

Sizmek Blocks

Weather Block

Build Guide

SizmekTM

Table of Contents

Overview	2
 The Weather Block Templates	2
Basic Weather Block	3
Switch Weather Block	3
Supported Platforms	4
Demos/Downloads	4
Known Issues.....	4
Benefits	4
Use Case	4
Included Files	5
Implementing the Block.....	6
Configuring in Flash.....	6
Configuring the Basic Weather Block.....	7
Flash Implementation.....	8
Configuring the Switch Weather Block.....	8
Flash Implementation.....	9
Custom Interactions.....	10
HAMWeather Aeris Account Setup.....	11

Overview

The **Weather Block** loads current and future forecasts into your creative. Users can track the weather and change the creative based on the temperature or the weather type. The Weather Block is built using the **HAMweather Aeris API** (<http://www.hamweather.com>). The Weather Block utilizes the Aeris auto location method to detect the location of the audience and display weather that is accurate based on their location.

Please note: this block utilizes the HAMweather Aeris Weather API. You must set up an account with HAMweather and obtain a unique Consumer ID and Consumer secret. These will be used to connect to the Aeris API. The Weather Block **will not** work until you have entered your Consumer ID and Consumer Secret into the Config object.

The Aeris Weather API has multiple available endpoints that allow for different responses, for example, Forecasts, Tides, etc. For more information on the options available visit the HAMweather website.

<http://www.hamweather.com/support/documentation/aeris/>

The Weather Block utilizes the Aeris auto location method to detect the location of the audience, however this feature can be bypassed in the Config object under options.

The Weather Block Templates



Basic Weather Block

This template displays the current location, temperature, feels like and time.



Switch Weather Block

This template allows you to assess the weather on either the temperature or the weather type (ie rain/snow) and display creative specific to the results. For example, if it is warm, show a cold drink, if it is cold, show a warm drink.



Supported Platforms

Platform	Supported Version
Windows	Internet Explorer 9+, Firefox, Chrome, Safari
Mac	Firefox, Chrome, Safari

Demos/Downloads

To download the Block, view a demo of the Block or get the latest copy of the build guide, please visit the [Formats & Blocks tab of the Creative Zone](#). For more information, contact your Creative Development Specialist.

Known Issues

- **Error 1120** - Access of undefined property Blocks
 - If you receive this error when publishing the FLA, you need to update your Workshop™ for Flash (MXP).
- The Weather Block uses Javascript with External Interface calls, therefore it will not work in Instream Ads.

Benefits

- An existing API makes it easy to set up the Block in your creative with variable configurations.
- Advertise your product or services by displaying dynamic content contingent on the weather.
- Customize the creative to make the experience more personalized for your audience.

Use Case

A fast food chain wants to advertise products based on the current weather. For instance, if it's cold, show a hot cup of coffee, if it's hot, show a cold iced tea.

Included Files

Filename	Description
Basic Block	
DefaultImage500x250.gif	Default Banner Image
Weather_Basic_AS2.eba	Workshop file
Weather_Basic_DB.fla	Polite Format Banner file
Weather_Basic_RB.fla	Polite Format Rich Banner file
com/greensock/	Greensock Tweening Class
com/sizmek/weather/Config.as	Actionscript class file for Block settings
com/sizmek/weather/weatherBlock.as	Actionscript class for Block methods
Switch Block	
DefaultImage300x250.gif	Default Banner Image
Weather_Switch_AS2.eba	Workshop File
Weather_Switch_Asset_1.fla	Warm Weather Additional Asset
Weather_Switch_Asset_2.fla	Cold Weather Additional Asset
Weather_Switch_Asset_3.fla	Good Weather Additional Asset
Weather_Switch_Asset_4.fla	Bad Weather Additional Asset
Weather_Switch_Asset_5.fla	Default Weather Additional Asset
Weather_Switch_Asset_6.fla	Rain Additional Asset
Weather_Switch_Asset_7.fla	Thunderstorm Additional Asset
Weather_Switch_Asset_8.fla	Snow Additional Asset
com/sizmek/weather/Config.as	Actionscript class file for Block settings
com/sizmek/weather/weatherBlockSwitch.as	Actionscript class for Block methods

Implementing the Block

Before you Begin

Make sure you have the following resources available:

- Weather Block template

Note:

This block utilizes the HAMweather Aeris Weather API. You must set up an account with HAMweather and obtain a unique Consumer ID and Consumer secret. These will be used to connect to the Aeris API.

The Aeris Weather API have multiple available endpoints the allow for different responses, ie Forecasts, Tides, etc. For more information on the options available visit the HAMweather website.

<http://www.hamweather.com/support/documentation/aeris/>

The Weather Block utilizes the Aeris auto location method to detect the location of the audience, however this feature can be bypassed in the Config object under options.

Configuring in Flash

The Weather Block is accompanied by two templates:

Basic Weather Block – This template displays the current location, temperature, feels like and time.

Switch Weather Block – This template allows you to assess the weather on either the temperature or the weather type (ie rain/snow) and display creative specific to the results. For example, if it is warm, show a cold drink, if it is cold, show a warm drink.

Configuring the Basic Weather Block

The **Basic Weather Block** settings can be configured by editing the variable values in **Config.as** located in the com/sizmek/weather directory. Open the file in Flash and edit the variables as necessary. The Basic Weather Block requires very minimal editing to work. The **Config.as** file includes seven properties that can be edited, but only two are required to run a live campaign: The **WTHR_CLIENT_ID_STRING** and **WTHR_CLIENT_SECRET_STRING**.

Conf.as variables

Variable	Description
Initial Settings	
GEO_OVERRIDE_CITY:String	GEO_OVERRIDE_CITY is used if you do not wish to use the GEOLocation API. By entering a city name, you may manually override this functionality. Used in conjunction with GEO_OVERRIDE_STATE.
GEO_OVERRIDE_STATE:String	GEO_OVERRIDE_STATE is used if you do not wish to use the GEOLocation API. By entering a state name, you may manually override this functionality. Used in conjunction with GEO_OVERRIDE_CITY.
WTHR_ICON_SVR:String	WTHR_ICON_SVR establishes the location of the policy file on the server that is hosting your optional custom weather icons. ie. http://hosting.serving-sys.com/crossdomain.xml
WTHR_ICON_STRING:String	WTHR_ICON_STRING establishes the location of the weather icons used within the creative. If you wish to point to custom icons, enter the image directory here.
WTHR_CLIENT_ID_STRING:String	WTHR_CLIENT_ID_STRING should be updated with your HAMweather Aeris API client_id.
WTHR_CLIENT_SECRET_STRING:String	WTHR_CLIENT_SECRET_STRING should be updated with your HAMweather Aeris API client_secret.
WTHR_SVR_STRING:String	Weather API override. The Aeris Weather API uses multiple APIs endpoints, or data requests, to obtain the weather. By default, this block uses the observation endpoint.

Flash Implementation

1. In the Workshop™ for Flash open the '**Weather_Basic_AS2.eba**' or '**Weather_Basic_AS3.eba**' file using the **Open Existing Ad** option.
2. Open the **Config.as** file located in the com/sizmek/weather/ directory.
3. Enter **GEO_OVERRIDE_CITY** override location. Optional.
4. Enter **GEO_OVERRIDE_STATE** override location. Optional.
5. Enter **WTHR_ICON_STRING** directory link of custom images. Optional.
6. Enter your **WTHR_CLIENT_ID_STRING** from your HAMweather Aeris API account.
7. Enter your **WTHR_CLIENT_SECRET_STRING** from your Hamweather Aeris API account.

Configuring the Switch Weather Block

The **Switch Weather Block** settings can be configured by editing the variable values in **Config.as** which is located in the com/sizmek/weather directory. Open the file in Flash and edit the variables as necessary.

Conf.as variables

Variable	Description
Initial Settings	
GEO_OVERRIDE_CITY:String	GEO_OVERRIDE_CITY is used if you do not wish to use the GEOLocation API. By entering a city name, you may manually override this functionality. Used in conjunction with GEO_OVERRIDE_STATE.
GEO_OVERRIDE_STATE:String	GEO_OVERRIDE_STATE is used if you do not wish to use the GEOLocation API. By entering a state name, you may manually override this functionality. Used in conjunction with GEO_OVERRIDE_CITY.
WTHR_CLIENT_ID_STRING:String	WTHR_CLIENT_ID_STRING should be updated with your HAMweather Aeris API client_id.
WTHR_CLIENT_SECRET_STRING:String	WTHR_CLIENT_SECRET_STRING should be updated with your HAMweather Aeris API client_secret.
WTHR_SVR_STRING:String	Weather API override. The Aeris Weather API uses

Variable	Description
	multiple APIs endpoints, or data requests, to obtain the weather. By default, this block uses the observation endpoint.
WTHR_COND_SWTCH:Number	Determines condition on which the creative is chosen. 1 for temperature, 2 for type (sunny, rainy)
WTHR_CONT_SWTCH:Number	Determines creative type. 1 for frames, 2 for additional assets.
WTHR_TEMP_CF:String	Determines temperature measurement. 'F' for Fahrenheit, 'C' for Celsius.
WTHR_COND_TEMP:Number	Temperature Threshold, temperature at which creative is switched.
WTHR_COND:Array	Array that tracks weather codes. Add additional codes as needed, i.e. R = Rainy, S = Snow. Add will load unique asset or frame based on code.

Flash Implementation

1. In the Workshop™ for Flash open the '**Weather_Switch_AS2.eba**' or '**Weather_Switch_AS3.eba**' file using the **Open Existing Ad** option.
2. Open the **Config.as** file located in the com/sizmek/weather/ directory.
3. Enter **GEO_OVERRIDE_CITY** override location. Optional.
4. Enter **GEO_OVERRIDE_STATE** override location. Optional.
5. Enter your **WTHR_CLIENT_ID_STRING** from your HAMweather Aeris API account.
6. Enter your **WTHR_CLIENT_SECRET_STRING** from your Hamweather Aeris API account.
7. Edit **WTHR_COND_SWTCH**. This variable determines whether you want to evaluate the request based on the temperature or the weather type. If you want to switch your creative based on the temperature, choose 1. If you want to switch your creative based on the type of weather, ie sunny, rainy, choose 2.
8. Edit **WTHR_CONT_SWTCH**. This variable determines the type of creative you want to load. If you're going to place the content on separate frames, choose 1. If you'd rather load in additional assets, choose 2.
9. Edit **WTHR_TEMP_CF**. This variable determines whether you want to check the temperature in Fahrenheit or Celsius. Enter 'F' for Fahrenheit, or 'C' for Celsius. Only necessary if **WTHR_COND_SWTCH** is set to 1.

10. Edit **WTHR_COND_TEMP**. This variable determines the temperature threshold for the creative. Enter the temperature at which you want to show alternate weather. Only necessary if **WTHR_COND_SWTCH** is set to 1.
11. Edit **WTHR_COND**. If **WTHR_COND_SWTCH** was set to 2 and the content will switch depending on the type of weather. This array will hold the weather codes that you want to check the weather response for. For instance, 'R' equals rain, and 'S' equals snow. If the response sends the code 'R', we will load creative for Rain and if the response sends the code 'S', we will load creative for snow. Depending on what content type you chose, we will load a separate frame or additional asset will be loaded for each additional code entered here.
12. If frame content was chosen, you must create a new frame with the frame label equal to the code for each additional code added to the array. For instance, if you add 'R', for rain, to the array, then you must add a frame with the label 'R' to the 'mc_main' timeline and add your creative there.
13. If additional asset content was chosen, create a new additional asset for each additional code added to the array and make sure that the asset ID matches the array index, plus six. For instance, in the example below, Rain, or 'R' is the first item in the array and has an index of 0, so it would have an asset ID of 6.

For a full list of codes, please visit

<http://www.hamweather.com/support/documentation/aeris/codedweather/>

```
// Asset number should equal weather code array index - 5.
public static var WTHR_COND:Array = new Array ( 'R', 'T', 'S');
```

Custom Interactions

Interaction Name	Description
Warm Weather Frame	Block switching on temperature, weather response is warm.
Cold Weather Frame	Block switching on temperature, weather response is cold.
Good Weather Frame	Block switching on weather type, no code, weather response is clear.
Bad Weather Frame	Block switching on weather type, no code, weather response is not clear.
Default Frame	Block switching on weather type, no code, weather response is bad, default creative frame.

Interaction Name	Description
Rain Frame	Block switching on weather type, weather response is Rain.
Thunderstorm Frame	Block switching on weather type, weather response is Thunderstorm.
Snow Frame	Block switching on weather type, weather response is Snow.

HAMWeather AERIS Account Setup

The Weather Block utilizes the HAMweather AERIS Weather API. You must set up an account with HAMweather and obtain a unique Consumer ID and Consumer secret. These will be used to connect to the AERIS API. The Weather Block will not work until you have entered your Consumer ID and Consumer Secret into the Config object.

1. Visit <http://www.hamweather.com/account/member> and login or set up a new account if necessary.
2. Select API Application/Website Registration and enter a name for your Application Name, * for Application Domain/Identifier and leave Callback URL blank.

API Application/Website Registration

Complete the form to register a new application or domain to use the Aeris API. This will assign a unique client secret that identifies your application.

Application Details

Application Name

The name of your application. 32 characters max.

Application Domain/Identifier

*If the application is web-based, this value should be the application's publicly accessible top-level domain (e.g., mydomain.com or weather.mydomain.com). If the application is for mobile devices (iOS or Android), the domain value should be the package name conforming to the [Reverse DNS naming](#) convention (e.g., com.mydomain.MyProject). If you don't yet have a domain or package name and are testing on a local server, just use "localhost", but note that this **must** be changed to the correct value when used in your application to avoid affecting your service.*

***Multiple Namespaces** You can use multiple namespaces for a single application by providing a comma-separated list of namespaces (e.g., "test1.mydomain.com,test2.mydomain.com").*

Callback URL

The URL to be called when authenticating (not currently used in the API).

Organization Details

Organization Name

The organization or company developing this application, if any.

Organization Website

The web page for this application or the organization/company behind it, if any.

REGISTER

3. Click **Register**.

API Application/Website Registration

Your application was successfully registered!

Use the following access ID and secret when requesting data from the Aeris API for this application.

Consumer ID

yxn54qc6cWZjTtwxODifi

Consumer Secret

wuRlyZYlymtNgtkBiLaR5b5qPJsuEvTb4JA089sU

4. Save your **Consumer ID** and **Consumer Secret**. These are used in your Block.
5. Select **Add/Renew Subscription** and choose the right subscription level that best suits your needs.

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.

SizmekTM

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.