

Sizmek Formats

# Custom Header

Build Guide

Sizmek<sup>TM</sup>

## Table Of Contents

Overview .....	3
Supported Platforms .....	3
Specifications .....	3
Known Issues .....	3
Ad Behavior Screenshots .....	4
Ad Interaction Tracking .....	6
Template Included Files .....	6
Setting up in Workshop .....	6
Customizing the Format .....	7
Leave behind (leaveBehind fla) .....	7
Panel (panel fla) .....	7
Fallback images .....	13
Setting up in Platform .....	14
Custom Variables .....	15
Configuring in Workshop .....	15
Configuring in Platform .....	15
Custom Variables Definitions .....	15
Setting Up Custom Interactions/Ad URLs in Platform .....	16
Configuring in Platform .....	16
SUPPORT .....	17

## Overview

The Custom Header is a new format designed specifically for The New MSN site launching in October 2014. It is a Flash-based rich media banner that will display itself in three different dimensions that match the responsive layout of The New MSN. In each state the Custom Header is able to display a different experience for the user if desired. The Custom Header includes a Show Ad/Hide Ad functionality very similar to the IAB Billboard. The state of view of the Custom Header is saved via a cookie on the user's machine.

## Supported Platforms

Platform	Supported Browser Version
Windows XP/7/8/Surface	Internet Explorer 7+, Firefox, Chrome, Safari
Mac OSX 10.8 & 10.9	Firefox, Chrome, Safari

## Specifications

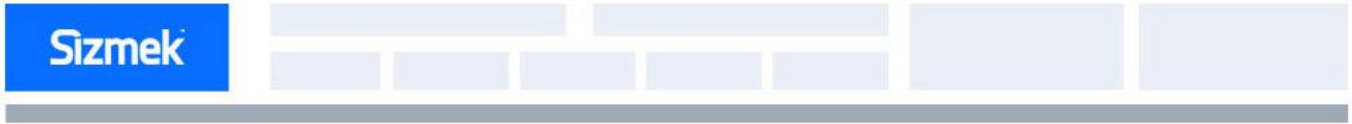
- Format Version 1.1
- Supports ActionScript 3.0
- Supports Flash Player 10 and above
- Template : [Download here](#)
- Demo : [Click here](#)

## Known Issues

- The Custom Header cannot be previewed locally when published. The ad must be uploaded to the Sizmek platform and viewed in the platform preview.

## Ad Behavior Screenshots

### Leave behind (flash)



Show Ad



### 1272x328 panel (flash)



**Sizmek™**  
Custom Header, 1272x328

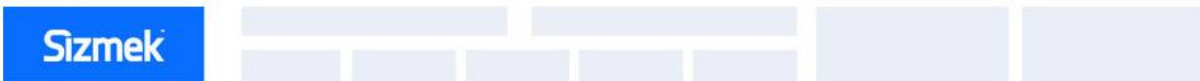
Hide Ad [x]

UserAction

ClickThrough



### 970x250 panel (flash)



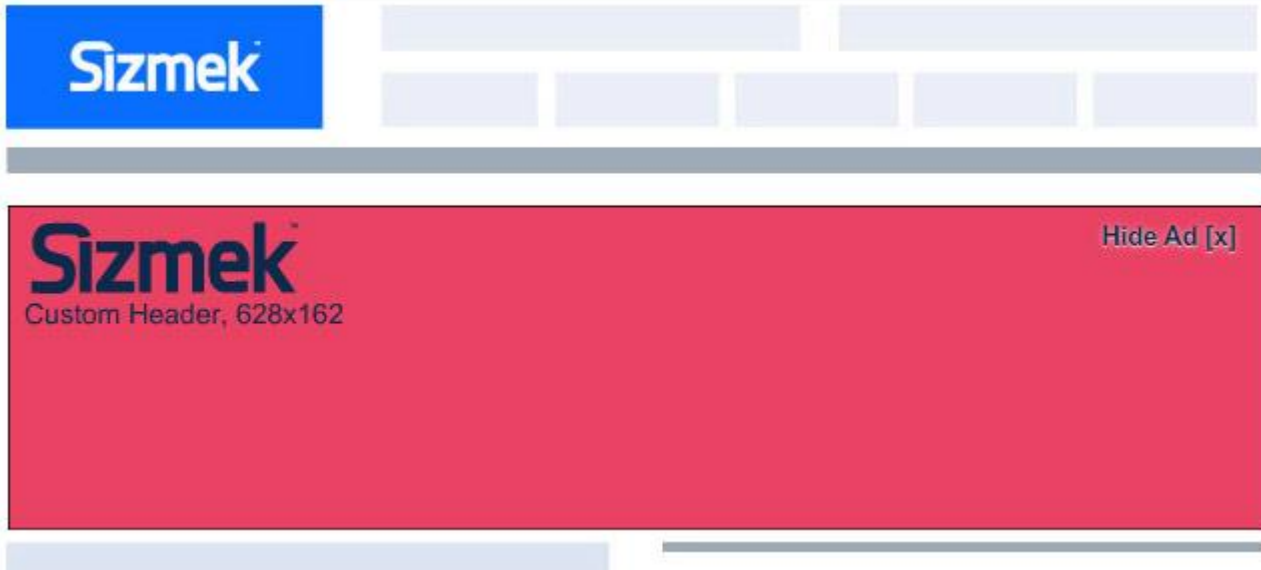
**Sizmek**  
Custom Header, 970x250

Hide Ad [x]

UserAction

ClickThrough

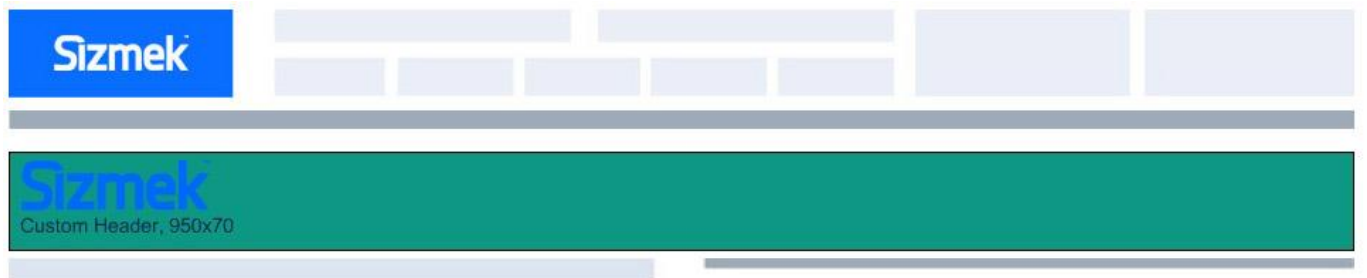
628x162 panel (flash)



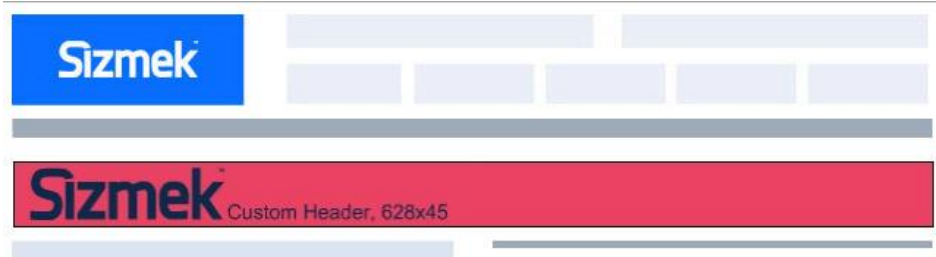
1272x92 (image)



950x92 (image)



628x45 (image)



## Ad Interaction Tracking

The Custom Header template provides the following custom interaction tracking by default. The name values of the custom interactions may be modified if desired by the developer.

Show Ad button click (Automatic Event)

Hide Ad button click (Automatic Event)

Ad Viewed in 2-column view (Automatic Event – fired when ad is loaded or page view is changed)

Ad Viewed in 3-column view (Automatic Event – fired when ad is loaded or page view is changed)

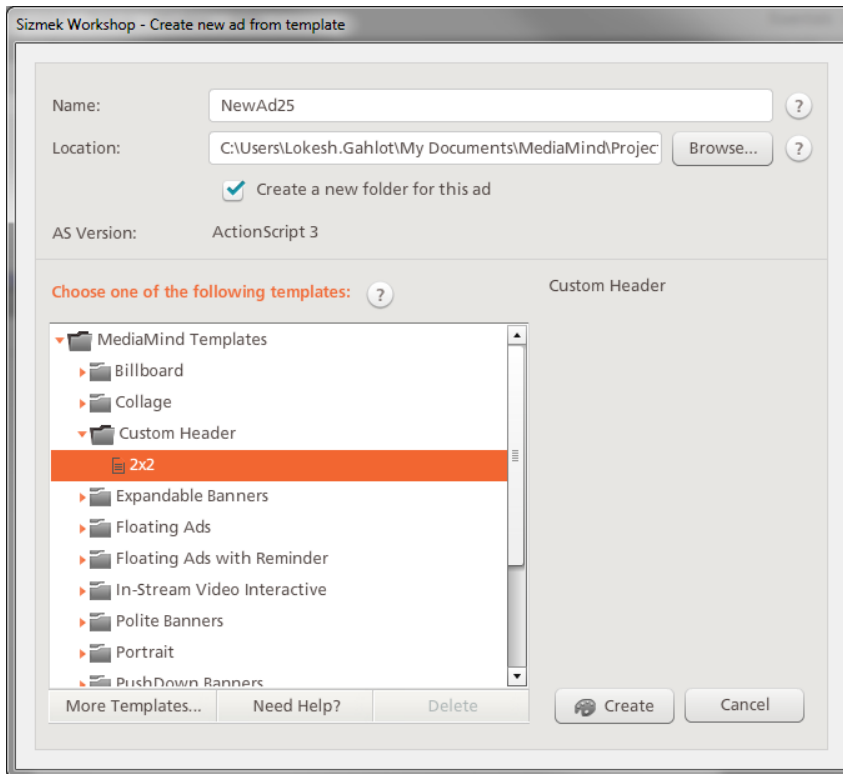
Ad Viewed in 4-column view (Automatic Event – fired when ad is loaded or page view is changed)

## Template Included Files

Filename	Description
2x2 fla	2x2 Flash banner, the place holder
panel fla	1272x328 Flash panel
leaveBehind fla	88x31 Flash leave behind
CustomHeader.as	AS3 class for applying custom header logic
2x2 gif	Backup image served when flash and Javascript is disabled
628x45.jpg	Backup image served when flash is disabled (In 2 columns view)
970x70.jpg	Backup image served when flash is disabled (In 3 columns view)
1272x92.jpg	Backup image served when flash is disabled (In 4 columns view)

## Setting up in Workshop

- In your file browser, browse to and run the Custom Header MXP/ZXP (Custom\_Header.mxp or Custom\_Header.zxp), and then follow the prompts in the Adobe Extension Manager to install the extension. You can skip this step if you have already installed the MXP.
- In the Sizmek Workshop™ for Flash, select New from template.
- For AS Version, select ActionScript 3.
- Expand the Sizmek Templates > Custom Header folder, and select 2x2.



- Click Create.

## Customizing the Format

### Leave behind (leaveBehind.fla)

**Do not** modify “leave behind” fla.

### Panel (panel.fla)

1. Open panel.fla
2. On the content layer there is movieclip called **mc\_content**, double click to open this movieclip
3. In side **mc\_content**, there are 5 layers.
  - action:- has action script code.
  - Hide Ad button: - Hide button graphic (Do not modify the graphic).
  - 628x162:- content for 628x162 size panel. Content graphic can be add/update.
  - 970x250:- content for 970x250 size panel. Content graphic can be add/update.
  - 1272x328:- content for 1272x328 size panel. Content graphic can be add/update.
4. All content movieclips should be align to top-left of the stage.
5. On “action” layer contents following code

## CUSTOM HEADER

```
var ch:CustomHeader = CustomHeader.getInstance();
var str:String = EBBase.GetVar("mdPanelWidths");
if(str)
{
    var panelWidth:Array = str.split(",");
    var settings:Object = {};
    settings["w"+ panelWidth[0]] = {content:mc628x162, props:{scaleX:1,
scaleY:1, playOnce:false}};
    settings["w"+ panelWidth[1]] = {content:mc970x250, props:{scaleX:1,
scaleY:1, playOnce:false}};
    settings["w"+ panelWidth[2]] = {content:mc1272x328, props:{scaleX:1,
scaleY:1, playOnce:false}};

    /*
    //One movieclip can be used for all columns. Here is the sample code
    settings["w"+ panelWidth[0]] = {content:mc1272x328,
props:{scaleX:0.4937, scaleY:0.4935, playOnce:false}};
    settings["w"+ panelWidth[1]] = {content:mc1272x328,
props:{scaleX:0.7625, scaleY:0.7624, playOnce:false}};
    settings["w"+ panelWidth[2]] = {content:mc1272x328, props:{scaleX:1,
scaleY:1, playOnce:false}};
    */
    ch.addJSEventListener(CustomHeader.STAGE_RESIZE, resizeHandler);
    ch.init(settings, this);
}

function resizeHandler(e:Object)
{
    switch(e.stageWidth)
    {
        case 628:
            EBBase.AutomaticEventCounter("Two_Columns_Viewed");
            break;

        case 970:
            EBBase.AutomaticEventCounter("Three_Columns_Viewed");
            break;

        case 1272:
            EBBase.AutomaticEventCounter("Four_Columns_Viewed");
            break;
    }
}
```

6. Tracking also added on 'action' layer. Please see **yellow** highlighted code.

```
var ch:CustomHeader = CustomHeader.getInstance();
var str:String = EBBase.GetVar("mdPanelWidths");
if(str)
{
    var panelWidth:Array = str.split(",");
    var settings:Object = {};
```



## CUSTOM HEADER

```

        settings["w"+ panelWidth[0]] = {content:mc628x162, props:{scaleX:1,
scaleY:1, playOnce:false}};
        settings["w"+ panelWidth[1]] = {content:mc970x250, props:{scaleX:1,
scaleY:1, playOnce:false}};
        settings["w"+ panelWidth[2]] = {content:mc1272x328, props:{scaleX:1,
scaleY:1, playOnce:false}};

        /*
        //One movieclip can be used for all columns. Here is the sample code
        settings["w"+ panelWidth[0]] = {content:mc1272x328,
props:{scaleX:0.4937, scaleY:0.4935, playOnce:false}};
        settings["w"+ panelWidth[1]] = {content:mc1272x328,
props:{scaleX:0.7625, scaleY:0.7624, playOnce:false}};
        settings["w"+ panelWidth[2]] = {content:mc1272x328, props:{scaleX:1,
scaleY:1, playOnce:false}};
        */
        ch.addEventListener(CustomHeader.STAGE_RESIZE, resizeHandler);
        ch.init(settings, this);
    }

    function resizeHandler(e:Object)
    {
        switch(e.stageWidth)
        {
            case 628:
                EBBase.AutomaticEventCounter("Two_Columns_Viewed");
                break;
            case 970:
                EBBase.AutomaticEventCounter("Three_Columns_Viewed");
                break;
            case 1272:
                EBBase.AutomaticEventCounter("Four_Columns_Viewed");
                break;
        }
    }
}

```

7. User can modify scale of the content movie clips (mc628x162, mc970x250, mc1272x328). Also one content movie clip can be used for all three sizes. Check the **highlighted** code. Here is the example.

```

var ch:CustomHeader = CustomHeader.getInstance();
var str:String = EBBase.GetVar("mdPanelWidths");
if(str)
{
    var panelWidth:Array = str.split(",");
    var settings:Object = {};
    /*settings["w"+ panelWidth[0]] = {content:mc628x162, props:{scaleX:1,
scaleY:1, playOnce:false}};
    settings["w"+ panelWidth[1]] = {content:mc970x250, props:{scaleX:1,

```

## CUSTOM HEADER

```

scaleY:1, playOnce:false}};
    settings["w"+ panelWidth[2]] = {content:mc1272x328, props:{scaleX:1,
scaleY:1, playOnce:false}};
    */

    //One movieclip can be used for all columns. Here is the sample code
    settings["w"+ panelWidth[0]] = {content:mc1272x328,
props:{scaleX:0.4937, scaleY:0.4935, playOnce:false}};
    settings["w"+ panelWidth[1]] = {content:mc1272x328,
props:{scaleX:0.7625, scaleY:0.7624, playOnce:false}};
    settings["w"+ panelWidth[2]] = {content:mc1272x328, props:{scaleX:1,
scaleY:1, playOnce:false}};

    ch.addJSEventListener(CustomHeader.STAGE_RESIZE, resizeHandler);
    ch.init(settings, this);
}

function resizeHandler(e:Object)
{
    switch(e.stageWidth)
    {
        case 628:
            EBBase.AutomaticEventCounter("Two_Columns_Viewed");
            break;

        case 970:
            EBBase.AutomaticEventCounter("Three_Columns_Viewed");
            break;

        case 1272:
            EBBase.AutomaticEventCounter("Four_Columns_Viewed");
            break;
    }
}
}

```

8. 628x162, 970x250 and 1272x328 layers contain following code.

```

var ch:CustomHeader = CustomHeader.getInstance();
ch.addJSEventListener(CustomHeader.STAGE_RESIZE, resizeHandler);
function resizeHandler(e:Object)
{
    var str:String = "w" + e.stageWidth;
    if(ch.settings[str].content == this)
    {
        if(this.visible == false)
        {
            this.visible = true;
        }

        if(ch.settings[str].content.playOnce)
        {
            //video or animation play code should be here

```

## CUSTOM HEADER

```

        //video or animation play only once when it loads.
        ch.settings[str].content.playOnce = false;
    }
}
else
{
    this.visible = false;
}
}
}

```

This code handles visibility of the all 3 content when browser resize. User can also handle video or animation start/stop with above code for each content. It starts video on **every time** when panel **resize and show**.

Here is the example.

```

var ch:CustomHeader = CustomHeader.getInstance();
ch.addJSEventListener(CustomHeader.STAGE_RESIZE, resizeHandler);
function resizeHandler(e:Object)
{
    var str:String = "w" + e.stageWidth;
    if(ch.settings[str].content == this)
    {
        if(this.visible == false)
        {
            Videoplayer.loadAndPay(4);
            this.visible = true;
        }

        if(ch.settings[str].content.playOnce)
        {
            //video or animation play code should be here
            //video or animation play only once when it loads.
            ch.settings[str].content.playOnce = false;
        }
    }
}
else
{
    Videoplayer.stop();
    this.visible = false;
}
}
}

```

If user wants to play video or animation only on **1<sup>st</sup> load**.

Here is the example.

Inside **mc\_content**, on 'action' layer. Change the highlighted code

## CUSTOM HEADER

```

var ch:CustomHeader = CustomHeader.getInstance();
var str:String = EBBase.GetVar("mdPanelWidths");
if(str)
{
    var panelWidth:Array = str.split(",");
    var settings:Object = {};
    settings["w"+ panelWidth[0]] = {content:mc628x162, props:{scaleX:1,
scaleY:1, playOnce:true}};
    settings["w"+ panelWidth[1]] = {content:mc970x250, props:{scaleX:1,
scaleY:1, playOnce:true}};
    settings["w"+ panelWidth[2]] = {content:mc1272x328, props:{scaleX:1,
scaleY:1, playOnce:true}};

    /*
    //One movieclip can be used for all columns. Here is the sample code
    settings["w"+ panelWidth[0]] = {content:mc1272x328,
props:{scaleX:0.4937, scaleY:0.4935, playOnce:false}};
    settings["w"+ panelWidth[1]] = {content:mc1272x328,
props:{scaleX:0.7625, scaleY:0.7624, playOnce:false}};
    settings["w"+ panelWidth[2]] = {content:mc1272x328, props:{scaleX:1,
scaleY:1, playOnce:false}};
    */
    ch.addJSEventListener(CustomHeader.STAGE_RESIZE, resizeHandler);
    ch.init(settings, this);
}

```

And inside the content (mc\_628x162, mc\_970x250 and mc\_1272x328) movie clip.

```

var ch:CustomHeader = CustomHeader.getInstance();
ch.addJSEventListener(CustomHeader.STAGE_RESIZE, resizeHandler);
function resizeHandler(e:Object)
{
    var str:String = "w" + e.stageWidth;
    if(ch.settings[str].content == this)
    {
        if(this.visible == false)
        {
            this.visible = true;
        }

        if(ch.settings[str].content.playOnce)
        {
            //video or animation play code should be here
            //video or animation play only once when it loads.
            Videoplayer.loadAndPay(4);
            ch.settings[str].content.playOnce = false;
        }
    }
    else
    {
        Videoplayer.stop();
        this.visible = false;
    }
}

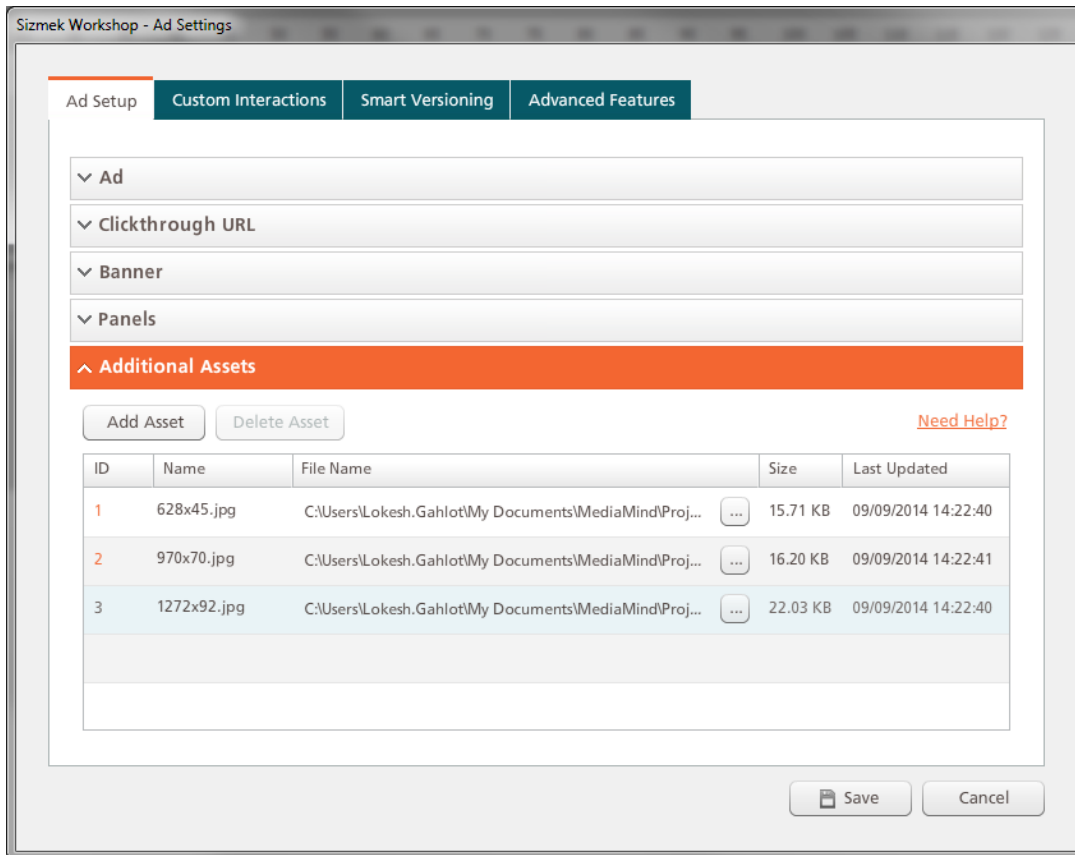
```

```
}  
}
```

## Fallback images

### Update in the MDX workshop

1. Open MDX workshop.
2. Create custom header ad.
3. Go to **Setting>>Ad setup**
4. Click on **Additional Assets**



5. Update the images according to dimension. If user update the additional id of image, user have to have update the **mdFallbackImagesID** in the custom var section.

## CUSTOM HEADER

### Update in the MDX platform

1. Create the ad
2. Go to Additional assets section



Asset ID	Creative Asset	Size	Last Updated
3	Lokesh_assets/Custom_header_QA_Sep8_2014/1272x92.jpg	22KB	9/8/2014 1:11:56 PM
2	Lokesh_assets/Custom_header_QA_Sep8_2014/970x70.jpg	16KB	9/8/2014 1:11:56 PM
1	Lokesh_assets/Custom_header_QA_Sep8_2014/628x45.jpg	15KB	9/8/2014 1:11:56 PM

3. Update the images according to dimension. If user update the additional id of image, user have to have update the **mdFallbackImagesID** in the custom var section.

## Setting up in Platform

The custom script should already be attached to the ad when it is uploaded from the MXP/ZXP. You can confirm this by doing the following.

1. In Sizmek MDX/ZXP, go to the Setup tab.
2. Select the Advanced Features section, and expand the Custom JS area.
3. Copy and paste this code to the Trigger JavaScript on Scripts Pre-load text field:

**New Client Delivery :**

```
http://ds.serving-sys.com/BurstingRes/CustomScripts/PL_CustomHeader_CFV001.js;  
http://ds.serving-  
sys.com/BurstingRes/CustomScripts/PL_CustomHeader_Plugin_MSN_CFV001.js
```

- 3a. Make sure the “Use new client mechanism” checkbox is selected.
4. Click **Save**.

## Custom Variables

Custom Variables for the Custom header can be configured using either the Sizmek Workshop or Sizmek MDX once your ad is created.

### Configuring in Workshop

- In the Sizmek Workshop, go to the Settings tab.
- Next select the Variables menu.
- Click the Add Variable button to enter the appropriate values or double click an existing variable to edit it.
- When are done click the Save button.
- When you upload the ad these values will maintain their state in the platform.

### Configuring in Platform

- In Sizmek MDX, go to the Setup tab.
- Select the Advanced Features section, and expand the Variables area.
- Click the New button to enter the appropriate values or double click an existing variable to edit it.
- When done, click the Save button.

## Custom Variables Definitions

Name: <i>Type</i>	Default Value	Description	Accepted Values
<b>mdAutoRepositionInterval</b> :Number	100	The amount of milliseconds to realign the default panel.	1+ If 0 then the interval will not start.
<b>mdEyeDivZIndex</b> : <i>Number</i>	undefined	The default z-index of the eye div.	0 to 10000 To leave the z-index unchanged us the string "undefined"
<b>mdFallbackImagesID</b> :String	1,2,3	The default additional assets id for fallback images.	String, comma separated assets ids.

**Note: -**

For MSN prime website set **mdEyeDivZIndex** value to 3 and set **mdAutoRepositionInterval** to 200.

## Setting Up Custom Interactions/Ad URLs in Platform

### Configuring in Platform

- In Sizmek MDX, go to the **Interactions & Ad URLs** tab.
- Select the **Custom Interactions** section.
- Click to **[Edit]** to change the setting. (*Refer the highlighted parts below.*)
- Uncheck both “*inherit*” and “*Close Ad Parts*” boxes for **clickthrough** button.
- Remember to insert the URL(s).

*NOTE : If you have more than one button, kindly uncheck the boxes and insert the URLs respectively.*

Inherit	Interaction Name	Interaction Type	Display Name	Close Ad Parts	Include in Interaction Rate	URL
<input checked="" type="checkbox"/>	Four_Columns_Viewed	Automatic Event	Four_Columns_Vie...	<input type="checkbox"/>	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	HideAd_Button	Automatic Event	HideAd_Button	<input type="checkbox"/>	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	ShowAd_Button	Automatic Event	ShowAd_Button	<input type="checkbox"/>	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	Three_Columns_Viewed	Automatic Event	Three_Columns_Vi...	<input type="checkbox"/>	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	Two_Columns_Viewed	Automatic Event	Two_Columns_Vie...	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	clickthrough	Clickthrough	clickthrough	<input type="checkbox"/>	<input checked="" type="checkbox"/>	http://www.sizmek.com/
<input checked="" type="checkbox"/>	InteractionName	User Action	InteractionName	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

- When done, click the **Save** button.



## SUPPORT

### MSA Ad Submission Guidelines

All creative must meet the [MSA Ad Submission Guidelines](#)

### MSA Creative Acceptance Policy

All creative must meet the requirements of the [MSA Creative Acceptance Policy](#)

Contact your local yield representatives to find out the availability and price of this ad product in your market.

### Browser Support

Please refer to [Sizmek's support matrix](#) for more information.

### Sizmek Support Contacts

For questions and assistance please contact:

- **South East Asia** creative support : [sea.creative@sizmek.com](mailto:sea.creative@sizmek.com)

### Online Sizmek training:

[More training material can be found in the Sizmek Training Zone](#)

A login is required to be able to access this material. [Please contact your local Sizmek office for support.](#)

## Notice

The information contained in this document is proprietary and confidential to Sizmek and/or any of its affiliated companies. Disclosure, copying, reproduction, storing or any use of this document or any part thereof without the express prior, written consent of Sizmek or its authorized representatives is strictly prohibited. The information furnished in this document is believed to be accurate and reliable. However no responsibility is assumed by Sizmek for the use of this information. Sizmek reserves the right to make changes to the information included in this document at any time and without notice.

Copyright © 2014 Sizmek. All rights reserved.

# Sizmek<sup>TM</sup>

Flash is either a registered trademark or trademark of Adobe Systems Incorporated in the United States and/or other countries.

Trademark Note: Sizmek, the Sizmek logo, Sizmek Rich Media, Sizmek Mobile, Sizmek Video, Sizmek Channel Connect, Sizmek Workshop, etc. are trademarks and/or registered trademarks of Sizmek. All other trademarks are the property of their respective owners.