Webmaster Guide

February 2008

Notice

CONFIDENTIAL

Table of Contents

Webmaster Guide

This guide deals with a number of topics related to configuring a website to serve the various ad formats.

The topics include:

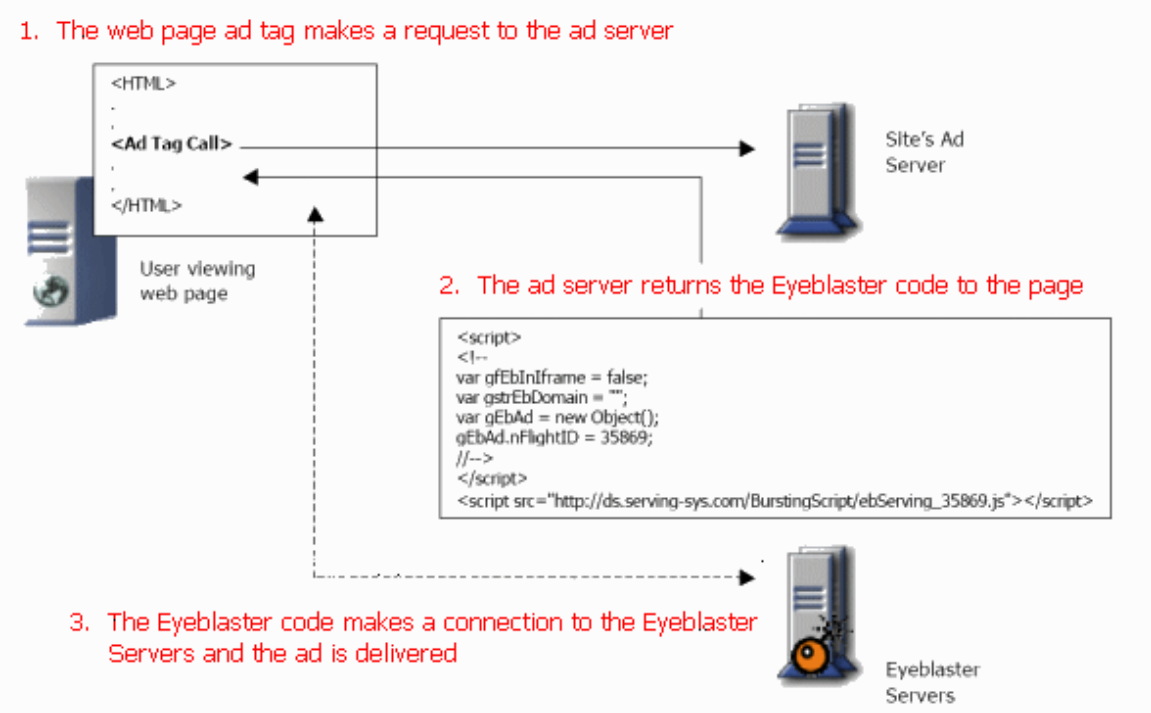
- How ads are served
- Handling Code
- Handling Code for In-Stream Video Ads
- Serving Eyeblaster Code
- Trafficking Considerations
- Ad Serving Limitations
- Tracking on the Publisher Ad Server
- Using Special JavaScript Variables
- Special Tagging Instructions for Certain Ad Formats
- How Ads Interact with Objects on a Web Page

Notes:

- *All code must be used exactly as it appears in the Eyeblaster code generator. Therefore it is advisable to always generate the code in the Eyeblaster Platform. This will ensure that the code appears in the correct format. If needed, the code can also be copied from this document: To copy from a PDF document, click on the Text button (next to the Magnifying Glass button), within the Adobe Acrobat Reader.*
- *All information within corresponds to the supported browsers and platforms only.*
- *The Eyeblaster code uses non-secure standard http protocol. If using the code within a secure environment (for example SSL), the user will be prompted that a non-secure connection is being made.*

How Ads Are Served

To display an ad, a few lines of code (referred to as "Eyeblaster Code") are served to the web page, via the website's ad server. Once the Eyeblaster Code is served to the page being viewed, a connection is made from the user's browser to the Eyeblaster servers. This process is illustrated in the following diagram:



Out-of-Banner Ads

Once a connection is made to the Eyeblaster servers, a "polite download" process starts. This means that Eyeblaster waits until the page is fully loaded before the ad itself starts loading in the background. Once the page is fully loaded, the ad is downloaded and displayed. An impression is counted when the ad is displayed.

If, while the ad is being downloaded, a user jumps to a new page, the download immediately stops, and continues from where it has stopped only after the next page has been fully loaded (and only if that page has also been served the Eyeblaster Code).

The result is a rich ad displayed to users, without affecting the site's speed and performance.

***Note**: It is possible to override the "polite download" setting, and allow the ad to load along with the rest of the page, in order to speed up the load of the ad, and thus increase the available inventory for Eyeblaster ads.*

Banner Ads

Once a connection is made to the Eyeblaster servers, a default gif or a default Flash file is downloaded immediately to the page. This file is not politely downloaded, but is downloaded with the page and will adhere to the site's specifications for banner ads. After the page finishes downloading, the Eyeblaster servers will be called to download the Rich Flash file for polite banners or any panels for expandable banners. An impression is counted when the default gif or Flash is served to the page. This means that included in the number of impressions served are impressions that did not present the full rich media experience to the user.

***Note**: Eyeblaster Code should be served within the <BODY> section of the HTML page, and not within the <HEAD> section.*

Handling Code

All code must be used exactly as it appears in the Eyeblaster code generator. Therefore it is advisable to always generate the code in the Eyeblaster interface. This will ensure the code appears in the correct format. If needed, the code can also be copied from the table below:

Flight Type	JavaScript Tag Code	IFRAME Tag Code

Out-of-Banner
(Can have one or more of the following ad formats: floating ads, floating ads with banner reminders, floating ads with floating reminders, commercial breaks, window ads and wallpaper ads).
	<script>
	<!--
	var gfEbInIframe = false;
	var gEbAd = new Object();
	gEbAd.nFlightID = 68086;
	//-->
	</script>
	<script src="http://ds.serving-sys.com/BurstingScript/ebServing_68086.js"></script>
		<script>
		<!--
		var gfEbInIframe = true;
		var gEbAd = new Object();
		gEbAd.nFlightID = 68086;
		//-->
		</script>
		<script src="http://ds.serving-sys.com/BurstingScript/ebServing_68086.js"></script>

Rich Media Banner
(Can contain either polite banners or expandable banners).
	<script>
	<!--
	var gfEbInIframe = false;
	var gEbBAd = new Object();
	gEbBAd.nFlightID = 50072;
	gEbBAd.nWidth = 468;
	gEbBAd.nHeight = 60;
	var gstrEbRandnum = new String(Math.random());
	gstrEbRandnum = gstrEbRandnum.substring(gstrEbRandnum.indexOf(".")+1 , gstrEbRandnum.length);
	gEbBAd.strNonSupported = "http://bs.serving-sys.com/BurstingPipe/NonSupportedBanner.asp?FlightID=50072&Page=&PluID=0&Width=468&Height=60&Pos=" + gstrEbRandnum;
	//-->
	</script>
	<script src="http://ds.serving-sys.com/BurstingScript/ebBannerServing_50072.js"></script>
	<noscript>
	<img src='http://bs.serving-sys.com/BurstingPipe/BannerSource.asp?FlightID=50072&Page=&PluID=0&Pos=7515' border=0 width=468 height=60'>
	</noscript>
		<script>
		<!--
		var gfEbInIframe = true;
		var gEbBAd = new Object();
		gEbBAd.nFlightID = 50072;
		gEbBAd.nWidth = 468;
		gEbBAd.nHeight = 60;
		var gstrEbRandnum = new String(Math.random());
		gstrEbRandnum = gstrEbRandnum.substring(gstrEbRandnum.indexOf(".")+1 , gstrEbRandnum.length);
		gEbBAd.strNonSupported = "http://bs.serving-sys.com/BurstingPipe/NonSupportedBanner.asp?FlightID=50072&Page=&PluID=0&Width=468&Height=60&Pos=" + gstrEbRandnum;
		//-->
		</script>
		<script src="http://ds.serving-sys.com/BurstingScript/ebBannerServing_50072.js"></script>
		<noscript>
		<img src='http://bs.serving-sys.com/BurstingPipe/BannerSource.asp?FlightID=50072&Page=&PluID=0&Pos=8770' border=0 width=468 height=60'>
		</noscript>

Standard Banner
(Contain low size (up to 30K) .gif, .jpg or Flash ads only).
	<script>
	<!--
	var gEbStdBanner = new Object();
	gEbStdBanner.nFlightID = 77339;
	gEbStdBanner.nWidth = 160;
	gEbStdBanner.nHeight = 600;
	var gstrEbRandnum = new String(Math.random());
	gstrEbRandnum = gstrEbRandnum.substring(gstrEbRandnum.indexOf(".")+1 , gstrEbRandnum.length);
	gEbStdBanner.strNonSupported = "http://bs.serving-sys.com/BurstingPipe/NonSupportedBanner.asp?FlightID=77339&Page=&PluID=0&Width=160&Height=600&Pos=" + gstrEbRandnum;
	//-->
	</script>
	<script src="http://ds.serving-sys.com/BurstingScript/ebStdBanner_77339.js"></script>
	<noscript>
	<img src='http://bs.serving-sys.com/BurstingPipe/BannerSource.asp?FlightID=77339&Page=&PluID=0&Pos=2173' border=0 width=160 height=600'>
	</noscript>
		<script>
		<!--
		var gEbStdBanner = new Object();
		gEbStdBanner.nFlightID = 77339;
		gEbStdBanner.nWidth = 160;
		gEbStdBanner.nHeight = 600;
		var gstrEbRandnum = new String(Math.random());
		gstrEbRandnum = gstrEbRandnum.substring(gstrEbRandnum.indexOf(".")+1 , gstrEbRandnum.length);
		gEbStdBanner.strNonSupported = "http://bs.serving-sys.com/BurstingPipe/NonSupportedBanner.asp?FlightID=77339&Page=&PluID=0&Width=160&Height=600&Pos=" + gstrEbRandnum;
		//-->
		</script>
		<script src="http://ds.serving-sys.com/BurstingScript/ebStdBanner_77339.js"></script>
		<noscript>
		<img src='http://bs.serving-sys.com/BurstingPipe/BannerSource.asp?FlightID=77339&Page=&PluID=0&Pos=1809' border=0 width=160 height=600'>
		</noscript>

Video Clip
(Up to 3 VideoClip ads, each up to 30 seconds in length. Each ad can include up to 1MB of 300Kbps video).
	<script>
	<!--
	ebSetVideoFlightID(48528);
	ebTagsEnd();
	//-->
	</script>
		<script>
		<!--
		ebSetVideoFlightID(48528);
		ebTagsEnd();
		//-->
		</script>

Legend

Flight Type	JavaScript Tag Code
`var gfEbInIframe`	Variable for determining if the ad is served inside an

	iframe or not
`var gstrEbDomain = "";`	Variable for determining the cookie to be used when serving to multiple sub-domains
`var gEbAd = new Object();` `gEbAd.nFlightID = 34440;`	<Variable that sets the Flight ID

`var gEbBanner = new Object();`
`gEbBanner.nFlightID = 35467;`
`gEbBanner.nWidth = 180;`
`gEbBanner.nHeight = 150;`
	Variable that sets the Flight ID, height and width of banner ads

`gEbBanner.strNonSupported = "<a href='http://bs.servingsys.com/BurstingPipe/BannerRedirect.asp?FlightID=35467&Page=&PluID=0&Pos=4023' target='_blank'><img src='http://bs.servingsys.com/BurstingPipe/BannerSource.asp?FlightID=35467&Page=&PluID=0&Pos=4023' border=0 width=180 height=150></a>";`	Variable that sets the call to the Eyeblaster servers for a default image when the OS and browser combination do not meet the Eyeblaster system requirements
`<script src=http://ds.servingsys.com/BurstingScript/ebServing_35467.js>`	Script tag that calls the Eyeblaster servers for an ad
`<img src='http://bs.servingsys.com/BurstingPipe/BannerSource.asp?FlightID=35467&Page=&PluID=0&Pos=4023' border=0 width=180 height=150>`	Image tag that calls the Eyeblaster servers for gif ads when the browser cannot read JavaScript
`<a href='http://bs.servingsys.com/BurstingPipe/BannerRedirect.asp?FlightID=35467&Page=&PluID=0&Pos=4023' target='_blank'>`	Click tag that calls the Eyeblaster servers for the clickthrough URL of an ad

`var gEbStdBanner = new Object();`
`gEbStdBanner.nFlightID = 35560;`
`gEbStdBanner.nWidth = 468;`
`gEbStdBanner.nHeight = 60;`
	Variable that sets the Flight ID, height and width of standard banner ads

Handling Code for In-Stream Video Ads

Different code is generated for the In-Stream ad format depending on the method used by the publisher to play the video. The method can be a WMP playlist, the Brightcove system, or another method for which Eyeblaster's generic solution will be used.

Notes

- *To work with In-Stream, your account manager must enable the option for your account.*
- *The In-Stream format must be selected as the ad format when defining the ad and when defining the flight.*

The publisher should handle the code according to the method used to play the In-Stream video:

- **WMP Playlist:** The generated code is placed in the in the playlist (ASX) file in the entry where the ad is to be played.

  Note: *Only the WMV format is supported for WMP player.*

- The following example shows the ASX code with the generated code (in blue). The code in green indicates the content entry.

An example of how the ASX should look after adding the generated code:

```
<ASX version = "3.0">
      <!--A simple playlist with entries to be played in sequence.-->
            <ENTRYREF HREF="http://bs.serving-sys-int.com/BurstingPipe/adServer.bs?cn=is&c=23&pli=222925&pl=wmp&pi=0&pos=9706&ord=[timestamp]" />
      <Entry>
            <Ref href = "http://ds.serving-sys-int.com/BurstingRes/Site-2/Type-13/78789FD2-BA28-4C1F-9031-E99053F1FBBA.wmv"/>
      </Entry>
</ASX>
```

Notes:

- The tag requires the publisher to implement cache busting using the “timestamp” variable. Verify that your publisher’s player can identify the “timestamp” variable. If not, replace the variable with a variable recognizable to the ad server. This should be done on the account level in the system. The default variable, if not defined, is “timestamp”.
- To use the WMP solution, publishers should use the above format (<ENTRYREF HREF …/>), and not the entry node:

```
<Entry>
  <Ref href …/>
<Entry/>.
```

- **Brightcove:** The generated code for Brightcove supported ads includes XML code for In-Stream flights that is accepted by the Brightcove player.

  Note: *Only the FLV format is supported for Brightcove.*

  An example of code generated for Brightcove:

```
<videoAd trackPointTime="" version="1" trackPointURLs="" duration="30"
trackStartURLs="http://bs.serving-sys.com/BurstingPipe/adServer.bs?cn=isi&amp;interactionsStr=~UnSpecified~0^ebVideoStarted~0~0~1~0~1~0~0&amp;pos=3477&amp;ebRandom=[timestamp]&amp;cim=1" trackMidURLs="http://bs.serving-sys.com/BurstingPipe/adServer.bs?cn=isi&amp;interactionsStr=~UnSpecified~0^eb50Per_Played~0~0~1~0~1~0~0&amp;pos=3477&amp;ebRandom=[timestamp]"
trackEndURLs="http://bs.serving-sys.com/BurstingPipe/adServer.bs?cn=isi&amp;interactionsStr=~UnSpecified~0^ebVideoFullPlay~0~0~1~0~1~0~0&amp;pos=3477&amp;ebRandom=[timestamp]">
<videoURL>http://bs.serving-sys.com/BurstingPipe/adServer.bs?cn=is&amp;c=23&amp;pli=258272&amp;pi=0&amp;vt=FLV8&amp;pos=3477&amp;ord=[timestamp]&amp;cim=1
</videoURL>
<videoClickURL>http://bs.serving-sys.com/BurstingPipe/adServer.bs?cn=bisi&amp;pli=258272&amp;pi=0&amp;p=UnSpecified&amp;pos=3477
</videoClickURL>
</videoAd>
```

- **Custom Player:** This code should be generated for all publishers using methods that are not WMP playlist or Brightcove.

  The generated code includes links (URLs) that are sent to the ad servers when implemented by the publisher. (It is up to the publisher to implement the links. The tracking is done by the publisher's player.) One link is used to serve the ad and the additional links are triggered for interactions. For example, a Video Mute interaction is performed, and if the player knows how to record the interaction, then a URL specifically for the Video Mute interaction is recorded (sent to the ad server).

  An example of generated code:

  //Note: It is mandatory to report impressions if you want to report any other additional interactions.

  //To track impressions, use the following URL:

  //To call the asset use the following URL (not included for In-Stream Video Tracking flights):

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=is&c=23&pli=[FligthID]&pi=[PLUID]&vt=[VideoFormat]&pos=[RandomNumber]&ord=[timestamp]
  ```

  //For click through URL and tracking use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/BannerRedirect.asp?FlightID=[FlightID]&Page=[PageName]&PluID=[PLUID]&pos=[RandomNumber]
  ```

  //To track Video Started use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebVideoStarted~0~0~1~0~1~0~0&ebRandom=[timestamp]
  ```

  //To track Video 25% Played use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^eb25Per_Played~0~0~1~0~1~0~0&ebRandom=[timestamp]
  ```

  //To track Video 50% Played use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^eb50Per_Played~0~0~1~0~1~0~0&ebRandom=[timestamp]
  ```

  //To track Video 75% Played use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^eb75Per_Played~0~0~1~0~1~0~0&ebRandom=[timestamp]
  ```

  //To track Video Ended use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebVideoFullPlay~0~0~1~0~1~0~0&ebRandom=[timestamp]
  ```

  //To track Video Mute use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebVideoMute~0~0~1~1~1~0~0&ebRandom=[timestamp]
  ```

  //To track Video Unmuted use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebVideoUnmuted~0~0~1~1~1~0~0&ebRandom=[timestamp]
  ```

  //To track Video Paused use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebVideoPause~0~0~1~1~1~0~0&ebRandom=[timestamp]
  ```

  //To track Video Replay use the following URL:

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebVideoReplay~0~0~1~0~1~0~0&ebRandom=[timestamp]
  ```

  //to track Video Duration use the following URL:

  Note: *It is up to the publisher to replace the [Duration] token with the actual playing duration.*

  ```
  http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebVideoPlayDuration~[Duration]~0~1~0~1~0~0&ebRandom=[timestamp]
  ```

//to track Video Full Screen Start use the following URL:

```
http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebFSStart~0~0~1~1~1~0~0&ebRandom=[timestamp]
```

//To track Video Full Screen End use the following URL:

```
http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebFSEnd~0~0~1~1~1~0~0&ebRandom=[timestamp]
```

//To track Video Full Screen Mute use the following URL:

```
http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebFSVideoMute~0~0~1~1~1~0~0&ebRandom=[timestamp]
```

//To track Video Full Screen Unmuted use the following URL:

```
http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebFSVideoUnmuted~0~0~1~1~1~0~0&ebRandom=[timestamp]
```

//To track Video Full Screen Paused use the following URL:

```
http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebFSVideoPause~0~0~1~1~1~0~0&ebRandom=[timestamp]
```

//to track Video Full Screen Duration use the following URL:

Note: *It is up to the publisher to replace the [Duration] token with the actual playing duration*

```
http://[BurstingServerPath]/BurstingPipe/adServer.bs?cn=isi&interactionsStr=~[PageName]~[PLUID]^ebFSVideoPlayDuration~[Duration]~0~1~0~1~0~0&ebRandom=[timestamp]
```

Note: *The tag requires the publisher to implement cache busting using the "timestamp" variable. Verify that your publisher's player can identify the "timestamp" variable. If not, replace the variable with a variable recognizable to the ad server. This should be done on the account level in the system. The default variable, if not defined, is "timestamp".*

Serving Eyeblaster Code

There are three common methods for serving rich-media through an ad server:

Through an IFRAME ad tag

When implementing Eyeblaster through an IFRAME ad tag, an additional one-time setup is required. An HTML file (called "addineyeV2.html") needs to be saved within a sub-directory called "eyeblaster" within the site's domain (i.e., off of the root directory of the site's web server).

For example: if the site is www.yahoo.com, the file will be located at: http://www.yahoo.com/eyeblaster/addineyeV2.html.

This step needs to be done only once, and not on a per-campaign basis. Note, that if the site has multiple sub-domains, served from a number of web servers, the file should be placed separately for each sub-domain.

If needed, it is possible to place the additional file in a different directory, and to use a different file name as well. In this case, an additional script variable is required, to be served along with the rest of the Eyeblaster Code. This additional tag contains the relative path of the file:

For example: if the file is to be located in www.domainname.com/ads/eyeblaster/adcode.html, the tag will be:

```
var gstrEbIframeLocation = "ads/eyeblaster/adcode.html";
```

To generate this variable, expand the **Advanced Settings** section within the **Generate Code** tab and place the path in the **IFRAME File Location** field.

In some cases, sites use a "double IFRAME" configuration, where a call is made to the ad server, and instead of serving the Eyeblaster Code directly, another IFRAME tag is served, which in turn makes another call, that serves the actual code.

In such a case, an additional script variable should be used, in addition to the tags already served, as follows:

```
Var gstrEbDisplayPos = "top";
```

Through a JavaScript ad tag

When the ad tag on the page is a JavaScript tag, generate the code directly through the Eyeblaster interface selecting **Script** as the tag type. Do not remove from, add to or modify the tags.

Using a server-side include

To serve Eyeblaster ads using a server-side include (i.e., when the ad call is made by the site's web server before the page loads, rather than by the user's browser), follow the same instructions used in serving the code through a JavaScript ad call.

Remember that an Eyeblaster ad can be served to the page only after the ad has been approved by the publisher. To allow publishers to test an Eyeblaster ad before approving it, select the **Test** mode when generating the Eyeblaster Code. The **Test** option appears in the dropdown menu called **Tags Generated For** within the **Generate Code** tab.

Once the code is served with the additional Test tag, up to 1,000 test impressions can be served, and these are counted as ordinary impressions within the campaign, i.e., they are not distinguishable from other impressions for the same ad. If a different limit of test impressions is needed, contact Eyeblaster.

Note: *If you would like to test an ad after it's been approved, but before it goes live, you can do this without test tags. There is no need to change the start date of the flight. You have up to 1000 impressions that can serve prior to the start date.*

Trafficking Considerations

Out-of-Banner ads are served in a similar way to Standard Banner ads in terms of targeting and scheduling. The publisher typically uses his/her server to set up the ad targeting and to book the ads. However there are a few properties belonging to Out-of-Banner which need to be dealt with differently as described below.

Out-of-Banner Ads

Impressions vs. Ad-Plays

Due to the polite download process, a user may browse many times through a number of Eyeblaster-enabled pages before seeing the actual ad. In other words, the Eyeblaster Code will be served a number of times to that user, while the actual ad will be seen and counted only once the ad has been fully loaded. Therefore, when setting up the ad within the ad server, it is important to pay attention to the following:

- The "impressions" logged on the publisher's ad server show the number of times the Eyeblaster Code has been served, not the number of actual ad-plays.
- The actual number of ad-plays can be accessed through the publisher's Eyeblaster account, where detailed reports on ad-plays, clicks, and other data are provided. Alternatively, actual ad-plays can be tracked by the site's ad server, by configuring impression Tracking URL's and additional variables in the Eyeblaster interface.
- The ratio between code-impressions and ad-plays is on average 2:1. This means that in order to serve 100,000 ad-plays, a site must book approximately 200,000 impressions in the site's ad server for the duration of the campaign. In practice, it is advisable to use a ratio of 2:1, and after a few days into the campaign, measure the actual ratio (using the Eyeblaster serving-statistics report in the Eyeblaster interface), and use that ratio for booking the rest of the campaign.

In the case above, 200,000 should be the number of impressions reserved for the campaign when the campaign is being planned and 100,000 should be the number of plays guaranteed to the agency. To ensure that the campaign delivers smoothly, inventory must be confirmed for 200,000 in the site's ad server.

Notes:

- *It is possible to improve the impression-to-play ratio, by implementing the "instant download /instant play" option, which overrides the default "polite download". See the section Instant Download and Play section for details on how to implement this option.*
- *Additional requirements, such as Frequency Capping, will substantially lower the ratio between code impressions and ad-plays. For example, if the ad is capped at "1 play per unique user", the*

available inventory is determined by the number of unique users accessible to the ad, and not the number of page views. To minimize the ratio when using frequency caps, set a slightly less restrictive frequency cap inside the site's ad server.

- *When using DFP5 Interstitial Format, the number of impressions booked within DART should be equal to the number of ad plays needed.*

Separate Ad Tags vs. Piggybacking

It is recommended that the publisher set up dedicated ad tags on the site pages for the serving of rich media ads. By doing so, maximum flexibility is achieved in scheduling the ad. However, this may sometimes require modifying the site web pages (to add a tag that did not exist there before). Therefore, another method to serve the ad is to piggyback the Eyeblaster Code onto an existing banner space, serving the banner code and the Eyeblaster code simultaneously.

The disadvantage of this type of deployment is that the trafficking of the Eyeblaster ad is always linked to the trafficking of a particular banner space and thus linked to the number of impressions the banner is set to serve (Note that this restriction is resolved when using DFP5 Interstitial Format).

However, there are 2 notable advantages to piggybacking the Eyeblaster Code onto an existing banner:

- There will be no additional calls made to the site's ad server due to serving of Eyeblaster ads, thereby lowering costs for the publisher.
- At times it is part of the campaign's specifications that a rich-media ad appears only on pageswhere a specific banner appears, therefore making piggybacking a natural choice.

Instant Download and Play

In order to allow faster download and display of Out-of-Banner ads, it is possible to override the default "polite download", allowing the ad to load in parallel to the page content, and even to display before the page has been fully loaded. The following two options define the different modes of operation available:

- **Polite Download / Polite Play**: This is the default setting, where the ad is downloaded and played only after the page has been fully loaded, as described earlier.
- **Instant Download / Instant Play**: With this setting, the ad is downloaded together with the page, and plays immediately after download is completed, even if the page has not yet fully loaded. This is the quickest way to display the ad.

There are 2 main advantages of using the instant download and play:

- The impression to play ratio decreases. This means here will be less code impressions needed for each ad play. This can increase the site's available inventory dramatically.
- Ads will appear sooner on the site page, before a user starts reading content. In many cases it is advantageous to display the ad as soon as possible, so that the user is not interrupted while reading content and articles. Using the instant play option will display the ad as soon as possible, improving the overall user experience.

***Note**: In the case of the intrommercial, the Instant Download / Instant Play setting is always used.*

***Note** In cases of network latency, using instant download may have an adverse effect on the load time of the page. In cases where extreme slowing of the page occurs, switch back to polite mode.*

Implementing instant download and play is done via an additional JavaScript variable, which can be generated using the Eyeblaster Code Generator. It has the following format:

```
gEbAd.nDownloadMode = Mode;
```

Where:

Mode defines which of the above modes is being used, where:

- Mode=1 corresponds to Polite Download / Polite Play
- Mode=3 corresponds to Instant Download / Instant Play
- (Mode=2 is obsolete)

As in other code options, it is advisable to always use the Eyeblaster Code Generator to generate the correct value of the Mode parameter.

Banner Ads

Banner ads should be served the same way 3rd party ad server tags are served. These tags should be booked in the site's ad server with the same impression goal and date range as is listed in the Eyeblaster interface.

Note that Eyeblaster banners will serve an ad every time the site's ad server serves the code on the page, regardless of the dates and flight volume set i. The dates and impressions in the site's ad server determine when and how many impressions are served. This means if the end date in the site's ad server is past the end date in the Eyeblaster ad server, Eyeblaster will continue to serve the ad up to the ad server end date because the Eyeblaster code was being served to the page.

All frequency capping for banner ads must be placed in the site's ad server as well. There is no frequency capping available in the Eyeblaster interface for these ads.

***Note**: For polite banners and standard banners, it is recommended to always use the JavaScript version of the Eyeblaster code, even when the banner is served through an IFRAME on the page. The reason is that the IFRAME version of the Eyeblaster code requires an additional step in the loading process, and using the JavaScript version will result in quicker loading and smaller discrepancy with the ad server measured impressions. This recommendation is not applicable to expandable banners, only to the polite banner and the standard banner formats.*

Ad Serving Limitations – Supported Platforms and Browsers

Out-of-Banner Ads

Platform	Browser	Floating Ad	Wallpaper Ad	Commercial Break	Window Ad
Win (95, 98, 2K, XP, Vista)	**IE5**	**No**	**No**	**No**	**No**
Win (95, 98, 2K, XP, Vista)	**IE5.5**	Yes	Yes	Yes	Yes
Win (95, 98, 2K, XP, Vista)	**IE6, AOL** (IE6)	Yes	Yes	Yes	Yes
Win (XP, Vista)	**IE7**	Yes	Yes	Yes	Yes
Win (95, 98, 2K, XP, Vista)	**NS6.1, NS6.2**	**No**	**No**	**No**	**No**
Win (95, 98, 2K, XP, Vista)	**NS7**	Yes	Yes	No	Yes
Win (95, 98, 2K, XP, Vista)	**NS8**	**No**	**No**	**No**	**No**
Win (95, 98, 2K, XP, Vista)	**FF**	Yes	Yes	No	Yes
Win (95, 98, 2K, XP, Vista)	**Safari**	**No**	**No**	**No**	**No**

Platform	Browser	Floating Ad	Wallpaper Ad	Commercial Break	Window Ad
Mac OS X	**FF 1.x, Flash Player 8,9**	Yes	Yes	No	Yes
Mac OS X	**FF 1.x (**)**	**No**	**No**	**No**	**No**
Mac OS X	**FF 1.x (***)**	**No**	**No**	**No**	**No**
Mac OS X	**NN All Version**	**Image only**	**Image only**	**Image only**	**Image only**
Mac OS X	**Safari 1.3 and Up Flash Player 8**	Yes (****)	Yes (****)	Yes (****)	Yes (****)
Mac OS X	**Safari 1.3 and up (**)**	**No**	**No**	**No**	**No**
Mac OS X	**Safari 1.3 and up (***)**	**No**	**No**	**No**	**No**

Rich Banners

Platform	Browser	Polite Banner	Expandable Banner	Pushdown Banner	Yahoo Floating Ad	MSN Push Down	Video Strip	Floating Expandable
Win (95, 98, 2K, XP, Vista)	**IE5**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**
Win (95, 98, 2K, XP, Vista)	**IE5.5**	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Win (95, 98, 2K, XP, Vista)	**IE6, AOL** (IE6)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Win (XP, Vista)	**IE7**	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Win (95, 98, 2K, XP, Vista)	**NS6.1, NS6.2**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**
Win (95, 98, 2K, XP, Vista)	**NS7**	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Platform	Browser	Polite Banner	Expandable Banner	Pushdown Banner	Yahoo Floating Ad	MSN Push Down	Video Strip	Floating Expandable
Win (95, 98, 2K, XP, Vista)	**NS8**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**
Win (95, 98, 2K, XP, Vista)	**FF**	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Win (95, 98, 2K, XP, Vista)	**Safari**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**
Mac OS X	**FF 1.x, Flash Player 8,9**	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mac OS X	**FF 1.x (**)**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**
Mac OS X	**FF 1.x (***)**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**
Mac OS X	**NN All Version**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**
Mac OS X	**Safari 1.3 and Up Flash Player 8**	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mac OS X	**Safari 1.3 and up (**)**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**
Mac OS X	**Safari 1.3 and up (***)**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**	**Image only**

Standard Banner and VideoClip Ad

		Banner		VCM
Platform	Browser	Standard Banner	Popup Banner	Video Clip Ad
Win (95, 98, 2K, XP, Vista)	**IE5**	**Image only**	**Image only**	**No/ Not supported tags**
Win (95, 98, 2K, XP, Vista)	**IE5.5**	Yes	Yes	Yes
Win (95, 98, 2K, XP, Vista)	**IE6, AOL** (IE6)	Yes	Yes	Yes

		Banner		VCM
Platform	**Browser**	**Standard Banner**	**Popup Banner**	**Video Clip Ad**
Win (XP, Vista)	**IE7**	Yes	Yes	Yes
Win (95, 98, 2K, XP, Vista)	**NS6.1, NS6.2**	**Image only**	**Image only**	**No/ Not supported tags**
Win (95, 98, 2K, XP, Vista)	**NS7**	Yes	Yes	**No/ Not supported tags**
Win (95, 98, 2K, XP, Vista)	**NS8**	**Image only**	**Image only**	**No/ Not supported tags**
Win (95, 98, 2K, XP, Vista)	**FF**	Yes	Yes	**No/ Not supported tags**
Win (95, 98, 2K, XP, Vista)	**Safari**	**Image only**	**Image only**	**No/ Not supported tags**
Mac OS X	**FF 1.x, Flash Player 8,9**	Yes	Yes	**No/ Not supported tags**
Mac OS X	**FF 1.x (**)**	Yes	Yes	**No/ Not supported tags**
Mac OS X	**FF 1.x (***)**	**Image only**	**Image only**	**No/ Not supported tags**
Mac OS X	**NN All Version**	**Image only**	**Image only**	**No/ Not supported tags**
Mac OS X	**Safari 1.3 and Up Flash Player 8**	Yes	Yes	**No/ Not supported tags**
Mac OS X	**Safari 1.3 and up (**)**	Yes	Yes	**No/ Not supported tags**
Mac OS X	**Safari 1.3 and up (***)**	**Image only**	**Image only**	**No/ Not supported tags**

Notes:

(*) All browser/OS combinations not listed here are served gifs for Banner formats and nothing for OOB formats.

(**) Flash Player 7 and less and AD ID from 260500 and up.

(***) Flash Player 7 and less and AD ID lower than 260500 (prior to 6.1.1)

(****) Preview/ offline demo only

In Game

	In Game		
Platform	**GIF**	**Video**	**Integrated Image**
Win 98 SE & above	Yes	Yes (*)	Yes
MAC OS	**No**	**No**	**No**

Note: *For WMP7 and up, the WMV9 codec is required.*

Tracking on the Publisher Ad Server

Apart from the web-based reports provided through the Eyeblaster Platform , a publisher can track Eyeblaster ad impressions and clicks through his/her own ad server.

Out-of-Banner Ads

To track Eyeblaster impressions and clicks, the following steps must be followed:

1. In addition to the ad space setup for the ad, set up a new ad on the ad server. This is usually a 1x1 transparent gif whose sole purpose is to track the ad-impressions and clicks for:

   - **Impressions**: An impression logged for that gif is equivalent to an ad play. This is because each time the ad is played, a request for that gif is made directly from the page being viewed by the user (and only upon the ad being played). The ad server, when such a request is made, then logs an impression for the gif. This happens every time there is an actual ad play, thereby also logging the number of impressions.
   - **Clicks:** A click logged for this gif is equivalent to an ad click. This is because each time the ad is clicked, a request for that gif's redirection URL is made, directly from the page being viewed by the user (and only upon the Eyeblaster being clicked). The ad server, when such a request is made, then logs a click for the gif. This happens every time there is an actual click on the ad.

   Note: *The actual 1x1 gif is never displayed anywhere. It is simply a dummy space set up only for the sake of logging Eyeblaster ad impressions and clicks.*

1. Generate URL's:

   - **An image URL, for calling the 1x1 gif (the "Play-Tracking URL")**. The logs on the ad server for requests made, through this URL, to the 1x1 gif, and will show the number of ad impressions. Note that the 1x1 tracking flight should be targeted exclusively to that specific image URL.
   - **A click-redirection URL, for calling the ad server when an Eyeblaster ad is clicked on (the "Click-Tracking URL").** The logs on the ad server for requests to this URL show the number of clicks on the ad. Note that some site ad servers require that the 1x1 pixel being generated include a clickthrough. In these cases, when asked, you can include any clickthrough link when creating the 1x1 pixel, since this link will be ignored in the click-tracking URL, as the click-tracking URL is used for tracking purposes only and not for clicking through to a site.

   Note: *The 1x1 tracking flight should be targeted exclusively to the specific image URL.*

2. Set the Play-Tracking URL and the Click-Tracking URL in the account on the Eyeblaster Platform.
3. Alternatively, the tracking URL's can also be configured directly through the code served to the page, as follows:

   - For tracking impressions:

     ```
     //Remote servers
     gEbAd.playRS = new Object();
     gEbAd.playRS.strNUrl = "Play Tracking URL";
     ```

     Where:

Play-Tracking_URL is the image URL that the Eyeblaster script will call to report an ad play.

- For tracking clicks:

```
//Interactions
gEbAd.interactions = new Object();
gEbAd.interactions["_eyeblaster"] = "ebN=Click Tracking URL;";
```

Where:

Click-Tracking_URL is the click redirection URL that the Eyeblaster script will call to report a clickthrough.

Notes:

- *These variables, when present, override any URL's entered in the "Third-party tracking" fields at the ad level in the Eyeblaster interface.*
- *Whenever cache-busting is required (as in tracking through DART or OAS), it is possible to include a random number within the call to the tracking server, by including the token `[ebRandom]` or `[timestamp]` within the tracking URL, for example:*

```
gEbAd.interactions["_eyeblaster"] = "ebN=
http://iv.doubleclick.net/ad/view.ivillage
/fdoc;ad=fdoc;sz=1x1?ord=[timestamp];";
```

  - Using variables in the code tracks on the flight level only, not on the individual ad level within the flight. To track on the ad level, enter the URL's directly into the Eyeblaster interface.
  - If dynamic variables from the site's ad server are part of the tracking code, use variables to track. This code will be passed through the site's ad server where it will replace the variables with the correct data.

Banner Ads

Banner ads do not need impression tracking. The ratio between Eyeblaster ad impressions and the ad server impressions is 1:1. To track clicks for banner ads, follow the steps above for click tracking of out-of-banner ads.

Working with DFP5

Doubleclick's ad server "DART for Publishers (DFP)" version 5 defined a new ad type, the "interstitial" ad which enables easier trafficking and tracking of out-of-banner ads. This feature should not be used for any of the banner formats.

This ad type is used in conjunction with modified ad tags, containing "`DCOPT=ist`". This section is only applicable if the website's ad tags contain the new `DCOPT` variable.

- **Trafficking**: When trafficking using the interstitial ad type, the rich-media ad created in DFP5 needs to be defined as an "interstitial" ad. The ad can then be targeted to any tag on the site pages that is set to accept the interstitial type (i.e., contains the "`DCOPT=ist`" variable).

  The special property of the tag is that two ads can be targeted to it at the same time – a standard banner ad (such as a gif image or a Flash movie), and an out-of-banner ad. In the past, to display these two ads together required piggybacking the out-of-banner ad within the same ad space as the standard banner ad. Now, this piggybacking is done "on-the-fly" without needing to piggyback the Eyeblaster ad on all the ROS banners targeted to the tag on the page. Therefore, the banner can be a ROS banner, for example, and the rich-media ad an Eyeblaster ad.

- **Tracking**: For the Interstitial Format, a new mechanism was built in, to synchronize the impression count on the ad server (i.e., on DART) with the actual plays and clicks. For that purpose, the out-of-banner ad needs to be defined as an "interstitial" ad type, and the tracking URLs assume this generic format:

  Play-tracking URL: `%i%h/dot.gif`

Click-tracking URL: **"**%c

These URL's are added to the Eyeblaster code automatically once the DART interstitial format is selected for the flight or at the account level.

Notes:

- *The Eyeblaster interface includes an option for sites using the DART Interstitial Format. If your site is using this feature, go to* ***Preferences -/ Account -/ Settings*** *in the Eyeblaster interface and select the option* ***Out of banner flights are served using the DART Interstitial Format****.*

  This means the following:

  - The frequency cap is always set to unlimited.
  - Control over-delivery is turned off.
  - Eyeblaster will serve ads as long as DART makes calls to Eyeblaster, ignoring the volume set in the Eyeblaster interface.
  - The impression and click tracking variables will automatically be inserted into the Eyeblaster code when it is generated in the Eyeblaster interface.
- If there is an out-of-banner flight that is not being served through the Interstitial Format for some reason, this option can be disabled at the flight level, within the **Rotation and Frequency** tab. Floating ads with banner reminders is such an example, and this ad format should be served as rich media, not as interstitial.
- Since DART tracks plays and clicks, the number of impressions booked in DART should be the actual number of contracted plays (DART will automatically adjust for the ratio between impressions and plays).
- Since DART tracks plays and clicks, frequency capping must be done directly in DART instead of in the Eyeblaster interface. With the Interstitial Format feature available through the Eyeblaster interface, frequency capping is automatically set to unlimited.
- Although DART correctly records impressions and clicks, it will not accurately predict inventory. This is because DART serves as many tags as needed to reach the impression goal, but does not record how many times the tags were served.
- Always use the Code Generator to generate the tags to ensure the tags appear in the right format and order.

Defining Publisher 3rd Party Tracking of Ads

Once all the flights are published and the Ads are approved and copied to their relevant flights, you can start adding tracking URL's. For small campaigns, tracking URL's can be added per ad in the Eyeblaster interface; for larger campaigns, it is recommended to export the media plan and fill in the URL's in the resulting Excel file which, after being updated, can then be imported.

To add tracking URL's:

1. From the main menu, click **Creative / Ads**. The **Ads** dialog appears.
2. Either right-click the relevant ad and select **Edit** from the popup menu, or select the ad and click  **Edit Ad** on the horizontal taskbar. The **Edit Ad** dialog appears.
3. Click the Actions & Tracking tab.
4. Expand the Publisher 3rd-party tracking section and enter the Click tracking URL and Impression tracking URL.

Using Special JavaScript Variables

In addition to the standard Eyeblaster tags and the Eyeblaster tracking variables, further advanced options for trafficking and tracking using JavaScript variables are available.

Tracking Ad Performance across Website Pages

Using the following script variables allows a publisher to generate detailed reports for examining an ad's performance across different pages of the site, or across different sites within a network:

```
gEbAd.strPage = “Page Name";
```

Where:

Page Name is the name of the page that is being tracked.

Adding this variable to the rest of the Eyeblaster Code will allow Eyeblaster to distinguish ad plays and clicks on the page from ad plays and clicks on other pages. Consequently, the Publisher can view data in the Campaign reports section on how the ad performed across the different pages or network sites.

Serving Ads within a Frameset

The simplest way to serve an ad within a Frameset is to serve the Eyeblaster Code directly to the Frame on which the Eyeblaster ad is to be displayed.

However, it may happen that the ads are to appear on a frame within a frameset, while the code cannot be served directly to the same frame. In this case, for an IFRAME implementation only, the following variable should be added to the Eyeblaster Code served to the IFRAME:

```
var gstrEbDisplayPos = “top.FrameName";
```

Where:

FrameName is the name of the Frame on which the ad should appear.

Right-aligning Ads

At times, it becomes necessary to right-align an ad, so that the ad always remains in a designated area, even when a user resizes the browser window. This is typical for sites that implement a column of ads, aligned to the right edge of the browser.

To set an ad to align with the right edge, the following variable must be served along with the rest of the Eyeblaster Code:

```
gEbAd.nRightAlignment="Stop_pixel";
```

Where:

Stop_pixel denotes the pixel (i.e., page width) where right-alignment is no longer effective.

***Note**: When setting the pixel positioning of the ad, within the **Positioning** settings in the Eyeblaster interface, X=0 becomes the right-hand side instead of the left.*

Disabling Eyeblaster Ad Compression

Eyeblaster automatically compresses all JavaScript and Flash files that are served to the user's browser. If the ad is to be served without compression, the following variable must be used within the Eyeblaster code:

```
gfEbUseCompression = false;
```

This tag cannot be generated in the Eyeblaster interface. Consult your Eyeblaster Client Services representative before using this tag.

Preventing Ad Caching

Eyeblaster automatically caches the JavaScript and Flash files that are served to the user. This helps improve the serving ratio when the user is frequently exposed to ads. If the ad is to be downloaded every time the call is made to the Eyeblaster ad servers, the following variable should be used:

```
gfEbCacheResources = false;
```

This tag cannot be generated in the Eyeblaster interface. Consult your Eyeblaster Client Services representative before using this tag.

Special Tagging Instructions for Certain Ad Formats

Certain ad formats require additional attention when setting up the Eyeblaster Code to ensure proper delivery. Below are important notes for these ad formats.

Window Ad Format

The Window Ad is an ad set to appear within a new browser window. The Eyeblaster Code used to deliver the ad is the same as in other Eyeblaster formats, and is generated within the Eyeblaster Code Generator as in other formats. Note, however, that the Window Ad can be set up (usually by the agency) to appear in one of two modes:

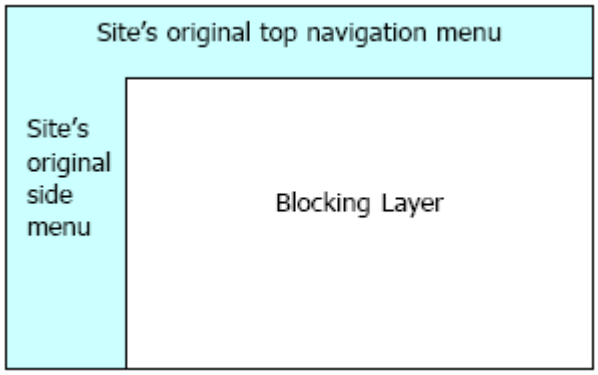
- **Standard mode**: This is the default mode, where the ad is set to appear as soon as it is ready. I.e., once the content has been downloaded, the new window appears.
- "**Next Page**" mode: In this mode the ad is set to appear only when the user has moved on to another page. In this mode, the ad will be downloaded, but will not be displayed, until the Eyeblaster Code for the flight is encountered a second time. It is essential, therefore, that the code be delivered more often, to increase the probability of a user encountering the Eyeblaster Code for the second time.

  It is therefore recommended, when booking the ad, to assume a higher impression-to-play ratio, typically 6:1 instead of 3:1. As before, it is recommended to measure the actual ratio after the start of the campaign, and adjust the booked impressions if needed.

Commercial Break Format

The Commercial Break allows you to display a full-page ad immediately when a user enters the site, or in a transition between two website pages. What distinguishes the Commercial Break from other formats is that the ad does not interact with the page it is running on. Rather, a portion of the page (determined by the publisher) is blocked from view. Within that area,called the "blocking layer", the ad can play freely, while it cannot leave the bounds of the area and interact with the rest of the page.

This mechanism allows the site to maintain its look-and-feel when displaying the ad on the page. A typical setup of the blocking layer is:



Playing the Commercial Break Ad

Since it is essential that the ad in a Commercial Break appear as soon as possible, before the page content is shown to the user, it is highly recommended to:

- Serve the Eyeblaster Code at the beginning of the HTML page, so that it is executed by the browser prior to other elements on the page being rendered. The ad code should be placed right underneath the <body> tag.
- Use the instant download and instant play mode. This way the ad will immediately start downloading with the page and will appear as soon as possible.
- Make sure the "Immediate display" checkbox is selected within the "Ad Format" tab in the ad definitions.

This is typically determined by the agency setting up the ad. If the Immediate Display option has not been selected, please contact Eyeblaster to verify the setup.

How Ads Interact with Objects on a Web Page

The following information describes how ad elements behave in relation to other elements on a site's page, clashes that may occur and how it affects what is displayed.

Interaction with IFRAMEs and Drop-down Menus on a Page

- In IE5.0, Netscape and Firefox browsers, the floating ad always appears behind an IFRAME. A known issue with DHTML-based "floating" ads. This is, however, resolved in IE5.5 or higher and in Netscape.
- Behind drop-down menus. In all IE browsers supported by Eyeblaster. (There is no problem with Netscape and Firefox).

One solution for both of these issues is to correctly position the Eyeblaster ad to minimize interaction with IFRAMEs and Drop-down menus on the page. This is not done as part of the Flash design; instead, you position the ad while setting up the ad in the Eyeblaster Interface. While creating or editing the ad, click the positioning tab, and set the X (horizontal) and Y (vertical) coordinates to control the positioning of the ad.

You can also hide the IFRAMEs and drop-down menus that collide with the ad. This is also not done as part of the Flash design, but in the Positioning tab in the Eyeblaster Interface. While creating or editing the ad, click the positioning tab, and mark the **Hide IFRAMEs** and **Hide drop-down's** checkboxes. The IFRAMEs and/or drop-down menus that collide with the ad are then hidden while the ad plays.

There are two exceptions in which case *all* IFRAME's and drop-downs on the page are hidden:

- If, when defining a floating ad template, a parameter other than the default "no movement" is selected.
- If the ad is defined to scroll with the page.

Interaction with Java Applets on a Page

In all browsers supported by Eyeblaster, the floating ad always appears behind Java applets on the page. You can change this using the Eyeblaster Interface. While creating or editing the ad, click the positioning tab, and select **Hide Java Applets**. This hides the Java applets on the page while the ad plays.

Interaction with Shockwave Elements on a Page

In all browsers supported by Eyeblaster, the floating ad always appears behind Shockwave elements on the page.

Interaction with Flash Movies on a Page

By default, a floating ad appears behind other Flash elements on the page when these elements have their "wmode" attribute set to "window" or not set at all (default).

You can change this in the following ways:

- Define the Flash object on the website as "Opaque". No special configuration to the Flash movie is required. The Eyeblaster Flash movie will appear on top of the site's Flash.
- While creating or editing the ad, click the positioning tab and select the option for hiding Flash elements while the ad is playing. All Flash elements that are configured in "window" mode (i.e. non-opaque or transparent) on the page and overlap with the ad resource are hidden while the ad plays.
- Set the ad to "window" mode. In this case the ad resource will not be transparent and will block everything under it.

  Setting the ad "wmode" is done by adding the "strWmode" tag to the ad tags

Interactions between DHTML and Expandable Banners

When a panel of an Expandable Banner opens, the panel displays on top of the content of the page. The exception is if there is a DHTML layer on the page, such as a site DHTML navigation menu, in which case

the DHTML layer shows on top of the Eyeblaster panel. This also depends on the zindex of the navigational menu. (Eyeblaster uses a zindex of 10,000).

If the expanded panel is rectangular or square, (with no transparent areas) you can set the panel as non-transparent in the Eyeblaster Interface. This sets the panel to appear above any other elements on the page.

If the expanded panel is not rectangular or square, or if it includes transparency by design, such as a floating element, then there is an additional option within the Eyeblaster interface to correct DHTML conflicts. While creating or editing the ad, click the positioning tab, and select **Auto-set DHTML layer hierarchy**. This sets the panel to show on top of the DHTML elements on the page. (This is applicable to a specific page structure and needs to be tested carefully when used).

Corporate Headquarters, Eyeblaster, Inc.
135 West 18th Street, 5th Floor, New York, NY 10011, Tel: 646.202.1320, Fax: 212.686.9208
info@eyeblaster.com www.eyeblaster.com