

# How to use Load Movie

## Using the Load Movie action

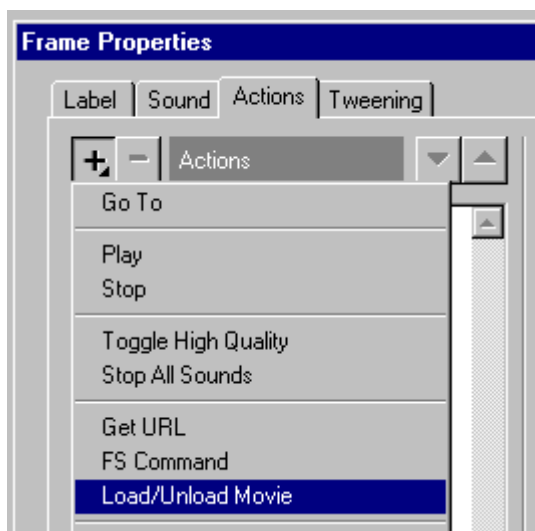
The Load Movie action is used to load an external SWF file into the current Flash timeline without reloading a new HTML page.

There are several advantages to using Load Movie instead of a larger, single file SWF.

- Load Movie plays the new SWF without making the browser load another HTML page. This avoids the pause associated with reloading. This feature can be used to string together multiple SWF files without large delays.
- Because the project is broken up into multiple movies, the individual SWF files are smaller in size. Smaller files load faster, and manage memory more efficiently.
- A complex Flash interface can be created that does not rely on a single Flash file. This simplifies editing, because an individual, smaller SWF can be revised without affecting the other movies. Multiple authors can collaborate on the same visual interface.

## Load Movie: The basics

1. Double-click a keyframe or button instance to open either the Frame Properties or Instance Properties dialog boxes. Both dialog boxes contain identical Actions tabs.
2. Choose the Actions tab. Assign the action Load/Unload Movie.



3. Load Movie actions require that three items are set:
  - **Action:**  
Tells Flash what the movie should do. One of these actions must be selected. Choose "Load movie into location".
  - **URL:**  
Points to the location of the external SWF to load.

In this basic example, no path is stated. This will tell Flash to look for "myNavClip.swf" in the same folder on the server that the current Flash movie is playing from. The simplest solution is to keep all Flash SWF files in the same directory on the server, so that no path statements will be required in the URL field.

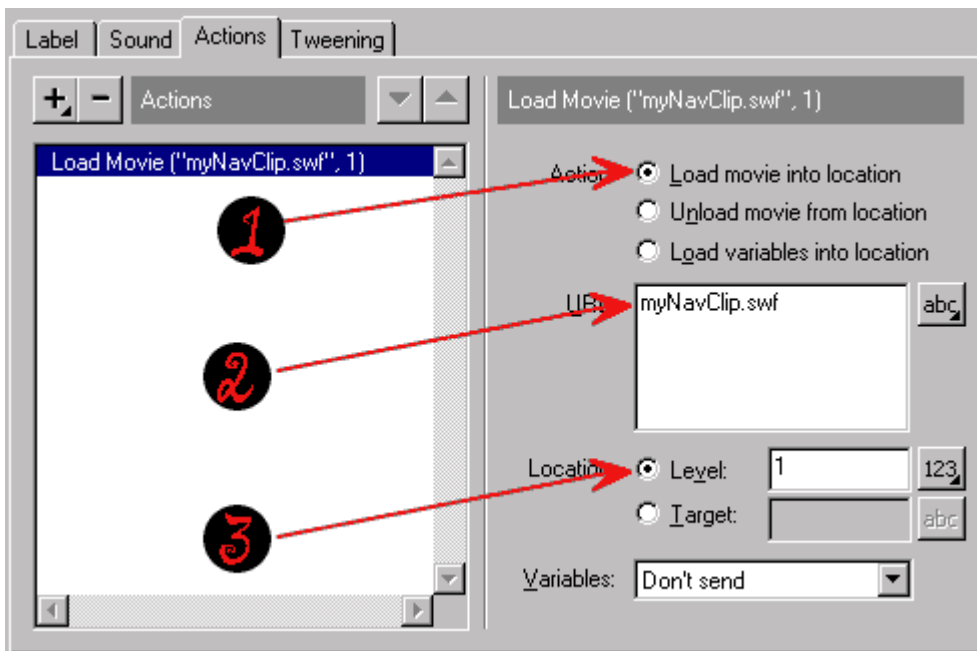
See [Additional information](#) below for references about absolute and relative paths in Flash.

o **Location:**

Flash can play multiple timelines simultaneously. Each timeline is assigned a level. The main timeline is always Level0. Level0 sets the frame rate, background color and frame size for all the other loaded movies. Additional timelines load into higher numbered levels and are displayed in a "stacking" order with level0 being the bottom-most level.

When using the Load Movie action, a level must be specified, to define where the SWF will be loaded. If loading one movie only, any level 1 or higher will work fine.

A movie can also be loaded into level0. This will replace the contents of the main timeline with the loaded movie. This is an effective method of stringing multiple SWF files together in a continuously running show.



### Load Movie: Positioning

If the loaded movie is loaded into a level, it is positioned relative to the upper left corner of the movie that loaded it.

In the example below, drag the smaller movie into the larger movie to see the positioning of the smaller movie when loaded with a Load Movie action.

Notice how the loaded movie aligns itself to the upper left hand corner. The simplest way to insure correct alignment of the loaded movie on top of the parent movie is to create all the files with the same stage size.

### Aligning the Stage

The following example illustrates what happens when a movies with the same stage size are loaded:

- Click "Do it!" to simulate loading the "loaded movie" on a layer above the main timeline. Both movies are the same size. Because both movies are the same size, the loaded movie is loaded with its upper left corner

at 0,0 of the main timeline. This places the items in the loaded movie in exactly the same position, only on a layer above the main timeline.

Aligning loaded movies with differing stage sizes can be problematic. Creating all movies at the same size removes this difficulty.

- Notice that the blue background color of the loaded movie disappears. The background color of the loaded movie is always made transparent, allowing loaded movies to act as interfaces floating above the lower levels beneath them.

## Targeting instances

Load Movie can be used to replace a loaded movie in a level. It can also be used to replace a movie clip instance on the stage. The loaded movie takes on the same characteristics of the movie clip on the stage, such as frame rate and background color, as well as instance name. However, when targeting a movie clip with Load Movie, the location of the replacement is NOT the upper left corner of the movie clip. Instead, the loaded movie is placed with its upper left corner at the center point of the movie clip it's replacing. (see "Using Flash", page 112 for the definition of an object's registration point.)

In the following example, drag the 'loaded movie' onto the target instance to simulate loading a movie to a target. In this example the green crosshair indicates the registration point of "target instance". Notice that the upper left corner of 'loaded movie' locates itself at this registration point when the Load Movie action loads a SWF into a movie clip.

## How to use the Tell Target action

- [Timelines](#)
- [When to use Tell Target](#)
- [How to use Tell Target](#)

## Timelines

Every Flash movie has a main timeline. With movie clips and the ability to load movies into a [Flash Player](#)\*, multiple timelines can exist at once. Multiple timelines run independently and the structure is similar to a directory structure. The main movie timeline is similar to the root directory. Movie clips are similar to subdirectories of the main timeline.

The following is an example of multiple timelines.

MovieX.swf is the main movie and is on level 0 in the player. It contains two movie clips MCA with an Instance name of Jack and MCB with an Instance name of Jill. Within MCA there are also two movie clips MCM with an Instance name of Bert and MCN with an Instance name of Ernie. Also, there is another movie, MovieY.swf, loaded into the player (with a Load Movie action) on level 3. In MovieY.swf there is a movie clip MCT with an Instance name of Smith.

Basically the outline structure within the player is as follows:

- MovieX.swf (in level 0)
  - MCA (Instance name: Jack)
    - MCM (Instance name: Bert)
    - MCN (Instance name: Ernie)
  - MCB (Instance name: Jill)

- MovieY.swf (in level 3)
  - MCt (Instance name: Smith)

This means that there are 7 timelines in the player at once. Any timeline can control any other timeline. This is done using the Tell Target action.

## When to use Tell Target

The Tell Target action is used to send an action from one timeline to another.

Using the sample above, here are some examples of when a Tell Target action would be used:

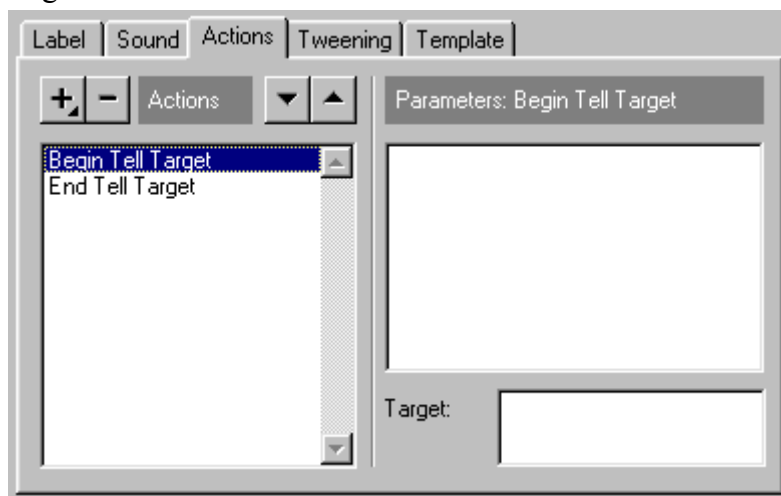
- If you have a button in MovieX and you want it to Stop movie clip MCa.
- If you want movie clip MCn to Play as soon as movie clip MCm has reached its last frame.
- If you want to Goto and Stop at frame labeled 'zephyr' of MovieX when a button in movie clip MCb is pressed.
- If you want movie clip MCa to Goto and Play at a frame labeled 'reset' when MovieY has reached its last frame.
- If you want a button in movie clip MCt to have MovieX Play.

These actions are all possible.

**Note:** A timeline must currently be in the Flash Player in order to be targeted. A timeline of a movie clip is present in the Player when the playback head is positioned in the frame that the movie clip exists. For example: if there is a movie clip placed on the timeline in frames 1 - 20, it can be targeted while the movie is playing or stopped at any frame between 1 and 20. Once the movie reaches frame 21 the movie clip is no longer in the Player and cannot be targeted.

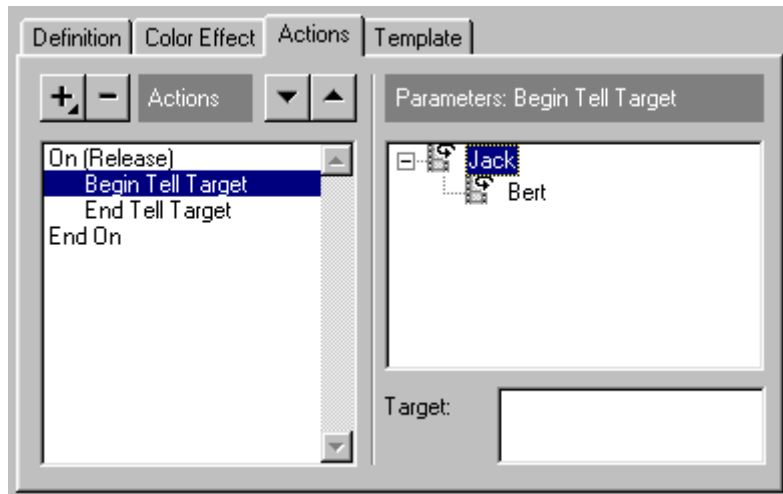
## How to use Tell Target

There is one field to fill in for the Tell Target action - the Target: field. The target is the timeline that is to be controlled. Any actions nested within the Begin Tell Target and End Tell Target actions will be directed at the targeted timeline.



In some situations, you will be able to see movie clips that are accessible in the window above the Target field. If this is the case and one of these movie clips is to be targeted, you may simply double-click it in the window. The

correct information will automatically be entered in the Target field. At other times, the timeline that you wish to target will not be in the window and the path will need to be entered manually.



## Main Timelines

The target for the main timeline can most easily be specified as a forward slash character: /  
This can be used from any place in the current movie to control the current movie's main timeline.

The main timeline can also be controlled by using the ../ structure to move back up levels. Using this syntax to control MovieX's main timeline from movie clip MCa or MCb, the Target is ../

Similarly, to control MovieX's main timeline from movie clip MCm or MCn, the Target is ../..

To address the main timeline for a loaded movie use the following syntax:

\_levelN/, where N represents the number of the level that the movie is loaded on

To control MovieX's main timeline from any other timeline with this syntax, the Target is

\_level0/

To control MovieY's main timeline from any other timeline with this syntax, the Target is

\_level3/

## Movie Clips

Instance names are used to target movie clip instances in nearly every situation. The distinction between a movie clip and an Instance of a movie clip is important. A movie clip is a Symbol in the movie's Library. An Instance of the movie clip is the placement of the movie clip on the work area. There can be more than one instance of a movie clip. Each Instance name for that movie clip needs to be unique.

From the main timeline, to control a movie clip use the movie clip's instance name in the Target field. To control movie clip MCt (Instance name: Smith) from the main timeline of MovieY, the target is **Smith** . The Target can also be /**Smith** or **\_level3/Smith** .

To control movie clip MCm (Instance name: Bert) from MovieX, the Target is **Jack/Bert** or /**Jack/Bert** or **\_level0/Jack/Bert**

To control movie clip MCm (Instance name: Bert) from movie clip MCa, the Target is **Bert** or /**Jack/Bert** or **\_level0/Jack/Bert**

To control movie clip MCm (Instance name: Bert) from movie clip MCn, the Target is **../Bert** or /**Jack/Bert** or **\_level0/Jack/Bert**

To control movie clip MCm (Instance name: Bert) from MovieY or movie clip MCt, the Target is **\_level0/Jack/Bert**

The only situation when an Instance name is not required to control a movie clip is when a nested clip is controlling a movie clip above it. For example, movie clip MCm or MCn can control movie clip MCa with a Target of ../

To make loaded movies line up correctly with target instances of movie clips on the stage, change the registration point of the movie clip. This can be done by editing the movie clip symbol and moving the contents, so that its center point is in the upper left corner.

### **Frame rate**

A loaded movie also inherits the parent movie's frame rate.

The movie below is playing at 10 frames per second (FPS). Its frame counter is updated with every new frame. This movie is playing at 1 frame per second. Click the "Load the 10 fps movie" button to find out what happens when it is loaded into the parent movie.

Notice that the loaded movie clip uses the frame rate of the timeline that loaded it.

### **Load Movie: Relative paths**

Using relative paths with load movie can be confusing. Since any timeline can perform a load movie action, what path is the movie being loaded relative to? Is it relative to the main timeline at `_level0`? Or is it relative to the timeline that performed the Load Movie action?

The answer is simple: Loaded Movies are always relative to the timeline that loaded them.

Please refer to [Using Relative URLs with Get URL Action](#) (TechNote 4157) for a discussion of relative paths, which contains information that is relevant to loading movies.