

The behaviours may be one of the most powerful addition to Multimedia Fusion and certainly one of the most ignored feature of the entire package! The tutorial will explain what exactly are behaviours, where to find them, how to use them and what are the benefits of using them.

☑ **What are behaviours.**

▲ Behaviours are small lists of events, specific to one kind of objects. They are stored as a property within the other properties. This allow you to pre define the reactions of one object, and create automated objects that work immediatly when dropped on the frame, with no need of extra events.

Applications are endless :

- ▲ Automated buttons, that reacts under the mouse
- ▲ Objects with complex movements, and preset rebounds
- ▲ Preprogrammed system objects, for example a MCI object that plays sounds from the disc and manage them automatically.
- ▲ Etc!

A proper use of the behaviours in a complex project can dramatically reduce the number and the complexity of the main list of events.

▲ We are going to discover the behaviours by creating an automated button that automatically highlights under the mouse.

- ▲ Start Multimedia Fusion, create a blank application and open the frame editor.
- ▲ Open the library window (Control R, or from the menu View, Toolbars, Library window) , and choose **Buttons and pointers / Media - Widgets #1**. Drag and drop the object **Play 2** on your frame.

▲ This button contains two images, the flat button affected to the right direction of the stopped animation (direction number 0), the highlighted button affected to the left direction of the same animation (direction number 16). So, if we want to highlight the button, we simply should chage the direction of the button to the right or to the left, when the mouse is above it or not.



Direction 0



Direction 16

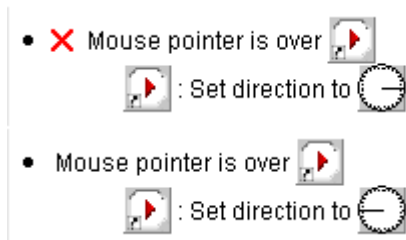
- ◆ Highlight the **Play 2** object and in the Properties Toolbar select the Events tab and click on the New button.



- ◆ This will create a new behavior named "**Behaviour #1**". Click on the edit area and finally on the New button. This will open the usual event editor editing the behaviour of the button. Notice that the icon of our button has a "shortcut" arrow appended to it, like a Windows shortcut. More on this later.

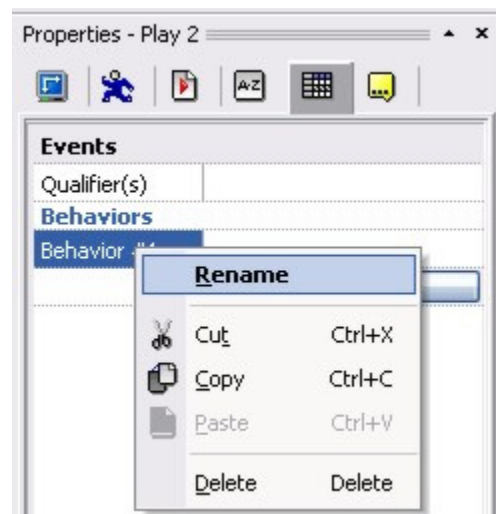


- ◆ Enter the following events:



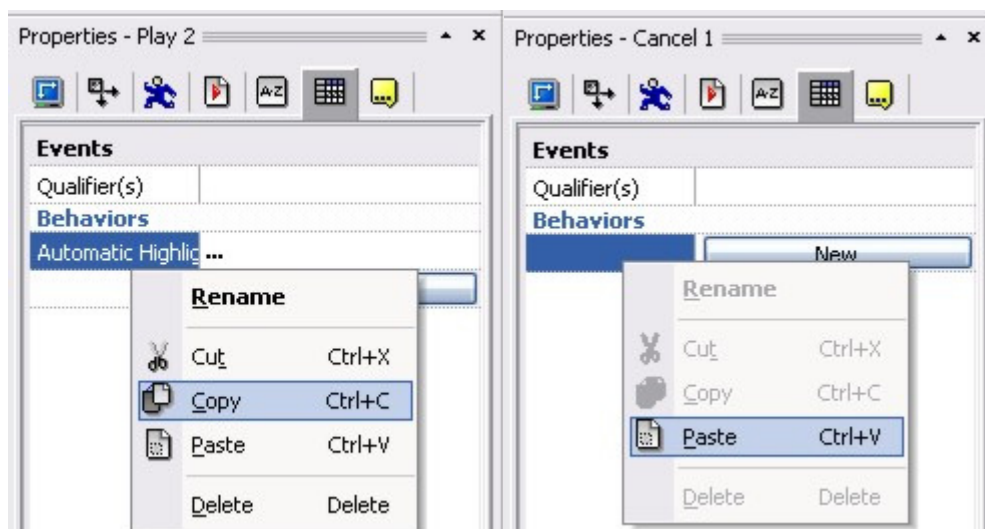
- ◆ This very simple list of events makes the object react to the mouse. Run the application and you will see the object change when the mouse crosses over it.

- ◆ Lets rename the behaviour to a proper name: in the Properties toolbar right click on the name "**Behavior #1**" choose "rename" and enter : "**Automatic Highlight**", click OK to save.



Properties are stored within the object.

- 🔥 Now close the event editor, close the frame editor so that only the project window is left,
- 🔥 Open a new, blank application.
- 🔥 Open the frame editor of the blank application.
- 🔥 In the project window, display the list of objects of our first application, and drag and drop our button onto the newly created frame.
- 🔥 Click in the frame editor to select the window, and run the new frame: the button works automatically, its behaviour was stored inside of it and dropped at the same time in the application.
- 🔥 Now, open the library window (Control R), and drag & drop the button named **Cancel 1** into the frame. If you run the frame, you will see that this button does not react to the mouse.
- 🔥 Select the **Play 2** button and from the Properties toolbar right click on the **Automatic Highlight** behavior and choose copy. , Now select the **Cancel 1** button and in the Properties toolbar right click on the blank area next to the New button, and select Paste..



- 🔥 Run the frame, you should see both buttons reacting to the mouse.
- 🔥 This little demonstration has made you understand hope the hidden power of the behaviours. They will make you gain time, they are re-useable, and will greatly reduce the number of events in the main event list (imagine that you have 10 buttons like these two!).

☑ Behaviours in details.

- 🔥 A simple list of events, stored in the object itself.
- 🔥 A behaviour does not contain direct references to the objects (as in the event editor), but shortcuts to the object. This is the reason why a small arrow is pasted over the icon of the object.



Shortcuts to objects

- 🔥 The main object in a behaviour, is the object who owns it. This object will always be present in the behaviour event editor, and cannot be removed. If you drag & drop the behaviour from one object to another, the main object and icon will be renamed to the new object (in the example above, **Play 2** has been replaced by **Cancel 1** after the drop).
- 🔥 Behaviours can reference other objects from the frame. The reference is made from the name of the object: if the referenced object is present on the frame, then the events are activated, if not, they simply will have no effect. This feature make them even more powerful : you can prepare objects that will automatically trigger other objects if they are lying beside them, or stay quiet if not.





Lets program a quick demonstration.

- 🔥 Close the second application (the one with the Cancel button), we do not need it.
- 🔥 Reopen the frame of the first application (the one with the Play button),
- 🔥 Create an AVI object in the frame. Make it point to whatever video you want, and uncheck "Play at start" in the video Properties toolbar.
- 🔥 Rename this object to "Video".
- 🔥 Reopen the behaviour event-editor of our play button as before (Properties toolbar / Events tab / Edit). You can see a new object displayed on the right of the object bar : the "Import" object.
- 🔥 Click on this **import** object will display the other objects present in the frame at that moment.



🔥 Select the "Video" object and click OK: it should appear in the event editor.

Add the following events to the list:

- User clicks with left button on 
 -  : Play video
- User clicks with right button on 
 -  : Stop video

🔥 These basic events will start the video when the user left-clicks on the button, and stop it when he right-click.

Of course, if you run the frame, it should work : the behaviour of the button changes the state of the video.

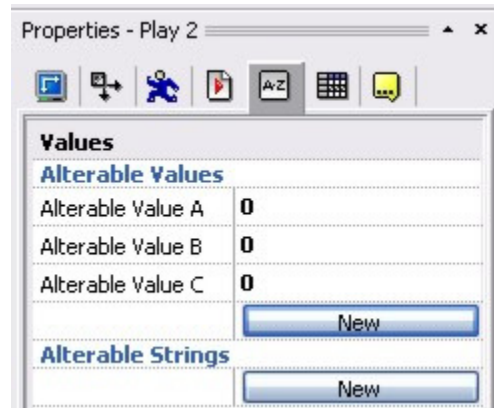
Lets redo the same manipulation as before.

- 🔥 Open a second application, open the frame editor;
- 🔥 Drag and drop our new **Play 2** button into the frame.
- 🔥 Now create a second video, same as before (not to play at start), but do not rename it : leave the default name **AVI**. If you run the application, it won't be triggered by a click on the play button : upon running the application, Fusion links the different objects shortcuts with the objects themselves. Our behaviour contains references to a **Video** object, thus Fusion cannot find it, and inactivates the play/stop actions.
- 🔥 Rename the **AVI** object to **Video**, and run the frame : it works now, the link is established.
- 🔥 Once the link is established, you can safely rename the other objects, Fusion will preserve the links by renaming the shortcuts inside the behaviours. For example rename **Video** to **Video 2**, open the **Automatic highlight** and see that it has been updated.

Things to know about the behaviours and shortcuts.

- 🔥 You can create or shoot objects from within a behaviour, but the original object must have been previously dropped in the frame, otherwise, Fusion will not establish the link.
- 🔥 You cannot drag & drop a behaviour from one kind of object to a different kind (example active object behaviour onto edit box)

- 🔥 You can play sounds and musics from within a behaviour, but here too, the sound must be inserted somewhere in the real list of event. The link is established from the name of the sound.
- 🔥 As behaviours are dealing with shortcuts and not the real objects, there is a small limitation related to user animation of objects and names of alterable values: Fusion cannot safely get the real name of the alterable values of an object, they will always be displayed as "Alterable value A/B/C...", regardless of their true name.



Alterable value are not displayed with their true names

- 🔥 When using **Change animation sequence** action, Fusion will only display the 12 first animations of an active object (the default animations). Extra animations, will not be displayed. You can access them though by using a calculation instead of a direct link to the animation : the first user animation is number 12, the second 13 etc...

☑ **Behaviours welcomed !**

- 🔥 Please send us your behaviours (with a small documentation for each), we will propose them for upload on Clickteam. If everyone shares his own work with the other, we will quickly build a library of premade behaviours that will save a tremendous amount of time for everyone!

Copyright © 2006 Clickteam. All rights reserved.

Written by François Lionet & D.T. Holder. Please **do not copy or re-use** without written permission.