

## CNET Networks' Flash Tracking Solution

### Overview

1. Introduction
2. Downloading and Initializing the CNET Networks Tracking Class
3. Implementing and Tracking Submovie Loads
4. Tracking Exit Links
5. Tracking Internal Events
6. Implementing and Tracking a Flash Video (FLV)
7. Implementing and Tracking Expand Ads
8. Publishing Your FLA Files
9. Testing Your Flash Creative
10. Next Steps and Quick Links

### 1. Introduction

This guide explains how to use our tracking class file (provided in Section 2) to add CNET Networks tracking into a Flash creative. The process described in this document should be used for any CNET Networks-hosted Flash creative that meets one of the following criteria:

- Contains multiple movies
- Contains Flash Video (FLV)
- Contains internal events that need tracking
- Is an expand ad

If you are building a single-movie Flash creative with no internal events or Flash Video, please use the clickTAG tracking document:

<http://www.cnetnetworks.com/i/b20/clickTAG.pdf>

CNET Networks can record and report tracking data for five types of actions: movie subloads, exit links, internal events, Flash video player events, and expand events. The same tracking class file can be used for any and all of those action groups. Each group has a limited number of actions available for tracking:

- **Movie Subloads:** A total of four movie subloads can be tracked. One of these subloads may be a Flash Video (FLV).
- **Exit Links:** A total of 10 exit links can be tracked.
- **Internal Events:** A total of 10 internal events can be tracked. Internal events can be triggered by rollovers, button clicks, or any other interaction allowed in Flash including select, drag, and scroll.
- **Flash Video Player Events:** A total of 10 video player events can be tracked and are limited to the following:
  - Video Start
  - Video Middle
  - Video End
  - Mute
  - Unmute
  - Pause
  - Unpause

- Replay
  - Fast-Forward
  - Rewind.
- **Expand & Collapse:** There is one event for an expand action and one event for a collapse action.

## 2. Downloading and Initializing the CNET Networks Tracking Class

### Step 1: Download and unzip the CNET Networks Tracking Class file

	<a href="http://www.cnetnetworks.com/advertise/specifications/tracking_files/cnetnetworks_tracking_class.zip">www.cnetnetworks.com/advertise/specifications/tracking_files/cnetnetworks_tracking_class.zip</a> Includes: <i>cnetnetworks.as</i> Download and unzip this ActionScript file (ZIP)
---	---

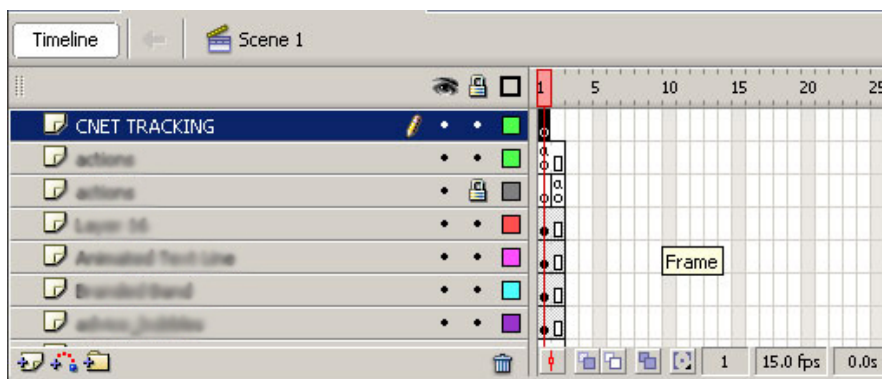
### Step 2: Save the ActionScript file to the correct directory:

Save the file to the same directory that does or will contain your creative source files (FLAs).

### Step 3: Set up your creative file and initialize the tracking class

In this step, the tracking code is initiated in your Flash file and elements to be tracked are identified and defined.

1. Open your main FLA file (if you have multiple Flash movies, this would be your main SWF file that will load the other SWF files).
2. Create a new layer named “CNET TRACKING” on the top level of your creative’s main timeline for Scene 1.



3. Select the Keyframe on Frame 1, Scene 1 of the CNET TRACKING layer and open the ActionScript panel. Add the following ActionScript:  

```
new cnetnetworks();
```
4. The CNET Networks tracking code is ready to use. You may now add tracking commands to the rest of your movie.

### 3. Implementing and Tracking Submovie Loads

This section provides instruction on loading and tracking SWF movies only. The CNET Networks tracking class will track up to 4 SWF loads or 3 SWF loads and 1 FLV load. For instructions on loading a FLV video into your creative, see the “Implementing and Tracking a Flash Video (FLV)” section below.

To set up tracking of a movie subload, you must first define the following items for each subload:

- The movie number (an integer between 1 and 4, identifying which of the four subloads you wish to track)
- The SWF name (the filename you gave to your submovie). Your submovie should be saved in the same directory as your main creative.
- The movie clip target (the instance name of the movie clip into which you wish to load your submovie)

Next, make a function call in the actions panel of any object or frame you wish to trigger the movie. The function is called `cnetnetworks.getMovie(id, SWF_name, movie_target)`. See the following example:

```
on(release){
    cnetnetworks.getMovie(1, "movie_Subload_1.swf", "movieclip_name");
}
```

This function will load the file “movie\_Subload\_1.swf” into the movie clip “movieclip\_name” and track the event as Subload 1. Make sure to enclose the SWF name and movie clip target in quotes. Add this line again for each movie subload in your creative, substituting the correct movie number, filename, and movie clip target.

**Please note:** Your movie clip target should always be relative to the `_root` of your movie. If you would like to make a subload within a subload, the movie clip target should reflect this hierarchy. For instance, if you would like to load a second submovie into a movie clip called “child\_clip” inside the movie clip you loaded above, the registerMovie call would look like this:

```
on(release){
    cnetnetworks.getMovie(2, "movie_Subload_2.swf",
    "movieclip_name.child_clip");
}
```

**Please note:** The CNET Networks Tracking class does **not** support loading movies into layers and does **not** support loading submovies from outside domains. All subloads must be hosted by CNET Networks.

To unload a movie, simply call ActionScript’s usual `unloadMovie()` function. There is no tracking available on movie unloads.

#### 4. Tracking Exit Links

You may track up to 10 exit links. This limit applies to the creative as a whole, not each subload.

##### Setting up an Exit Link

To set up an exit link, make a function call in the actions panel of any button or movie clip within an `on(release)` event. The function is called `cnetnetworks.exit(id, description)`.

Replace `id` with the exit link number you'd like to track (an integer between 1 and 10). Replace `description` with a phrase describing the exit link you're tracking. This description will show up in your reports to help you identify which links are which. See the following example:

```
on(release){
    cnetnetworks.exit(1, "Your exit link description");
}
```

**PLEASE NOTE:** The URLs for your exit links will be entered into the ad system when your ads are ready for trafficking. As a result, none of your exit links will be active until the ads are running inside the CNET Networks ad system.

**ALSO NOTE:** By default the URL will open in a new window. If you require the URL to open in the same window or in a different target window, please see the Advanced Notes below.

##### ADVANCED NOTE:

The exit function also supports some advanced functionality to aid in non-standard usage. The secondary function definitions include:

```
cnetnetworks.exit(id, description, target)
cnetnetworks.exit(id, description, target, url)
```

These alternate definitions can be used if a link needs to be targeted to a specific browser window or if the URL must be defined inside the ad.

For instance, if you would like to target your exit link to the same browser window:

```
on(release){
    cnetnetworks.exit(1, "Your exit link description", "_self");
}
```

Or if you would like to open a dynamic link URL (from an RSS feed for instance where `RSS_URL` is the name of your URL variable):

```
on(release){
    cnetnetworks.exit(1, "Your exit link description", "_blank", RSS_URL);
}
```

## 5. Tracking Internal Events

You may track up to 10 internal events. This limit applies to the creative as a whole, not each subload. Events can be triggered by rollovers, clicks, or any other interaction allowed in Flash, including select, drag and scroll.

### Setting up an Internal Event

Wherever in your movie you would like an event to be triggered, simply add the code:

```
cnetnetworks.event(id, description).
```

Replace `id` with the internal event number you'd like to track (an integer between 1 and 10). Replace `description` with a phrase describing the event. This description will show up in your reports to help you identify which events are which. See the following example:

```
on(release){  
    cnetnetworks.event(1, "Your event description");  
}
```

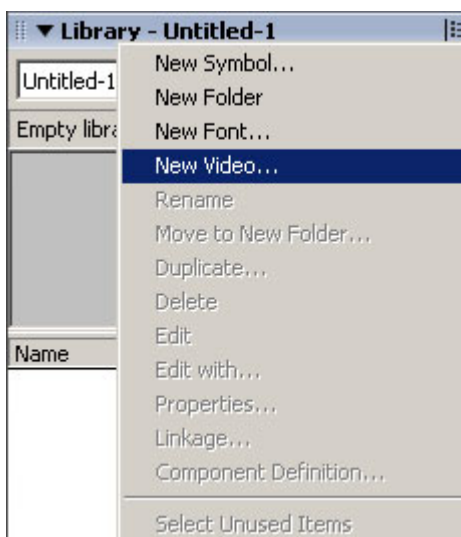
## 6. Implementing and Tracking a Flash Video (FLV)

You may track only one Flash Video. This limit applies to the creative as a whole, not each subload. These instructions assume you have an FLV file ready to use in your movie which adheres to CNET Networks' specifications (<http://www.cnetnetworks.com/advertise/specifications/>).

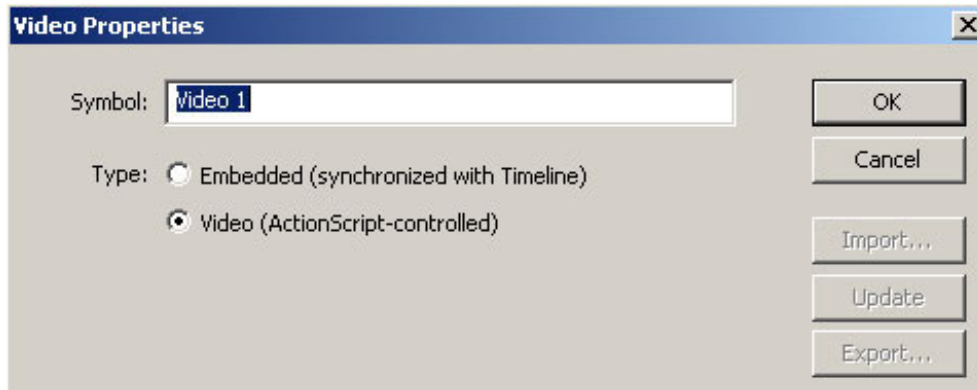
There are various ways to load FLVs into a Flash creative, but in order to be compatible with CNET Networks' Tracking, you must use Flash's NetStream Video object.

### Registering and Setup of a Flash Video Component

To set up tracking of a Flash Video using a NetStream Video Object, you must first create a new video object. To do so, first open the Library panel. Then, open the context menu at the top right of the panel. Select "New Video..." as shown below.



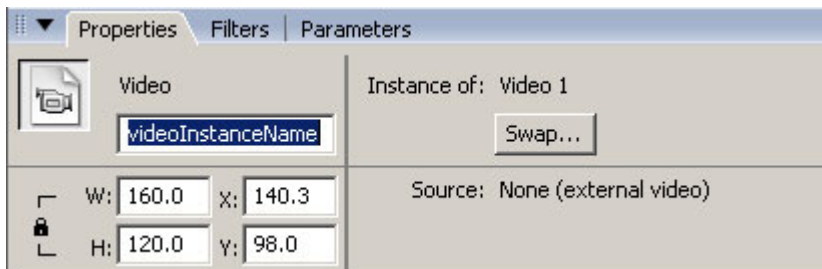
The symbol name you provide is not important, but make sure that “Video (ActionScript-controlled)” is selected:



Now the video object will be in the library of your movie. Drag the video object from your library onto the stage of your movie in the position where you would like the movie to play

**Please note:** The video object must exist on the stage in Keyframe 1 where you register your video. If you only want your video to appear later in the Flash movie, control the video’s visibility settings using the `_visible` parameter.

You may resize the video object to fit your movie (for best results, keep the aspect ratio of the video object the same as your FLV). Once the video is on the stage, select the video object and give it an instance name. Be sure to make note of the instance name as you will need it shortly.



Once you have an instance of the video object in your Flash creative, you’re ready to register the video using the CNET Networks Tracking Class. To register the video, you will need to supply the following:

- The instance name of your video object (as noted above)
- The filename of your FLV file (saved in the same directory as your main creative)
- The FLV’s duration in seconds
- A true/false value indicating if you would like the video to autoplay or not

With Frame 1 of the CNET TRACKING layer selected, on a line somewhere under “`new cnetnetworks ( ) ;`” use the following function to register your video:

```
cnetnetworks.registerVideo ( Video_instance_name, FLV_filename,
FLV_duration, Autoplay);
```

Replace "Video\_instance\_name" with the instance name you gave above. Replace "FLV\_filename" with the filename of your FLV. Replace FLV\_duration with the duration in seconds of your FLV. And replace Autoplay with a Boolean value of true or false. Your code should look like this:

```
cnetnetworks.registerVideo("VideoInstanceName",  
"FLV_filename.flv", 30.4, true);
```

Make sure to enclose the Video instance name and FLV filename in quotes. The duration of the FLV and the true/false autoplay value should **not** be in quotes.

If you chose your movie to autoplay, it will begin playing after the registerVideo function is called. Therefore, your video object must exist on the first frame of your creative timeline. By default, videos set to autoplay begin with the sound muted. Refer to the instructions below for information on muting and unmuting your video.

**NOTE:** Flash may take a second or two to buffer your video so playback may not begin immediately after registerVideo is called. See the advanced tip below for more information.

## Controlling Your Video

Once your video is registered, you have a variety of control functions available to use. The controls may be accessed from anywhere in your movie and may be attached to any buttons you create. The following is a list of the video controls and a description of each control:

- `cnetnetworks.videoPlay()`;  
The videoPlay function starts your video and is intended to be used as the action for a play button. Once the video starts playing, the "Video Middle" and "Video End" tracking actions will automatically be triggered as your video reaches its halfway and end points, respectively. The videoPlay function is designed to work when the video is not yet started, paused, or finished, so there are three possible tracking actions this function can invoke:
  - If your video is **not** set to autoplay and your video is starting for the first time, this function will start your video and count a "Video Start" tracking action.
    - **NOTE:** it may take a second or two to buffer your video so playback may not begin immediately. See the advanced tip below for more information.
  - If your video is paused, videoPlay will unpause your video and count a "Video Unpause" tracking action (see below).
  - If your video has completed playing, videoPlay will restart your video and count a "Video Replay" tracking action.

**Advanced tip:** The CNET Networks Tracking Class provides two callback functions to make synching events in your flash timeline with the video easier. The `cnVideoStarted` function is called when the Flash buffer is full and the video actually begins playback. The `cnVideoFinished` function is called when the video playback reaches the end of the video.

To execute your own code when the video ends, add a function in the root of your ActionScript called `cnVideoStarted()` or `cnVideoFinished()` and include any actions you like. When the buffer is full, `_root.cnVideoStarted()` will be executed. When the end of the video is reached `_root.cnVideoFinished()` will be executed. For instance, use this function to toggle a play/pause button back to play at the end of the video.

- `cnetnetworks.videoPause()`;  
The `videoPause` function pauses your video and counts a “Video Pause” tracking action. This function is intended to be used as the action for a pause button.
- `cnetnetworks.videoUnpause()`;  
The `videoUnpause` function resumes your video if it is paused and counts a “Video Unpause” tracking action.
- `cnetnetworks.videoReplay()`;  
The `videoReplay` function will restart your video from the beginning and count a “Video Replay” tracking action. This function will replay the video regardless of whether your video is currently playing, paused, or finished.
- `cnetnetworks.videoMute()`;  
The `videoMute` function will mute your video and count a “Video Mute” tracking action. This function will only mute your video; it will not affect other sounds in your movie.
- `cnetnetworks.videoUnmute()`;  
The `videoUnmute` function will restore sound to your video and count a “Video Unmute” tracking action. This function will only unmute your video; it will not affect other sounds in your movie.
- `cnetnetworks.videoFastforward()`;  
The `videoFastforward` function will advance your video by half a second each time it is called. It will also record a “Video Fastforward” tracking action.
- `cnetnetworks.videoRewind()`;  
The `videoRewind` function will rewind your video by half a second each time it is called. It will also record a “Video Rewind” tracking action.

Choose which video controls you would like to use and add each to a button in your movie. For instance:

```
on(release){  
    cnetnetworks.videoPlay();  
}
```

## 7. Implementing and Tracking Expand Ads

### Overview

To create and track an expand ad, you must create two Flash files:

- A Base Panel Flash file (a standard IAB-sized Flash ad)
- An Expand Panel Flash file

The Expand Panel is initially hidden from view and displays only upon user interaction. If the user initiates the “Collapse” action in the Expand Panel, the Expand Panel will hide itself.

There are two types of Expand actions:

- **Click-to-Expand** requires a user to click within the Base Panel to expand the Expand Panel.
- **Rollover-to-Expand** only requires the user to mouse over the Base Panel to activate the Expand Panel.

There are also two corresponding types of Collapse actions:

- **Click-to-Collapse** closes the Expand Panel when a user clicks.
- **Rolloff-to-Collapse** closes the Expand Panel when the user’s mouse cursor leaves the bounds of the Expand Panel.

To implement these types of actions, see the “Implementing Expand and Collapse Functions” section below.

### Setting up Keyframes in Your Expand Panel

The Expand Panel Flash file must follow these guidelines:

- The first Keyframe of your movie is reserved for your Tracking Class setup and registration code. It may also contain objects and video that will be used later in your movie, but should not contain any animation or other ActionScript.
- The second Keyframe is reserved as a ready state for the Expand Panel. This is the frame that the movie will return to while it is in its unexpanded, invisible state. This Keyframe should also not contain any animation or ActionScript code.
- The third Keyframe is the first frame of your Expand animation. This frame will be the first frame triggered each time your Expand Panel is activated. Start any animations on this frame and include any setup ActionScript needed for your own animations.

### Setting Up Tracking in Your Base and Expand Panels

To set up tracking for Expand ads, you must include an extra line of code before initializing the CNET Networks tracking class. The first Keyframe on Frame 1, Scene 1 of the CNET TRACKING layer should contain the following **two** lines of ActionScript:

```
System.security.allowDomain( "*" );  
new cnetnetworks( );
```

This modification allows your Base Panel and Expand Panel to communicate with each other. You must set up both Flash files in this way. Except for this additional line of code, Tracking should be set up as described in **Section 2** of this guide.

### Registering Movie Subloads, Exit Links, Events, and Video in Your Base and Expand Panels

Once you have finished initializing tracking in both your Expand and Base Panels, you can begin to start registering Exit Links, Movie Subloads, Events and a Flash Video as described in the previous sections of this guide.

**Please note:** You may only register a particular numbered Movie Subload, Exit Link or Event **once** across both your Expand and Base Panels. For example, it is **not** possible to have the line:

```
cnetnetworks.registerExit(1,  
"http://www.Your_Destination_URL.com", "_blank");
```

appear in both your Expand Panel and your Base Panel, regardless of if the destination URLs are the same or different. Each exit link number, event number or movie number may only appear once.

**Special notes on video:** You may only register one video and it must appear only in the Expand Panel. It must **not** be set up as an AutoPlay video and the videoPlay instruction should appear on or after the third Keyframe of your movie. The Video will automatically be stopped and reset should the user collapse your Expand Panel.

### Implementing Expand and Collapse Functions

To expand or collapse your Expand Panel, a function call must be added in the actions panel of any button or movie clip.

For a Click-to-Expand action, add a `cnetnetworks.expand("click");` function call to an `on(release)` event for a Button or MovieClip inside your Base Panel. For instance:

```
on(release){  
    cnetnetworks.expand("click");  
}
```

If you use a Click-to-Expand action, you must also include a Click-to-Collapse action. To do so, add a `cnetnetworks.collapse();` function call to an `on(release)` event for a Button or MovieClip inside your Expand Panel. For instance:

```
on(release){  
    cnetnetworks.collapse();  
}
```

For Rollover-to-Expand creatives, add a `cnetnetworks.expand("rollover");` function call to an `on(rollOver)` event for a Button or MovieClip inside your Base Panel. You may want your rollover Button or MovieClip to cover the entire Base Panel area to encourage the most users to expand your ad. See the following example:

```
on(rollOver){  
    cnetnetworks.expand("rollover ");  
}
```

**Please note:** CNET Networks automatically adds a 1 second delay to all Rollover-to-Expand creatives. If the user's mouse cursor leaves the boundaries of the Base Panel movie within 1 second of entering the rollover hotspot, the Expand Panel will not expand.

**Also note:** If you use a Rollover-to-Expand action, no additional code is needed to handle the corresponding Rolloff-to-Collapse action. This is handled and tracked automatically.

## 8. Publishing Your FLA Files

**Your Flash SWF must be published for Flash 8 Player. To publish your SWF for the Flash 8 Player, go to the “Publish Settings” in the “File” menu and go to the “Flash” tab.**

For information on CNET Networks’ restrictions on file size, frames per second, Flash version number, and dimensions, please refer to CNET Networks’ Ad Specifications (<http://www.cnetnetworks.com/advertise/specifications/>).

## 9. Testing Your Flash Creative

To test the creative’s Exit Links, Movie Subloads, and Flash Video with this Flash tracking code you can run the creative as you normally would using the “Test Movie” command within the Flash IDE. You can also test the Flash movie in a web browser if the files are published to a web server.

**Note:** Your Exit Links will not load in a browser until your Flash movie has been trafficked in the CNET Networks Ad Server.

For more information on the Flash Player 8 security, please see the Macromedia whitepaper at: [http://www.macromedia.com/devnet/flashplayer/articles/flash\\_player\\_8\\_security.pdf](http://www.macromedia.com/devnet/flashplayer/articles/flash_player_8_security.pdf)

## 10. Next Steps and Quick Links

If the exit links and the submovie loads work correctly, make sure your creative meets the ad unit’s specifications. If the exit links and the submovie loads do not work correctly, please review this documentation to re-implement the tracking functions.

Quick Links	
<b>CNET Networks Ad Specifications</b>	<a href="http://www.cnetnetworks.com/advertise/specifications/">http://www.cnetnetworks.com/advertise/specifications/</a>
<b>Questions About the Opportunity</b>	Contact your sales representative: <a href="http://www.cnetnetworks.com/advertise/contacts.html">http://www.cnetnetworks.com/advertise/contacts.html</a>
<b>Questions Regarding the Development of Your Creative</b>	Please send an email to <a href="mailto:advertiser@cnet.com">advertiser@cnet.com</a> .