SDRUŽENÍ
PRO INTERNETOVOU
REKLAMU

The Association for Internet Advertising (SPIR)

ILayer in single-and double swf files

This document describes the creation of ILayer types of ads in Flash (swf format), used under names such as Rollout, Hi-splash, I-Layer.

The basic type consists of the basic dimensions of the graphical element and the unfolded dimension that is displayed over the content of the webpage with time constraints and with the user option of removing/hiding it (by the close button). This extension of the banner is usually connected with the basic parts by one side.

Similarly, there are types and variations which use the overlapping of pages. They have common functionality, such as the close button - for example, separate banners floating above the page (floating ad, Sticky Ad, cosmic pilot).

All types are created in a similar way and functions are named in the same way. A linking (click) surface-button is created as if for a standard banner (see SPIR document “Creating a swf banner”).

When using native images (jpg,gif,png) it is the JavaScript utility script (supplied or most likely a part/characteristic of the ad system) which determines the way objects are behaving, displayed and placed.
When using swf, a call to JavaScript must be saved in individual files. Calls must be prepared before hand by the graphic designer/author of swf groundwork.

There are two ways to create swf ILayer type based on the basic type described above (a shared side or a portion of a side). Each has its advantages and disadvantages.

The first type is structurally similar to creating an ILayer of the native images. Creative documents are two swf files (hereafter referred to as starter and layer), which are installed/placed in their position using HTML and JavaScript, with JavaScript managing the display.
It's dual file layer.

The second type uses swf options more extensively and is formed by a single swf file containing both animations - for the basic "starter" , it is the part displayed at the assigned advertising space and animation for the expanded layer part . The overall dimensions of the swf animation are the same as the total / final expanded size of the banner. This solution is more logical, clear and relatively less dependent on the surrounding environment of the site (positioning the layer attribute) with all states and behavior encapsulated in swf .
Even so, it is necessary to correctly pass status changes to the host page, in order to reconfigure the characteristics of parent HTML container (typically a DIV) well.
We call it the single file layer.

Coding ILayer

Codes are written in Actionscript 2, a similar structure would be used for Actionscript3.

In the first frame of the main stage they all contain the general permission for external communication

```
if (! _root.domain1) domain1 = "*";
if (! _root.domain2) domain2 = "";
System.security.allowDomain (domain1, domain2);
```

To all areas which may be clicked on (not necessarily in all states of ILayer) or which lead to the target page, a button object with the following code to the mouse events should be added.

```
on (release) {
if (_root.clickthru==undefined && _root.clickTag) {
_root.clickthru = _root.clickTag;
}
if (_root.clickthru==undefined && _root.clickTAG) {
_root.clickthru = _root.clickTAG;
}
if ( _root.clickTarget ==undefined) {
_root.clickTarget = "_top";
}
getURL(_root.clickthru, _root.clickTarget);
}
```

all of the unfolded parts (layer animations) must contain a visible close button:

```
on (release) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_close" );
} else {
getURL( "javascript:" + _root.prefix + "_close()" , „_self“);
}
}
```

Codes in Two file layer

1. Starter swf file

In addition to the basic button, it also contains the following calls on the same button object :

```
on (rollover) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_starterIn" );
} else {
getURL( "javascript:" + _root.prefix + "_starterIn()","_self" );
}
}
on (rollout) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_starterOut" );
} else {
getURL( "javascript:" + _root.prefix + "_starterOut()","_self" );
}
}
```

The actual animation is like a common banner and so can be deployed as such (with capturing these events in the banner utility scripts that do not execute anything). If its required to display the layer part, for example when loading (initiating) a page with a banner, it is not dealt with in the swf codes, but in the JavaScript utility script where restrictive conditions may possibly be added (frequency capping, etc)

Two Layer swf file

May contain functions of a basic button. The following calls are used in the button object covering the whole surface of the layer:

```
on (rollout) {
if (_root.inout) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_layerOut" );
} else {
getURL ("javascript:" + _root.prefix + "_layerOut()","_self" );
}
}
}
on (rollover) {
if (_root.inout) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_layerIn" );
} else {
getURL ("javascript:" + _root.prefix + "_layerIn()","_self" );
}
}
}
```

In addition, there must be present a constantly visible button to close the animation, created as separate button above the layer. The closing function, XX_close (), can optionally be called automatically (after a certain time or at the end of the animation) without button. Using functions without a button (mouse events) must be treated so as to avoid the cyclical calls when playing (by defining the condition or action stop () at the end of the animation).

Codes in single-layer file

The animation is created as a one big file. The location of the animation in the site and it's visible dimensions are carried out by the utility JavaScript, which modifies styles and properties of the parent container of the swf file. In (folded) basic form (e.g. starter animation) only a slice/part is visible, when unfolded the whole swf becomes visible.

Creating a single-file animation is more challenging for placing button objects in layers,for logical restrictions of script calls, and of course, for navigating in the starter and layer animations. In the prepared animation, create the button object of a starter size (it may be part of a separate MovieClip for starter animation) and use the same codes just as for double-file starter including on(release) script for click handling. Moreover, the events will include jumps to unfold layer animation, for example:

```
on (rollover) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_starterIn" );
} else {
getURL( "javascript:" + _root.prefix + "_starterIn()","_self" );
}
_root.gotoAndPlay(„layerpart“);
}
on (rollout) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_starterOut" );
} else {
getURL( "javascript:" + _root.prefix + "_starterOut()","_self" );
}
}
```

If it's necessary to unfold the layer animation even when the html page is loaded (initiation), it can not be treated as the two layer and an automatic call (XX_starterIn ()) must be again placed into the SWF, preferably with some restrictive condition (a counter) or navigation structures in order to avoid the cyclical opening.
On The expanded part, or on the button, there are functions similar to the two file layer type:

```
on (rollOut) {
if (_root.inout) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_layerOut" );
} else {
getURL ("javascript:" + _root.prefix + "_layerOut()", "_self" );
}
}
}
on (rollOver) {
if (_root.inout) {
if ( _root.activex == 1 ) {
fscommand( _root.prefix + "_layerIn" );
} else {
getURL( "javascript:" + _root.prefix + "_layerIn()", "_self" );
}
}
}
```

Again with the present and visible close button and possibly an automatic call to close at the end of the animation. In each on(rollover, rollOout) call, before navigating the swf JavaScript calls can be placed.

Tips and advice:
1st Test, test and test again. Variables to the swf are transmitted via the GET parameter in the HTML object plug in - ....
starter.swf? prefix = 123 & ActiveX = 1 & clickthru = http://internet.billboard.cz .... Functions can be captured by simple structures <script> function 123_close () alert ("calling closed").....</script>
2nd Internal calls (navigational jumps) used on the buttons with a function call are to be inserted behind these JavaScript calls.
3rd Tests are to be performed over the HTTP protocol - on the local Web server, for reasons of security, some things work differently than when running the from the local file path on the hard disk.
4th Upon delivery attach the file which contains names for the functions used, dimensions and when using double-file layers, also about the place and direction of unfolding the layer part, if automatic function calls are used, document those as well and also possibly inform about times needed to play the layer animation .

Features Overview

	Duel-file layer banner	Single-layer banner
GIF, JPG alternative	Same size and coding behavior, alternative images in both dimensions.	Other, alternative is generally sufficient to create a starter in the base dimension.
Contained functions (except called ones)	Button in both dimensions, ILayer always start from the beginning, you can not synchronize animation between them (just initiate layer execution, automatically from the beginning), check stop (stop ()) at the end of the layer animation.	Button in all states. Navigation after animation for unfold and fold events. Loop playback for basic animation.
Calling js function	XXX_starterIn () and XXX_starterOut () for a basic starter banner XXX_layerOut () XXX_layerIn () XXX_close () for layer banner XXX_close () in addition on a separate CLOSE button. Target (Window) JS function always own "_self". XXX is replaced by variable prefix content	XXX_starterIn () XXX_starterOut () XXX_layerIn, XXX_layerOut () XXX_close i () located in one file at the assigned location and states of animation. Check to avoid cyclic function calls (in the infinite loop). Call to window "_self"
Benefits	The same procedure as for the classical image, possibility of free placement of the layer relative to the starter. Missing or defective function can be emulated by JavaScript with mouse captures (Same as image)	Simpler to design and debug in the development environment of Adobe Flash. Precise synchronization of follow up animations between starter and layer, Better debugging and testing
Disadvantages	More difficult to debug (in HTML page containing both swf elements) imprecise synchronization (on the level of execution from the beginning of the ILayer part, the exact time depends on displaying the element by JavaScript	Challenging the accuracy of coding and design in HTML (what is or is not transparent , functions in the right place, navigation in the animation in all states). Doesn't have to be accepted by all the advertising systems.