

MovieClip Transition Effects V3

ABOUT

Did you ever wanted to have a single easy to use MovieClip transition component that you can use to perform all possible transitions between 2 different states of a MovieClip or between 2 different MovieClips?

The MCTE is a flash component that, used with any Movie Clip will give you time saving show/hide transitions, fastest transition through, transitions to, transitions from a specific state using built-in patterns, their presets and their parameters.

Think about the MovieClip Transition Effects like a Square Effect, or Waves Effect prototype multiplied with lots of other state of the art patterns and presets for getting the coolest animations without coding at all.

Custom MCTE Patterns are also developed and sold separately at jumpeyecomponents.com. The MovieClip Transition Effect is the ONE and ONLY component that you need to perform the most script generated transitions without scripting, some patterns might have custom parameters that can be customized, but, you can change patterns and presets, usually, without the need to change the code or anything.

Once you have used MCTE for animating In and Out a single MC, you will want to use it for all possible transitions. Using MCTE you will reduce interminable frame animations that takes a lot of time to make, into single frame animation that is done in seconds, and hey, if you like to change the pattern, preset or params, it can be done by a blink of an eye.

Compatibility

ActionScript: **2.0, 3.0**

Flash IDE: **Flash 8, Flash CS3**

FlashPlayer: **FlashPlayer 8, FlashPlayer 9**

Features

- built from scratch based on the **V3 Standards**
- created for fast and easy setup (effects can be applied in seconds)
- drag&drop over a MovieClip, and MCTE starts working on defaults
- works together with the Loader Pro V3 for faster integration, no code needed, just drag&drop the MCTE over a instance of Loader Pro V3, and when it loads up a new clip, it will automatically use the selected MCTE setup to make the transition
- all effects can be configured from the MCTE fast and easy
- 9 state of the art presets included with the MCTE
 1. **SquareScale** (built after the Square Effects)
 2. **Waves** (built after the Waves Effects)
 3. **Stripes** (stripes masks effects, with a lot of presets)
 4. **AlphaBars** (semi-transparency bar masks effect)
 5. **Alpha** (alpha transparency effect, with cool presets)
 6. **Blur** (easy blur in, blur out)
 7. **Color** (color transition)

8. **Scale** (scale transitions with customizable motion blur)
 9. **Slide** (sliding transitions with customizable motion blur)
- customizable transition tween, customizable ease and duration
 - change the patterns used for the same MovieClip in seconds, and there you have a new cool transition
 - 5 customParams (free slots) to be used for customizing all possible parameters of every pattern
 - easy to use separately sold patterns such as Cube, Flip Over, Genie and others
 - easy to use custom developed patterns
 - can be fully configured using XML

The story of the MovieClip Transition Effect V3

We wanted to help flash users to make show/hide MovieClip animations without knowing to code, in a fast reliable way. After we released the Square Effects component we started to think about a component that will be used for all possible transitions between 2 different states of a MovieClip and we came up with the idea of THE MC Transition Effect (MCTE).

With lots of patterns, having their own presets, with some customParams, a general Gain parameter, and with customizable tweening, MCTE V3 is simply the most easy to use, the most customizable transition effect ever.

We are currently developing other cool effects for the MCTE so all previous customers will not need to learn new component models after using MCTE. Once you understand how MCTE works, you can use it for all your animations, both frame animations, and dynamic animations, and change the patterns and presets at a blink of an eye.

The mass preorder rate that MCTE encountered convinced us to develop further patterns and packages (Both AS2.0 and AS3.0) and to improve the MCTE in order to fulfill your needs; this is why we want to assure you that working with the MCTE will be an easy job.

MCTE Transition Types

The MCTE is able to perform 3 types of transitions:

- MovieClip show/hide transitions (from a invisible state to a visible one and reverse)
- MovieClip from state to state transitions (eg. scale from 60% to 90%)
- transitions between 2 different MovieClips

MCTE together with Loader Pro V3

Loader Pro V3 supports MCTE transitions on loading content. When the content is loaded, if an instance of MCTE is attached to it (by drag&drop over), based on the MCTE setup, the Loader Pro V3 will perform show transitions or even transition between MovieClips. (eg. FlipOver, 3DCube patterns)

How to use the MCTE

It's easy, it's fast and it's reliable. There are 2 different modes you can add the MCTE to a MovieClip or to a Loader Pro V3.

1. Visual based.
Drag & drop the MCTE from the components panel right over an instance of a MovieClip on the stage, or over the Loader Pro V3, just like you would drag & drop a Adobe Scrollbar over a Text Field.
Then simply configure the parameters in order to get the transition you need.

2. Code based.
You can instantiate the MCTE from ActionScript, dynamically or statically using the following syntax:

```
var myMCTEInstance = new MCTE(targetInstance, [autoPlay:Boolean, transitionType:String, [patternName:String, preset:String, gain:Number, tweenType:String, easeType:String, tweenDuration:Number, customParam1:String, customParam2:String, customParam3:String, customParam4:String, customParam5:String] ])
```

Note. You can code on a MCTE instance even after you drag&drop it manually over the MovieClip. The MCTE should perform very intuitively, you should try to use it as you feel it to, and if you have questions don't hesitate to contact our support team.

Custom Patterns

The MCTE internal API allows us to further develop and sell patterns separately, all you have to do in order to use them as patterns for the MCTE instances, is to drag them inside your project library and use their name as values for the customPattern parameter of the MCTE.

As stated before, we sell patterns that are not initially included inside the MCTE. You can purchase those patterns without the MCTE, but they will be useless without it.

We also do develop patterns for direct customers, who want a special custom developed pattern for a specific transition. In order to get a custom developed pattern just send us an email.

PROPERTIES

Property	Type Component Inspector	Description
xmlPath	String Yes	Optional parameter, sets the path to a XML file that can configure the MCTE V3. If the xmlPath is set, all parameters set from script or Component Inspector will be overwritten by their correspondents from XML. Usage: mcteInstance.xmlPath = "config/loaderConfig.xml"
autoPlay	Boolean Yes	A boolean parameter that handles the automated transition. This should be set to true, if you use frame animation, and/or if you are not using actionscript to start the transition effect. Also when using the MCTE along with the Loader Pro V3, autoPlay

		<p>should be left "true" and the Loader's transitionType should be set to "none". Default:true</p> <p>Usage: mcteInstance.autoPlay = false</p>
_targetInstanceName	String Yes	<p>A string, linking to the instance on which the effect will be applied. When drag&dropping the MCTE to the stage, over a MovieClip, it automatically updates this property to the MovieClip's instance name.</p> <p>Usage: mcteInstance._targetInstanceName = "myMcName"</p>
patternName	String Yes	<p>The name of the pattern that will be used to perform the transition effect. Values:[customPattern, Alpha, Blur, Color, Scale, Slide, SquareScale, Waves, Stripes, AlphaBars] Default:SquareScale</p> <p>Usage: mcteInstance.patternName = "AlphaBars"</p>
customPattern	String Yes	<p>The name of the custom pattern that will be used to perform the transition effect. In order to use this parameter you need to set the patternName property value to customPattern.</p> <p>Usage mcteInstance.patternName = "customPattern" mcteInstance.customPattern = "Genie"</p>
preset	Polymorphic (String,Number) Yes	<p>The value of the preset that will be used for the selected pattern. Since the values may differ from a pattern to another, we used polymorphism, and the preset dataType is not set. Presets are usually numbers, and are documented separately for each pattern. You can use "random" value in order to have a random preset each time you use the MCTE.</p> <p>Usage: mcteInstance.preset = 12 mcteInstance.preset = "random"</p>
transitionType	String Yes	<p>The type of the transition that MCTE will do based on the selected pattern and preset. Values:[show, hide, transitionTo, transitionFrom] Default:show</p> <p>show will perform an appearing transition from invisible state to the visible one.</p> <p>hide will perform a disappearing transition from the visible state to the invisible one.</p> <p>transitionTo will perform a transition from the current state to a different one, based on customParams and the rest of the params, depending on the pattern type.</p> <p>transitionFrom will perform a transition from the param setting to the current state. transitionTo and transitionFrom also may be used by patterns that make transitions between MovieClips, not just between states.</p> <p>Usage: mcteInstance.transitionType = "show"</p>
gain	Number Yes	<p>Most patterns will mainly depend on the gain param, the gain is a number, that will be used by every pattern differently. In some cases the gain signifies the speed of an effect, in others it may signify the height of a mask that is applied during the effect, the percentage of effect applied, and so on.</p> <p>gain is separately documented for each effect, however a value of 50 will work for most of them. Default:100</p> <p>Usage:</p>

customParam1	String	<p>mcteInstance.gain = 50</p> <p>customParams are used by every pattern in order to create the transition effect. Depending on the type of the pattern, the customParams may be used for different parameters. Think about the customParam like a free slot, where you can plug your params in order to use the effect.</p> <p>Usage: mcteInstance.customParam1 = "30" mcteInstance.customParam1 = "false"</p>
customParam2	Yes	
customParam3		
customParam4		
customParam5		
tweenDuration	Number Yes	<p>Sets the duration of the slide tweening in frames. Default: 15</p> <p>Usage: mcteInstance.tweenDuration = 10</p>
tweenType	String Yes	<p>Sets the tweening type. Values:[Regular, Strong, Bounce, Back, Elastic, None] Default: Regular</p> <p>Usage: mcteInstance.tweenType = "Back"</p>
easeType	String Yes	<p>Sets the ease type for the sliding transition tween. Values:[easeOut, easeIn, easeInOut, none] Default: easeOut</p> <p>Usage: mcteInstance.easeType = "easeOut"</p>

AS2 EVENTS

Event	Description
onTransitionStart	<p>Event is triggered when the transition starts.</p> <p>Usage: myEventListener = new Object(); myEventListener.onTransitionStart = function(){ trace("onTransitionStart event "); } mcteInstance.addEventListener("onTransitionStart",myEventListener);</p>
onTransitionEnd	<p>Event is triggered when the transition ends.</p> <p>Usage: myEventListener = new Object(); myEventListener.onTransitionEnd= function(){ trace("onTransitionEnd event "); } mcteInstance.addEventListener("onTransitionEnd",myEventListener);</p>

AS3 EVENTS

Event	Description
TRANSITION_START	<p>Event is triggered when the transition starts.</p> <p>Usage: import com.jumpeye.Events. MCTEEvents; myMcteInstance.addEventListener(MCTEEvents.TRANSITION_START,transitionStartHdl); function transitionStartHdl (evt: MCTEEvents):void{ trace("TRANSITION_START "+evt.target); }</p>
TRANSITION_END	<p>Event is triggered when the transition ends.</p> <p>Usage:</p>

```

import com.jumpeye.Events. MCTEEvents;
myMcteInstance.addListener(MCTEEvents.TRANSITION_END,transitionEndHdl);
function transitionEndHdl (evt: MCTEEvents):void{
    trace("TRANSITION_END "+evt.target);
}

```

METHODS

Method	Description
transitionEffect	<p>The method used to perform the transition effect, this is only for advanced use, you don't need to use this method in order to perform transitions on a frame animation basis. All parameters are optional, however, the transitionType (the first parameter) should be entered.</p> <p>Syntax: mcteInstance.transitionEffect([transitionType:String, [patternName:String, preset:NotSet, gain:Number, tweenType:String, easeType:String, tweenDuration:Number , customParam1:String, customParam2:String, customParam3:String, customParam4:String, customParam5:String]]);</p> <p>Usage: mcteInstance.transitionEffect("show"); mcteInstance.transitionEffect("show","SquareScale","6",100,"Strong","easeInOut",20, "", "100","20"); or targetInstance.MCTE.transitionEffect("show");</p>
removeEffect	<p>It instantly removes the effect and the MCTE from a specific target.</p> <p>Usage: mcteInstance.removeEffect(); or targetInstance.MCTE.removeEffect();</p>
show	<p>Performs a show transition based on the setup that the MCTE already has.</p> <p>Usage: mcteInstance.show(); or targetInstance.MCTE.show();</p>
hide	<p>Performs a hide transition based on the setup that the MCTE already has.</p> <p>Usage: mcteInstance.hide(); or targetInstance.MCTE.hide();</p>
transitionTo	<p>Performs a state-to-state transition on a single MovieClip (from first state to second) or a mc-to-mc transition on two MovieClips (from first clip to second) based on the setup that the MCTE already has.</p> <p>Usage: mcteInstance.transitionTo(); or targetInstance.MCTE.transitionTo();</p>
transitionFrom	<p>Performs a state-to-state transition on a single MovieClip (from second state to first) or a mc-to-mc transition on two MovieClips (from second clip to first) based on the setup that the MCTE already has.</p> <p>Usage: mcteInstance.transitionFrom(); or targetInstance.MCTE.transitionFrom();</p>

BUILT IN PATTERNS

Pattern	Description/Value : Preset
Alpha	<p>Alpha transitions, simple alpha transparency masks. No custom params needed. Supports show/hide transitions.</p> <p>Gain parameter functionality: sets the width/height of the alpha gradient used by presets.</p> <p>Note: The value of gain parameter is recommended to be less than or equal to value of width/height of the target clip. If it is greater than value of width/height of target clip, it will be set to maximum size possible.</p> <p>Quick Tip You can make pretty interesting animations by customizing the gain.</p> <p>Presets Random String value: random : uses a random preset each time depending on the customParams that are set, if one preset requires more customParams that are already set, the random filter will avoid it.</p> <p>Number values: 1 : left to right 2 : right to left 3 : top to bottom 4 : bottom to top 5 : bottom left to top right 6 : top right to bottom left 7 : top left to bottom right 8 : bottom right to top left 9 : left and right to center 10 : center to left and right 11 : bottom and top to middle 12 : middle to top and bottom</p>
Scale	<p>Supports transitionTo/transitionFrom transitions.</p> <p>Gain parameter functionality: sets the motion blur quantity. CustomParam1:String (numerical value) – sets the value of the scale for the target clip. The range of values is "0" – "100".</p> <p>Quick Tip Looks popping with elastic tweenType, supports the transition to a different scale.</p> <p>Presets Random String value: random : uses a random preset each time depending on the customParams that are set, if one preset requires more customParams that are already set, the random filter will avoid it.</p> <p>Number values: 1 : center 2 : right to left 3 : left to right 4 : bottom to top 5 : top to bottom</p>
Slide	<p>Performs sliding in and out transitions. Supports show/hide transitions.</p> <p>Gain parameter functionality: sets the motion blur quantity. customParam1: String (numerical value) – horizontal position for preset no. 5 customParam2: String (numerical value) – vertical position for preset no. 5</p> <p>Quick Tip You can make pretty interesting animations by customizing the gain and duration, since the</p>

blur depends on both gain and speed.

Presets

Random String value:

random : uses a random preset each time depending on the customParams that are set, if one preset requires more customParams that are already set, the random filter will avoid it.

Number values:

1 : left to right

2 : right to left

3 : top to bottom

4 : bottom to top

5 : custom - requires customParam1 (x) and customParam2 (y)

SquareScale

Performs square effects based on many different presets.

Supports show/hide transitions.

Gain parameter functionality: sets the time delay between the appearing of the next square in matrix.

customParam1: String (numerical value) – width of rectangle. Default: "40"

customParam2: String (numerical value) – height of rectangle. Default: "40"

Note: If customParam2 isn't set, it will be equal to customParam1.

Quick Tip

You can make pretty interesting animations using the possibility to set the height and width of the rectangles separately, and combine that with the customized tweening.

Presets

Random String value:

random : uses a random preset each time depending on the customParams that are set, if one preset requires more customParams that are already set, the random filter will avoid it.

Number values:

1 : left to right

2 : right to left

3 : top to bottom

4 : bottom to top

5 : middle to left and right

6 : left and right to middle

7 : top and bottom to middle

8 : middle to top and bottom

9 : top&left to bottom&right

10 : bottom&right to top&left

11 : bottom&left to top&right

12 : top&right to bottom&left

13 : left&top&right&bottom to middle

14 : middle to left&top&right&bottom

15 : spiral to middle - counterClockwise

16 : middle to spiral - clockwise

17 : spiral to middle clockwise

18 : middle to spiral- counterClockwise

19 : random

20 : one-one

Waves

Performs waves effects based on many different presets.

Supports show/hide transitions.

Gain parameter functionality: sets the height of the lines in pixels, and it rounds them up in order to cover all the boundaries of the target.

customParam1: String (numerical value) – the duration between waves, the incremental value, not the whole transition duration.

Quick Tip

You can tweak pretty much the transitions using the tweenType and easeType parameters. The effect is very suited for showing text contents. Adapt the size of the line to the font size + vertical spacing and get a nice text show effect.

Presets

Random String value:

random : uses a random preset each time depending on the customParams that are set, if one preset requires more customParams that are already set, the random filter will avoid it.

Number values:

- 1 : left to right s1
- 2 : left to right s2
- 3 : right to left s1
- 4 : right to left s2
- 5 : top to bottom s1
- 6 : top to bottom s2
- 7 : bottom to top s1
- 8 : bottom to top s2
- 9 : middle to left and right s1
- 10 : middle to left and right s2
- 11 : left and right to middle s1
- 12 : left and right to middle s2
- 13 : middle to top and bottom s1
- 14 : middle to top and bottom s2
- 15 : top and bottom to middle s1
- 16 : top and bottom to middle s2

Stripes

Performs very cool stripes effect, vertical and horizontal lines/bars that uncovers the target. Supports show/hide transitions.

Gain parameter functionality: sets the height of the lines in pixels, and it rounds them up in order to cover all the boundaries of the target.

Quick Tip

You can tweak pretty much the transitions using the tweenType and easeType parameters. The effect is very suited for showing text contents. Adapt the size of the line to the font size + vertical spacing and get a nice text show effect.

Presets

Random String value:

random : uses a random preset each time depending on the customParams that are set, if one preset requires more customParams that are already set, the random filter will avoid it.

Number values:

- 1 : left to right
- 2 : right to left
- 3 : top to bottom
- 4 : bottom to top

AlphaBars

Performs transitions using vertical and horizontal transparency bars. A similar alpha square effect is developed separately for MCTE. Supports show/hide transitions.

Gain parameter functionality: sets the height of the lines in pixels, and it rounds them up in order to cover all the boundaries of the target.

Quick Tip

You can make pretty interesting animations by adjusting the height of the lines; large bars/lines look very modern.

Presets

Random String value:

random : uses a random preset each time depending on the customParams that are set, if one preset requires more customParams that are already set, the random filter will avoid it.

Number values:

- 1 : left to right
- 2 : right to left
- 3 : top to bottom
- 4 : bottom to top

5 : middle to top and bottom
6 : top and bottom to middle
7 : left and right to middle
8 : middle to left and right

Blur

Performs blur transitions based on one single default preset.
Supports show/hide transitions.

Gain parameter functionality: sets the blur state values for blurX and blurY; however they can also be set separately using the customParam1 for blurX and customParam2 for blurY. customParam3:String (numerical value) - blur quality - Values:["1","2","3"].

Quick Tip

You can make pretty fast blur animations by customizing the gain and customParams separately.

Color

Performs color to and color from transitions.
Supports transitionTo/transitionFrom transitions.

Gain parameter functionality: the gain parameter is not used by this pattern.

Quick Tip

You can "white out" any image with this nice pattern.

Presets

Random String value:

random : uses a random preset each time white or black.

String values:

white : color from/to white

black : color from/to black

Hexa Number values:

custom color (eg. 0xFF0000) : color from/to custom color