

AdOcean technical specification for preparing creatives

(international version)

I. GENERAL REQUIREMENTS

Guideline for the preparation of ad creatives:

1. The landing page URL cannot be longer than 2000 characters.
2. Maximal weight of creative file served by AdOcean emitter is 50 kB.
3. **The submitted advertisement cannot overload the processor of a standard computer by 25% or more.** A standard computer is assumed to have the following hardware and software components: Intel Celeron 1.7 GHz or AMD Duron 1.7 GHz processor with Flash Player 8 installed on the machine.
4. **SWF creatives can be made in any Flash version up to Flash 10, but AdOcean cannot assist in checking the proper implementation of required functions if ActionScript 3 is used. The creative agency is always responsible for the proper implementation of the functions specified in this document!** In case ActionScript 3 is used and verification by AdOcean Support is required, the creative agency has to send AdOcean either the FLA source or – more commonly – the appropriate parts of the ActionScript only.
5. It is not permitted to cover creatives containing transparent areas with invisible clickable buttons – only parts of the creative that are visible to the user may redirect to the advertiser's website (landing page) upon clicking.
6. It is forbidden to use scripts that change the browser window (its size, position, etc.). The only exceptions are creatives that assume actions of this type.
7. It is forbidden to use scripts to monitor user's interaction (e-tuning).
8. Ad creatives or creative codes causing errors or warnings during an advertisement's delivery will not be accepted.
9. **AdOcean does not engage in preparation of creative codes for advertisements.** We kindly request you to send us your custom creative codes with the set of materials.
10. If a creative utilizes **more than one clickTag ('multi-click' creative)**, those clickTags should follow the following naming convention (**pay special attention to the character case**):

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

Landing URL 1 -> **_root.clickTag1**

Landing URL 2 -> **_root.clickTag2**, etc

11. When creatives are not served by AO the technical specification of the given third-party ad server should be used. This happens usually **when communicators or instant messengers are used** (e.g. MSN Messenger) or if the planner of the campaign makes such decision.

IMPORTANT: third party ad servers have their own restrictions concerning such factors as, the size of a creative, number of frame rates per second and the functions that operate the creatives.

II. THE COMPLETE SET OF MATERIALS AND NAMING CONVENTIONS

By the complete set of materials we mean:

- ✓ The complete set of creatives fulfilling the specification requirements,
- ✓ URL address(es) of active landing page(s),
- ✓ Custom creative codes, if they are required.

Naming convention of files:

- ✓ It is recommended to use file names following this pattern: *creativeform_size_version*;
- ✓ You should only use lowercase letters of the English alphabet, underscores and numbers; it is recommended to avoid spaces in file names.

Landing page URLs:

- ✓ If more than one landing page is used in a campaign, you should always provide information about which URL a particular creative should be assigned to.

III. GENERAL REQUIREMENTS FOR CREATIVE FORMS

:: IMAGES ::

- ✓ Creatives must have the same size as defined in the media plan.
- ✓ Images must have the same file name as SWF files (differing only in file name extension) if they are meant to be used as alternative versions of Flash creatives.

:: FLASH CREATIVES ::

- ✓ Creatives must have the same size as defined in the media plan.
- ✓ If a creative is not transparent, please provide the hexadecimal colour code to be used for the background.
- ✓ Regardless of their form, all SWF creatives must include the **clickTag** parameter to enable the ad server to track clicks. Here is how to do it:
 1. Create a new layer **on top** of the prepared animation.
 2. Draw a rectangle on it then transform it into a **button type symbol**.
 3. Set this button to be **completely transparent**.

Actionscript 2

4. Assign the following function call to the button's **on (release)** event:

```
on (release)
{
    getURL(_root.clickTag, "_blank");
}
```

IMPORTANT: please pay attention to the character case in **_root.clickTag** variable!

Actionscript 3

4. Add the following piece of code to its first frame:

```
var clickTag:String=LoaderInfo(this.root.loaderInfo).parameters.clickTag;
target_button.addEventListener(MouseEvent.CLICK, targetUrlHandler);
function targetUrlHandler(mouseEvent:MouseEvent):void {
var interactiveObject:InteractiveObject = mouseEvent.target as
InteractiveObject;
    var li:LoaderInfo = LoaderInfo(interactiveObject.root.loaderInfo);
    var url:String = li.parameters.clickTag;
    if (url) {
        if (ExternalInterface.available) {
            try {
                var userAgent:String = ExternalInterface.call('function(){ return
navigator.userAgent; }');
                if (userAgent && userAgent.indexOf("MSIE") >= 0) {
                    ExternalInterface.call('window.open', url, '_blank');
                    return;
                }
            } catch (e:Error) {
                //
            }
        }
        navigateToURL(new URLRequest(url), '_blank');
    }
}
```

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

IV. REQUIREMENTS FOR ADVANCED CREATIVE FORMS

:: BRANDMARK ::

Number of required files: 1

- a) The creative needs a closing cross to make it disappear on user interaction with a function call attached to it, named **onCrossClick**.

The 'closing cross' is a rectangular button with a size of at least 14x14 pixels; it usually displays not only a cross but also the word 'Close'.

- b) The creative needs a button to minimize the advertisement to a small bar with a function call attached to it, named **minimize**.

In that bar shape the advertisement cannot block the web page outside the bar's dimensions. The size of the visible area in the minimized state is defined by numerical parameters in the ad template (in the ad server interface). It is forbidden to let the transparent area of the standard sized advertisement be clickable or to make it impossible to click on links in the web page content.

IMPORTANT: in this type of creative, the **button symbol cannot cover the bar's area**, clicking on it should not take users to the landing page!

- c) The creative needs a button to reset the advertisement to its standard size with a function call attached to it, named **maximize**.

The function is responsible for expanding the visible area of the standard size. This size is defined by numerical parameters in the ad template.

Actionscript 2

- a) closing cross:

```
on (release){
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.onCrossClick);
    }else{
        getURL("javascript:"+_root.onCrossClick+"()", "_self");
    }
}
```

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

b) minimize button:

```
on (release){
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.minimize);
    }else{
        getURL("javascript:"+_root.minimize+"()", "_self");
    }
}
```

c) maximize button:

```
on (release){
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.maximize);
    }else{
        getURL("javascript:"+_root.maximize+"()", "_self");
    }
}
```

Actionscript 3

```
var onCrossClick:String=LoaderInfo(this.root.loaderInfo).parameters.onCrossClick;
var minimize:String=LoaderInfo(this.root.loaderInfo).parameters.minimize;
var maximize:String=LoaderInfo(this.root.loaderInfo).parameters.maximize;

closeButton.addEventListener(MouseEvent.CLICK,closeAds);
function closeAds(e:MouseEvent):void{
    if(onCrossClick){
        ExternalInterface.call(onCrossClick);
    }
}

minButton.addEventListener(MouseEvent.CLICK,minimizeAds);
maxButton.addEventListener(MouseEvent.CLICK,maximizeAds);
function minimizeAds(e:MouseEvent):void{
    if(minimize){
        ExternalInterface.call(minimize);
    }
}
function maximizeAds(e:MouseEvent):void{
    if(maximize){
        ExternalInterface.call(maximize);
    }
}
```

AdOcean Ltd.

:: TOPLAYER (OVERLAY) ::

Number of required files: 1

The creative needs a closing cross to make it disappear on user interaction with a function call attached to it, named **onCrossClick**.

The 'closing cross' is a rectangular button with a size of at least 14x14 pixels; it usually displays not only a cross but also the word 'Close'.

Actionscript 2

```
on (release){
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.onCrossClick);
    }else{
        getURL(" javascript:"+_root.onCrossClick+" ();", "_self");
    }
}
```

Actionscript 3

```
var onCrossClick:String=LoaderInfo(this.root.loaderInfo).parameters.onCrossClick;

closeButton.addEventListener(MouseEvent.CLICK,closeAds);
function closeAds(e:MouseEvent):void{
    if(onCrossClick){
        ExternalInterface.call(onCrossClick);
    }
}
```

:: EXPANDING ::

Number of required files: 1

Your creative should have an active area over its whole surface and throughout the complete animation sequence (both in folded and unfolded state) with the following function calls attached:

Actionscript 2

```
on (rollOver){
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.doexpand);
    }else{
        getURL("javascript:"+_root.doexpand+"()", "_self");
    }
}

on (rollOut){
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.dolittle);
    }else{
        getURL("javascript:"+_root.dolittle+"()", "_self");
    }
}
```

Actionscript 3

```
var doexpand:String=LoaderInfo(this.root.loaderInfo).parameters.doexpand;
var dolittle:String=LoaderInfo(this.root.loaderInfo).parameters.dolittle;

target_button.addEventListener(MouseEvent.ROLL_OVER, mouseRollOver);
target_button.addEventListener(MouseEvent.ROLL_OUT, mouseRollOut);
function mouseRollOver(e:MouseEvent):void{
    if(doexpand){
        ExternalInterface.call(doexpand);
    }
}

function mouseRollOut(e:MouseEvent):void{
    if(dolittle){
        ExternalInterface.call(dolittle);
    }
}
```

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

The size of the visible area in folded and unfolded state is defined by numerical parameters in the ad template. When moving the mouse pointer over the advertisement it should unfold, after moving it off the advertisement's area the banner should fold back to the original state and size. Please be aware that some publishers only allow expansion (unfolding) on click event, the campaign managers can always provide such information if needed.

If the creative may expand only on first occasion automatically then the following or a similar solution could be used:

Actionscript 2

```
// AD_ID: a unique string that we associate with a given creative, e.g.
"ProductX_240-480x120_publisher1"

so = SharedObject.getLocal (AD_ID);
if (so.data.num_of_disp == 1) {
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.dolittle);
    }else{
        getURL("javascript:"+_root.dolittle+"();", "_self");
    }
} else {
    var so:SharedObject = SharedObject.getLocal(AD_ID);
    so.data.num_of_disp = 1;
    so.flush();
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.doexpand);
    }else{
        getURL("javascript:"+_root.doexpand+"();", "_self");
    }
}
```

The above solution uses Local Shared Objects in Flash. These objects provide functionality similar to cookies and make it possible to store certain variables on a user's computer. This way, information may be stored about a creative, whether it is being displayed for the first time or not. The above code's minimum requirements are Flash 6 and ActionScript 1.

To ensure that automatic expansion takes place once per day, follow the same procedure, i.e. use Local Shared Objects. However, in this case we will store the date when the creative has been delivered and match it next time with the current date ("*new Date()*" and "*getDate()*" functions may be used).

IMPORTANT: in case of expand skyscrapers you should pay attention to the fact that on a given web site, for which the creative has been prepared, it should unfold to the right or to the left, maybe to both directions. The Flash creative should be prepared accordingly (alignment of small size animation to the expanded one is crucial). The resizing of the DIV that contains the creative is done by the creative code in AO/gDE+ ad server, at the moment

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

when it receives the proper function calls (**dolittle** or **doexpand** if we talk about expanding forms) from the Flash banner.

IMPORTANT: expanding creatives *do not work* when placed inside an iframe by the publisher! Please consult with the publisher concerning the possible placement of the creative before the campaign starts.

:: PUSH & UNPUSH ::

Number of required files: 1

Push creatives have to be displayed in a minimized state when they get delivered on a web page. When fully downloaded, a SWF creative calls the **dopushlock** function. It automatically expands the banner's visible area (i.e. the DIV which contains the banner) to the size defined for maximized state. After 5 seconds the **dopushunlock** function is called, which starts to gradually roll up the banner's DIV container and reset it to its minimized state. After this process has been completed, the **dopushon** and **dopushoff** functions (which operate the same way as the ones described before) are unblocked; their call is initiated by user interactions (moving the mouse pointer over/off of a creative). Animation effects should be blocked/paused until the functions get executed and the DIV container changes its size.

a) The creative should have an active area on its whole surface throughout the complete animation sequence (in default and pushed down state, too) with the following function calls attached:

Actionscript 2

```
on (rollOver){
    if (flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.dopushon);
    }else{
        getURL("javascript:"+_root.dopushon+"()", "_self");
    }
}

on (rollOut){
    if (flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call(_root.dopushoff);
    }else{
        getURL("javascript:"+_root.dopushoff+"()", "_self");
    }
}
```

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

Actionscript 3

```
var dopushon:String=LoaderInfo(this.root.loaderInfo).parameters.dopushon;
var dopushoff:String=LoaderInfo(this.root.loaderInfo).parameters.dopushoff;

target_button.addEventListener(MouseEvent.ROLL_OVER, mouseRollOver);
target_button.addEventListener(MouseEvent.ROLL_OUT, mouseRollOut);
function mouseRollOver(e:MouseEvent):void{
    if(dopushon){
        ExternalInterface.call(dopushon);
    }
}
function mouseRollOut(e:MouseEvent):void{
    if(dopushoff){
        ExternalInterface.call(dopushoff);
    }
}
```

- b) When fully downloaded, the creative should call the following function:

Actionscript 2

```
if(flash.external.ExternalInterface.available){
    flash.external.ExternalInterface.call(_root.dopushlock);
}else{
    getURL("javascript:"+_root.dopushlock+"()", "_self");
}
```

Actionscript 3

```
var dopushlock:String=LoaderInfo(this.root.loaderInfo).parameters.dopushlock;

if(dopushlock){
    ExternalInterface.call(dopushlock);
}
```

- c) 5 seconds after starting its animation for the first time, the creative should call the following function:

Actionscript 2

```
if(flash.external.ExternalInterface.available){
    flash.external.ExternalInterface.call(_root.dopushunlock);
}else{
```

AdOcean Ltd.

```
getURL("javascript:"+_root.dopushunlock+"()", "_self");  
}
```

Actionscript 3

```
var dopushunlock:String=LoaderInfo(this.root.loaderInfo).parameters.dopushunlock;  
  
if(dopushunlock){  
    ExternalInterface.call(dopushunlock);  
}
```

IMPORTANT: unpush creatives start from expanded (unfolded) state, thus the **dopushlock** function call is not needed in that case!

IMPORTANT: push creatives *do not work* when placed inside an iframe by the publisher! Please consult the publisher about possible placement of the creative before the campaign starts.

:: PEEL AWAY ::

Number of required files: 1

This form of advertisement is using exactly the same logic and function like the expanding creatives. The creative should have an active area on its whole surface and – throughout the complete animation sequence (in folded and unfolded state, too) – those function calls have to be attached, which are presented in the specification of **Expanding** ads of this document.

The size of the visible area in folded and unfolded state is defined by numerical parameters in the ad template. When moving the mouse pointer over the advertisement it should unfold, after moving it off the advertisement's area the banner should fold to the original state.

:: CORNER ::

Number of required files: 1

The creative could have an active area on its whole surface and throughout the complete animation sequence (in folded and unfolded state, too) with functions connected by the names of **doexpand** and **dolittle**. The code for these functions calls is included in the specification of **Expanding** ads of this document.

The creative also needs a closing cross to make it disappear on user interaction with a function call attached to it, named **onCrossClick**. The code for this function call is included in the specification of **Toplayer** ads of this document.

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

The 'closing cross' is a rectangular button with a size of at least 14x14 pixels; it usually displays not only a cross but also the word 'Close'.

The size of the visible area in folded and unfolded state is defined by numerical parameters in the ad template.

:: FOOTER / FULL SCREEN OVERLAY / SCROLL ::

Number of required files: 1

The creative also needs a closing cross to make it disappear on user interaction with a function call attached to it, named **onCrossClick**. The code for this function call is included in the specification of **Toplayer** ads of this document.

If there is no closing cross on the creative, a standard one may also be added via the ad template itself (in the ad server interface).

:: DRAWBRIDGE ::

Number of required files: 3

This creative format consists of three SWF files: two skyscrapers and a wide toplayer. The display of this ad starts by showing the skyscrapers and after user interaction (typically moving the mouse pointer over the one of the skyscrapers) the toplayer is revealed.

The user interaction should have the **doexpand** function call attached to it, while the folding (collapse) of the skyscrapers should be realised by the **dolittle** function call. The code for these function calls is included in the specification of **Expanding** ads of this document.

The middle element of the ad is the toplayer. It needs a closing cross with the **onCrossClick** function call attached to it. The code for this function call is included in the specification of **Toplayer** ads of this document.

:: SIDEKICK ::

Number of required files: 2

This creative format consists of two SWF files: a rectangle or skyscraper and a toplayer covering half of the page area. The display this ad starts by showing only the smaller rectangle or skyscraper creative (placed usually on the right side of the page content). On user interaction (usually after user clicked a dedicated button in the smaller creative), half of the web page shifted horizontally and at the same time, the larger second creative moves into the view.

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

The user interaction should have the **doexpand** function call attached to it. The code for these function calls is included in the specification of **Expanding** ads of this document.

The larger (half page) creative needs a closing cross with the **onCrossClick** function call attached to it. The code for this function call is included in the specification of **Toplayer** ads of this document.

:: FOOTER + SIDEKICK ::

Number of required files: 2

This creative format consists of two SWF files: a footer and a toplayer covering half of the page area. The display this ad starts by showing only the footer. On user interaction (usually after user clicked a dedicated button in the smaller creative), half of the web page shifted horizontally and at the same time, the larger second creative moves into the view.

The user interaction should have the **doexpand** function call attached to it, while the folding (collapse) of the skyscrapers should be realized by the **dolittle** function call. The code for these function calls is included in the specification of **Expanding** ads of this document.

Both creatives need a closing cross with the **onCrossClick** function call attached to it. The code for this function call is included in the specification of **Toplayer** ads of this document.

:: POP-UP & POP-UNDER ::

Number of required files: 1

We advise that pop-up and pop-under creatives should be prepared in at least Flash 8 version. In earlier Flash versions the pop-up and pop-under creative forms do not work properly in Internet Explorer browsers because they activate the built-in pop-up blocker instead of simply opening the landing page in a new tab or window.

Inserting custom interactions

To insert custom interaction into flash creative, the following function call should be assigned to the event (button, frame):

Actionscript 2

Add the following piece of code to the event (button, frame) at which interaction should be count:

```
if(flash.external.ExternalInterface.available){
    flash.external.ExternalInterface.call("function_name");
}else{
    getURL("javascript:function_name();","_self");
}
```

where 'function_name' is a string, which needs to be set in the ad server interface as well in exactly the same form, in order to ensure interaction counting for statistics.

Actionscript 3

Add the following piece of code to the event (button, frame) at which interaction should be count:

```
ExternalInterface.call("function_name");
```

where 'function_name' is a string, which needs to be set in the ad server interface as well in exactly the same form, in order to ensure interaction counting for statistics.

Example: counting a custom interaction when clicking a button in the ad:

Actionscript 2

Add the following piece of code to the event (button, frame) at which interaction should be count:

```
on (release){
    if(flash.external.ExternalInterface.available){
        flash.external.ExternalInterface.call("function_name");
    }else{
        getURL("javascript:function_name();","_self");
    }
}
```

AdOcean Ltd.

7 Woloska Street, MARS Building, 2nd Floor, Warsaw, Poland, Tel. (0-22) 874 41 51 fax. (0-22) 874 41 01
Polish Court Register (KRS): 0000208080, Tax Identification Number (NIP): 522-27-34-030, Share capital: 770 000,00 PLN

Actionscript 3

Add the following piece of code to the layer with button at which interaction should be count:

```
on(realise){  
    button1.addEventListener(MouseEvent.CLICK,pressButton1);  
    function pressButton1(e:MouseEvent):void{  
        ExternalInterface.call("function_name");  
    }  
}
```

where 'button1' is the name of the button control, which should initiate the interaction counting on click event and 'function_name' is a string, which needs to be set in the ad server interface as well in exactly the same form, in order to ensure interaction counting for statistics.

WORTH KNOWING

How to check, whether the .swf file appropriately contains clickTag to count number of clicks?

Copy the URL of the given SWF file from the ad server into your browser's location bar, or if you do not have access to the ad server then just upload it for testing to any web server's public directory. Add the following string at the end without additional spaces: **?clickTag=** and an arbitrary URL that is different from the campaign's landing page URL, e.g. <http://www.gemius.com>. Then click on the creative. If the newly opened window loads the web page that you pasted before at the end of the URL address then everything works fine.

Last update: 22nd March, 2012