# openX advertisement handling

Work in progress: openX is now https://en.wikipedia.org/wiki/Revive_Adserver

Here you´ll find a description on how you best integrate openX into your Newscoop project.

## Accounts and Login for OpenX

If an OpenX instance is hosted with us (openx.sourcefabric.net) we usually use the SSO from login.sourcefabric.org for creating the account. The sysadmin will ask the client to create an account with login.sourcefabric.org and then ask for the Username and the Email-Address of that user to grant them access to the respective OpenX instance.

## Best Practice of OpenX settings for campaigns and zones

*This is short introduction on how openX works. Official product overview is here* http://www.openx.com/publisher/open-source-ad-server

OpenX is complete open-source solution for those who want to run ad server on their own infrastructure. One installation of OpenX can be used for serving ads to multiple websites. For every publisher  (website) we create **advertising zones**. Zone definition consists of Name, Description, Category, Zone type, Size.

**Zone size is fixed, meaning that it can accept only banners of that same size.** For example, if you define zone with dimensions 468x60px, you cannot use it to show banner of dimension 300x50px.That is especially important regarding ad formats like skyscrapers, that can have different banner sizes (regular skyscraper, wide skypscraper). If I define a zone for the "regular" skyscraper, the wide skyscraper will never appear in that zone.

Publishers have zones, Advertizers have campaigns, and campaigns have banners. One advertising client can have as many campaigns as they want, but the zones are limited to all possible positions.
Be aware that there may be a lot of zones, specified by sections and / or required banner formats. There is no "run of site" option in Newscoop, each zone has to be created in the templates.

In order to show some banner on the site, it is necessary to assign campaign banners to appropriate zones. If more banners from different campaigns are assigned to same zone, they will rotate according to the rules set to each banner or campaign **(campaign weight)** Probability tab shows percentage value for probability of showing for every banner.

Because different sets of options can be applied to zones (ie. type of banners - ordinary banner, button, rectangle, or video banners, or floating dhtml objectS), campaigns (for example, whether to show more banners from one campaign in different zones on one page - or not) and banners (delivery limitations - ranging from os and browser to day of the week and daytime), always double-check all these settings, especially if banner delivery doesn't behave as you expect. **As best practice we found that it is most secure - though more work at the beginning - to create a seperate campaign for each banner format.**

## Installing openX on your server

How to install the OpenX ad server

## Invocation of openX zones

The most straightforward way of embedding zones into web pages would be direct invocation for every zone. This simply places almost same javascript code into html page structure which then contacts openX instance of yours and asks for zone id to be loaded. Zone calculates which banner of all banners assigned to it would be shown, and finally banner appears on the page. This type of invocation is simplest but for big sites with lot of zones and big traffic it is not recommended, to say the least.

Smarter way of doing this is by website invocation. After selecting which site to use, there is the option to generate invocation per site. New page opens with series of short js codes. First one is for header, it looks somethnig like this:

```
 <!-- Generated by OpenX 2.8.10 -->
<script type='text/javascript' src='http://openx.zentralplus.ch/delivery/spcjs.php?id=1'></script>
```

The rest of code snippets is for every defined zone and these snippets should be placed in html structure on places where banners are planned.

To further improve this, and to speed up the process of page loading, for every page with banners we can define exactly which zones should be loaded (otherwise, invocation is done for all defined zones, sitewide, regardless of the page). It should be done like this:

```
<script type='text/javascript'>
var OA_zones = {'813':813,'814':814,'815':815,'825':825,'816':816,'817':817,'823':823,'818':818,'1236':
1236,'821':821,'819':819,'820':820,'822':822,'824':824,'1216':1216,'1234':1234,'826':826};
</script>
```

Obviously, in OA_zones set ids of zones to be shown on the page are listed. This piece of javascript code should go before header invocation.

## Adaptive design, device detection and device specific zones

Because OpenX don't support device and browser detection (yet!), and because zones have fixed sizes, we need to find workarounds to display appropriate ads to visitors on mobile devices (smartphones). We do this in Newscoop, and then just load openX zone according to detection results.

**For detection, we create new template and include it in _html-head.tpl like this**

```
{{ render file="_tpl/_ismobdevice.tpl" }}
```

**Then the template _ismobdevice.tpl looks like this:**

```
{{ if isset($smarty.request.mobile) || isset($smarty.request.tablet) || isset($smarty.request.phone) || $gimme-
>browser->browser_working == "webkit" && $gimme->browser->ua_type == 'mobile' }}
    {{ assign var=isMobDevice value=1 scope="global" }}
{{ else }}
    {{ assign var=isMobDevice value=0 scope="global" }}
{{ /if }}
```

On the position of the page where we want to include some banner, we check what kind of banner (actually, zone, because zone containes one or more banners) to show

```
{{ if !$isMobDevice }}
<script type='text/javascript'><!--// <![CDATA[
    /* [id3] Fullbanner Front */
    OA_show(3);
// ]]> --></script><noscript><a target='_blank' href='http://openx.zentralplus.ch/delivery/ck.php?
n=2c5c0aa'><img border='0' alt='' src='http://openx.zentralplus.ch/delivery/avw.php?zoneid=3&amp;n=2c5c0aa' /><
/a></noscript>
{{ else }}
<script type='text/javascript'><!--// <![CDATA[
    /* [id6] Tablet and mobile Front */
    OA_show(6);
// ]]> --></script><noscript><a target='_blank' href='http://openx.zentralplus.ch/delivery/ck.php?
n=642d667'><img border='0' alt='' src='http://openx.zentralplus.ch/delivery/avw.php?zoneid=6&amp;n=642d667' /><
/a></noscript>
{{ /if }}
```

However, all this can be further improved and simplified if we introduce asynchronous banner loading.

## Asynchronous loading of Advertisements - why and how?

**What is asynchronous loading?** In simple terms, this means that firstly the whole page is loaded and after this the ads are loaded. So this shall prevent newscoop pages from crashing.

Why **asynchronous loading?** The asynchronous loading of advertisement shall prevent the whole page from crashing, when the ad server is offline or not available.

**Important:** asynchronous loading is not possible with external ad snippets from third providers (google adsense or other ad networks).

**How does asynchronous loading work?** Asynchronous loading works by deferring the injection of external elements/code until all required elements are loaded. This usually involves writing javascript code to manage the deferral. The example below is a snippet of javascript code that when placed at the bottom of the page will wait for the document_ready state before attempting to load/inject any external ad elements. This code would replace any of the code from the examples above. The position in the page is important, and note that we still include the single page call (spcjs.php) script as described above, only in the footer instead of the header.

This code should be placed at the bottom of your page somewhere at the end of the footer:

**Asynchronous Loading**

```
<script type='text/javascript' src='http://openx.zentralplus.ch/delivery/spcjs.php?id=1'></script>
<!-- async replacement for openad spc calls -->
<script type='text/javascript'>
/**
 * this code will wait for document ready and then inject OpenX banners
 * into predefined ad divs defined with id='open-X' where X is the OpenX zoneId
 */
$document_ready = 0;
$(document).ready(function() {
    $document_ready = 1;
});
function displayOpenxAd($id) {
    if(!$document_ready) {
        setTimeout(function(){ displayOpenxAd($id); },1000);
        return false;
    }
    if(typeof(OA_output[$id])!='undefined' && OA_output[$id]!='') {
        var flashCode,
            oDIV = document.getElementById('openx-'+$id);
        if (oDIV) {
            // if it's a flash banner..
            if(OA_output[$id].indexOf("ox_swf.write")!=-1) {
                // extract javascript code
                var pre_code_wrap = "<script type='text/javascript'><!--// <![CDATA[",
                    post_code_wrap = "// ]]> -->";
                flashCode = OA_output[$id].substr(OA_output[$id].indexOf(pre_code_wrap)+pre_code_wrap.length);
                flashCode = flashCode.substr(0, flashCode.indexOf(post_code_wrap));
                // replace destination for the SWFObject
                flashCode = flashCode.replace(/ox\_swf\.write\(\'(.*)'\)/, "ox_swf.write('"+ oDIV.id +"')");
                flashCode = flashCode.replace(/document.write/, "$('#"+ oDIV.id +"').append");

                // insert SWFObject
                if( flashCode.indexOf("ox_swf.write")!=-1 ) {
                    eval(flashCode);
                }// else: the code was not as expected; don't show it

            }else{
                // normal image banner; just set the contents of the DIV
                oDIV.innerHTML = OA_output[$id];
            }
        }
    }
}
</script>
```

In order for the above code to work, we must modify how each ad zone is defined. The following example shows the modified injection code for the OpenX ad zoneId 2. This code would be placed anywhere you wanted this ad zone to appear in your site:

**Asynchronous Ad Loading**

```
<div id='openx-2' class="ad">
<script>
    $(document).ready(function() {
        displayOpenxAd(2);
    });
</script>
</div>
```

It is important to note that this code will not work for all OpenX ad types. For instance most external ads types (these are ads that inject their own code into your OpenX server, which in turn injects into your site) will not work with the asynchronous loading method described above. Also many 3rd party plugins that allow for extended ad types in OpenX may cause problems with this method of asynchronous loading, such as expanding banners and overlay ads. It should also be noted that there are other ways of implemented asynchronous loading (such as loading banner ads in iframes, or even using ajax to load ads), however each method has its own advantages and disadvantages. However, if your ad requirements dictate it, you can implement several methods of asynchronous loading on the same shit if needed.

## Caching and advertisement zones

After changes are made in OpenX it takes 1200 seconds till they become visible on the respective page.

## OpenX Plugins, Hacks, Extensions

There are several plugins, hacks and extensions for OpenX available. We do not recommend using them, as they are only valid till a security update by OpenX comes up and will stop working after that. Also, it is difficult to make some of them run on OpenX and they require a lot of testing.

## OpenX Enterprise solution

General Information on Enterprise solution.pdf
**The following Information provided by OpenX, February 2013**

### Our Enterprise platform provides:

- A dedicated admin and database server – fast user interface and separate data storage
- Enterprise cloud serving ad delivery for better performance
- SLA – including performance metrics and a 99.75% uptime
- Support - access our client services team by phone or email
- Straight-forward pricing that includes a large base volume of impressions/month and a discounted rate for any overages (All plans are volume based and include all creative types as standard
- Access to Enterprise plug-ins such as forecasting and competitive exclusion
- API access and custom development opportunity

**Our OpenX Enterprise 3.0 product starts at €500 a month for 12 months, with a start up fee of €1,000, which allows you to serve up to 10 million impressions a month.**
**Additional benefits of all versions of OpenX include:**
- OpenX Market - OpenX allows any publisher to sell directly to ad buyers through a real-time auction with no entry fee or technology integration required.
- OpenX Real-Time Selling™ - OpenX is the first company to allow any publisher to sell directly to major real-time bidders.
- OpenX seamlessly combines ad serving and ad network optimization across all revenue channels, including guaranteed, non-guaranteed, and unsold.
- OpenX is flexible. Extensible open source technology and flexible business terms make doing business with OpenX easy.

**Demo for interested clients available:**
It would be great to know how many ad impressions you currently have per month and then do a DEMO to show you how our platform works. Then we would give you a test account to try it out for a week and after that if you would be happy to go ahead with the ad server would be able to start the on boarding process.