notifications available to PlayMaker FSMs by triggering FSM events. During this callback phase, the Payload Accessor actions may be used to access Payload content associated with the callback's Koreography Event.

Note: Koreography Tracks, Koreography Events, and Payloads are configured in the Koreography Editor. Please see the *Koreographer User's Guide* for more information.

The Koreography Event Trigger action and Payload Accessor actions are described in more detail in the following sections.

Koreography Event Trigger Action

When entered, the Koreography Event Trigger action registers the current state for Koreography Events associated with a specific Event ID. When exited, it unregisters the state for callbacks. When a callback occurs, the Koreography Event Trigger will trigger the specified FSM Event. Depending on the setup of the state, this may send an Event to other FSMs (by adding a **Set Event Target** action before the Koreography

¹ It is important to keep in mind that the registration is based on a Koreography Track's **Event ID**.

Event Trigger Action in a state's action list) or its own, causing a state transition. It is during such transitions that the Payload Accessor actions should be used.

Thus, there are several ways for your FSM to receive a Koreography Event callback:

- Via the Koreography Event Trigger action while a state with the action is active.
- From an FSM Event that was triggered by a Koreography Event Trigger in a *different* FSM.
- From a KoreographyEventTriggerProxy component (see [below](#) for more).

Payload Accessor Actions

The Payload Accessor actions are used to access the Payload of the current Koreography Event during a callback phase. If these actions are called outside of the callback phase then the specified variable will not be changed, a warning will be logged to the PlayMaker log, and processing will continue.

- **Get Koreography Event Payload Color:** Retrieves a Color value from stored Koreography Event info. This action will evaluate a Gradient Payload to get the Color at the current event time.
- **Get Koreography Event Payload Curve:** Retrieves an AnimationCurve from stored Koreography Event info.
- **Get Koreography Event Payload Float:** Retrieves a Float value from stored Koreography Event info. This action will evaluate a Curve Payload to get the float at the current event time.
- **Get Koreography Event Payload Int:** Retrieves an Int value from stored Koreography Event info.
- **Get Koreography Event Payload Text:** Retrieves a String value from stored Koreography Event info.

Koreographer Music Time API Actions

When music is played back through a properly configured component (see the “Custom Players” section of the *Koreographer User's Guide* for more), you can access Koreographer's Music Time APIs. The SimpleMusicPlayer and MultiMusicPlayer are two such components.

The Music Time API actions currently available are:

- **Get Music Beat Time:** Gets the current music time in beats as a float. This value is retrieved from the singleton Koreographer component instance.
- **Get Music Beat Time Delta:** Gets the delta in beat time of the current Koreography processing pass as a float. This value is retrieved from the singleton Koreographer component instance.
- **Get Music Beat Time Normalized:** Gets the percentage of the way through the current beat as a float in the range of [0, 1). This value is retrieved from the singleton Koreographer component instance.

Note that each of the above actions may be configured to run **Every Frame**.

Music Player Control Actions

Koreographer comes with two main music player components: the SimpleMusicPlayer and the MultiMusicPlayer (please see the *Koreographer User's Guide* for a description of each). This integration provides actions that enable the control of these music players.

Simple Music Player Actions

The following actions are included to control a SimpleMusicPlayer component:

- **Simple Music Player Is Playing:** Sends an event based on the playing state of a SimpleMusicPlayer. (For the purpose of this test, “Paused” is not playing.) If a Koreography is specified, the check will see if that specific audio is playing.
- **Simple Music Player Load Song:** Loads Koreography into a SimpleMusicPlayer for playback. Optionally begins playback of the audio specified by the new Koreography. This will stop and unload any previously playing Koreography.
- **Simple Music Player Pause:** Pauses a SimpleMusicPlayer.
- **Simple Music Player Play:** Causes a SimpleMusicPlayer to play. If a Koreography is specified, it will first load that Koreography. You may want to consider unchecking the “Auto Play On Awake” checkbox on the SimpleMusicPlayer component if you plan to control initial playback later.
- **Simple Music Player Seek To Sample:** Seeks the playhead of a SimpleMusicPlayer, causing any playing or paused audio to jump.
- **Simple Music Player Stop:** Stops a SimpleMusicPlayer.

Multi Music Player Actions

The following actions are included to control a MultiMusicPlayer component:

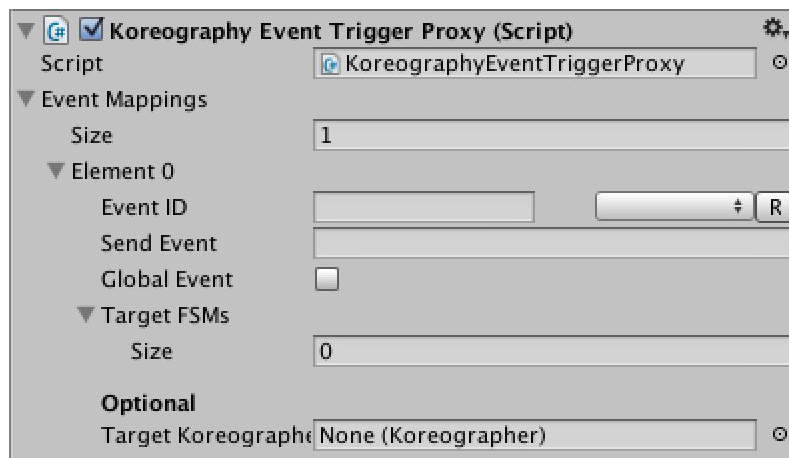
- **Multi Music Player Is Playing:** Sends an event based on the playing state of a MultiMusicPlayer. (For purposes of this test, “Paused” is not playing.) If a Koreography is specified, the check will see if the AudioClip it references is playing.
- **Multi Music Player Pause:** Pauses a MultiMusicPlayer.
- **Multi Music Player Play:** Causes a MultiMusicPlayer to play. You may want to consider unchecking the “Auto Play On Awake” checkbox on the MultiMusicPlayer component if you plan to control initial playback later.
- **Multi Music Player Seek To Sample:** Seeks the playhead of a MultiMusicPlayer, causing any playing or paused audio to jump.
- **Multi Music Player Stop:** Stops a MultiMusicPlayer.

Unlike the actions for the Simple Music Player, there is no “Load Song” equivalent in the current integration. We hope to overcome this shortcoming in a future release. Please [let us know](#) if this is impacting your workflow.

The Trigger Proxy Component

The Koreography Event Trigger Proxy component enables the configuration of Koreography Event to FSM Event mappings directly in the Hierarchy. This is a convenient workflow that removes the Koreography-registration responsibility from FSM states, reducing the number of Koreography Event Trigger actions that you have to add and manage. In this manner, the Koreography Event Trigger Proxy component is intended to be used *in lieu of* the [Koreography Event Trigger](#) action.

The Koreography Event Trigger Proxy can be found in the *Add Component* menu under *Koreographer→PlayMaker→Koreography Event Trigger Proxy*.



The Koreography Event Trigger Proxy component Inspector

Configurable Fields

When initially added, the Koreography Event Trigger Proxy contains only a single configurable field:

- **Event Mappings:** A list of “mappings” that allow you to configure the connection between a Koreography Event ID and an FSM Event, and to which FSM(s) you wish those events to be routed.

You can set the initial number of mappings you wish to create by adjusting the Size field. Each mapping has the following configurable fields:

- **Event ID:** The Koreography Event ID.
- **Send Event:** The FSM Event to send when a Koreography Event from a Koreography Track with the given Event ID is triggered.
- **Global Event:** If checked, the Target FSMs list is ignored and this event is sent to *all* running PlayMakerFSM components.
- **Target FSMs:** Specific PlayMakerFSM components to which to send the event. Only used if Global Event is unchecked.
- **Target Koreographer:** A specific Koreographer component with which to register for events. If this is null, it will use the singleton default.

Note: The Koreography Event Trigger Proxy’s **Global Event** is not related to PlayMaker’s Global Events. The checkbox merely indicates that the configured Send Event should be sent to all active FSMs in the scene.

Using the Koreography Event Trigger Proxy

Using the Koreography Event Trigger Proxy is extremely simple. The following steps assume that you have already configured your scene for Koreography playback (see the *Koreographer Quick Start Guide*) and have at least one PlayMaker FSM configured in your scene.

1. Add the **Koreography Event Trigger Proxy** to a GameObject in your scene. Recommended GameObjects include the one that contains a **Koreographer** component or any **PlayMakerFSM(s)** to which you wish to send events.

2. Add at least one Event Mapping to the Koreography Event Trigger Proxy and configure it with your desired settings.
3. Test the scene.

If everything is properly configured, your PlayMaker FSM should begin to receive Koreography Events!

Note: PlayMaker state transitions triggered by handling the events sent by **Koreography Event Trigger Proxy** components enable the use of the [Payload Accessor actions](#). If a Payload is configured on the Koreography Event that triggered the callback, its contents may be accessed at this time.

Integration Support

Need help getting your PlayMaker integration up and running? Have a use case that isn't covered by the integration? Need a special Action that isn't currently provided? **Please** reach [out to us for support](#)! We are here to help you get moving!