Timeline Playable Wizard

The Playable Wizard is a tool used to make creating Playables for use with Timeline easier. To open the Playable Wizard select “Timeline Playable Wizard…” from the Window menu. The wizard will create scripts with all boilerplate code already written. It is important to note that the Playable Wizard only creates scripts, to change how the resultant Playables work the scripts will need to be edited using your normal code editor.

The Timeline Playable Wizard works in two modes: Normal and Standard Blend Playable. This mode is selected by checking or unchecking the Standard Blend Playable checkbox. In Normal mode (with the box unchecked) the Playables created by the wizard are much more flexible but will need their mixers completed. To do this, once your Playable has been created, open the script that has been created with the MixerBehaviour suffix using your normal code editor and complete the code as appropriate for your particular Playable. Standard Blend Playables are less flexible and follow a specific format but all of the code will be written for you. They can still be edited after creation like all scripts.

Below are the settings for when the Timeline Playable Wizard is in Normal mode. Details of Standard Blend Playables and their creation follows afterwards.

1. Show Help
   This option pertains just to the wizard itself. Disabling it will hide all the help boxes for the wizard, enabling it will show them.
2. Playable Name
   A number of classes are required for Playables to work with Timeline. The Playable Name that you choose will be a prefix for those classes and so it must not have any special characters or spaces and must start with a capital letter. The suffixes created for the scripts are ‘Behaviour’, ‘Clip’, ‘MixerBehaviour’, ‘Track’ and optionally ‘Drawer’. Though it will not affect functionality it is best to not use these words in your Playable Name to avoid confusion.
3. Track Binding
   This is the output of the Playable. It is generally an object in the scene such as a Component or GameObject. If - for example - your Playable is designed to move a GameObject then the Track Binding is most likely to need to be a Transform.
4. Exposed References
   In order to reference objects in scenes Playables need to use Exposed References. These are useful for additional information from the scene that can affect how the Playable works. For example, to tween a ball GameObject between two Transform, the Transforms would need to be Exposed References.
5. Behaviour Variables
   These are all the variables that are required for your playable to perform its operations. For example, if your playable moves a Transform then a float variable called speed

might be appropriate. Note that any reference types (for example components) used here must not be in the scene, they may be references to assets such as prefabs.

6. Clip Caps
   These are the settings that affect the way that the Timeline Editor works with the Playable. For details on how Clip Caps work, see the Timeline documentation.
7. Track Color
   To make Playables and Tracks more easily identifiable they can be given a colored stripe. This color is set here. It is recommended that you choose color that is different to the other Playables that you will be using to make identifying what each Track does easy.
8. Create Drawer?
   Check this box in order to have a PropertyDrawer automatically created for the Playable. This has the effect of tidying how the inspector looks when the Playable is selected in the Timeline Editor. It will be shown as all Exposed References followed by all Behaviour Variables. It has the benefit of meaning that if you wish to adjust how the Playable is seen in the inspector you only need edit a script rather than write one from scratch. It is important to note that it is created for all the Behaviour Variables declared in the Wizard, if more variables are declared subsequently then they will need to be added to the PropertyDrawer. Also note that the PropertyDrawer targets the PlayableBehaviour script and so does not affect how the Exposed References are shown.

The Timeline Playable Wizard can also create a specific type of Playable called a Standard Blend Playable. These are a special case of the Playables which require no code to be written. The special case which they cover override the values of a component’s properties for the duration of a clip. This can be combined with blending between clips and setting curves for the values to animate a component in a piecemeal fashion without the use of Animators or AnimationClips. For example a Playable that changes the color and range of a Light component for its duration then returns to the values set on the Light in the scene. To make this type of Timeline Playable check the “Standard Blend Playable” box near the top of the Playable Wizard. Details of how to proceed are described below.

1. Playable Name
   A number of classes are required for Playables to work with Timeline. The Playable Name that you choose must work with those class names as it will be a basis for them. That means that the name must not have any special characters or spaces and must start with a capital letter.
2. Track Binding
   All Standard Blend Playables must have an output and this represents that output. In the above example of blending the variables of a Light, the Track Binding would be a Light component.
3. Default Values
   Normally all the properties of the Track Binding have no set values and default to

whatever their zero-value is. By creating a component matching the Track Binding and using it to set the default values all the Track Binding properties on the playable will have defaults equal to their equivalent property on the assigned component.

4. Standard Blend Playable Properties
   These are the variables of the Track Binding that will be affect. It is important that the Track Binding is selected before choosing which properties to affect because changing the Track Binding will clear this list. The list shows all public instance properties of the selected component that it makes sense to either blend or assign. For example, it is possible to blend floats and Vector3s because they can both be multiplied by a weight and added together. It is possible to assign bools and enums based on which Playable has the greatest weight at the time.
5. Track Color
   To make Playables and Tracks more easily identifiable they can be given a colored stripe. This color is set here. It is recommended that you choose color that is different to the other Playables that you will be using to make identifying what each Track does easy.

TransformTween

This type of playable should be used for very simple transform movements. It supports several different movement styles but all translation happens in a straight line. For more complex movement use an Animation playable. The TransformTween track binds to the Transform in the scene you wish to move.

Start Location	This is a reference to a Transform in the scene that marks the position and/or rotation of the moving Transform when the playable starts. If it is left null the position/rotation of the moving Transform when the playable starts will be used.
End Location	This is a reference to a Transform in the scene that marks the position and/or rotation of the moving Transform when the playable finishes.
Tween Position	Whether or not the position of the Transform should change.
Tween Rotation	Whether or not the rotation of the Transform should change.
Tween Type	In what way should the Transform move.
Linear	The Transform moves uniformly throughout the duration of the playable.
Deceleration	The Transform starts off moving quickly then slows down as

	it reaches the End Location.
Harmonic	The Transform starts off moving slowly, speeds up as it reaches the middle of the tween then slows down as it reaches the End Location.
Custom	When this option is selected, the playable tries to calculate the distance the Transform should move each frame based on normalised speeds set for the start and end of the tween. This works as an approximation - for fine tuned control use and Animation playable.
Custom Starting Speed	The normalised speed of the Transform as it starts the tween.
Custom Ending Speed	The normalised speed of the Transform as it ends the tween.

TextSwitcher

This type of playable should be used to display different messages to the screen using a UI Text component. It is ideal for things like subtitles. The TextSwitcher track binds to a Text component in the scene. This will be the component that the text is displayed through. This playable can be used with blending, meaning that you can blend between different clips and a weighted average will be used to determine the font size and color of the text. The message with the most weight will be used.

Message	This is the string that will be displayed by the generated by Text component.
Font Size	The size of the text to be displayed.
Color	The color of the text to be displayed.

TimeDilation

This type of playable will adjust the Time.timeScale for the duration of the clip. The most common use would be to create bullet-time effects. By default each clip will have an unchanging variable for time scale but it is recommended that you use the Timeline’s animate properties feature to change the time scale over the course of the clip. You can do this by clicking the record button for the track and setting values for the time scale at various points along the clip (controlled using the play head). It is important to note that changing the timescale will affect the speed that the timeline executes. This means if you slow down time, the TimeDilation clip itself will take longer to finish. Keep this in mind when deciding how long your clips should be.

Time Scale	The value Time.timeScale will be set to for the duration of the clip.

ScreenFader

This type of playable is used to act on an UI Image component, typically one that has been stretched across a camera’s field of view, though this is not a requirement. The ScreenFader track binds to an Image component. It is designed to have easily controllable fades when doing things like switching scenes or cameras. Blending is enabled for this playable so having one fade transitioning into another is possible. Note that fading a full screen image in and out is not the only use of this playable, just it’s intended purpose. It simply sets the color of an Image component with optional fading. It could also, for example, be used to make a menu button flash.

Color	The Color that the Image component will be set to for the duration of the clip.

LightControl

This type of playable is bound to a Light component and is used to override its properties for the duration of the clip. The purpose of the LightControl playable is to produce light effects without the need for an animator. Blending is supported so transitioning to new lighting is easy.

Color	The Color that will override the Color property of the Light component that the track is bound to.
Intensity	The value that will override the Intensity property of the Light component that the track is bound to.
Bounce Intensity	This is also called Indirect Multiplier on Light Components. It will affect how much light will bounce off surfaces it hits. Setting a value for this in the inspector will override the value on the Light component that is bound to the track.
Range	The value that will override the Range property of the Light component that the track is bound to.

NavMeshAgentControl

The NavMeshAgent track binds to a NavMeshAgent component and sets its destination. The destination is set on the first frame that the clip’s weight is greater than 0.5. For most cases this will be on the clip’s first frame. Blending is not supported for this playable and since the destination is generally set on the first frame, having a duration of longer than one frame has no purpose. Having a longer duration will not be detrimental, it is just not required.

Destination	A Transform whose position will be used for the destination of the nav mesh agent.