

flashTweets

User Guide revision 1.2
www.flashloaded.com

Help

This component is fully supported by the Flashloaded support team through our support forum. You will also find tips and additional information in the forum as well as announcements for version updates:

[flashTweets Support Forum](#)

Note: In order to post a question in the forum, you will need to [register](#) by creating a username and password. This registration differs from your account login.

Table of Contents

Installation	3
Getting started	4
Component Inspector parameters	6
XML	8
Refresh interval	9
Skinning	10
Crossdomain file	13
ActionScript methods	14
Help	15

Installation

You will need Adobe Extension Manager in order to install this component. Extension Manager should have been installed by default when you installed Flash. You may download the latest version of Extension Manager for free from the [Adobe website](#).

1. Ensure that Flash is closed before installing the flashTweets component.
2. Unzip/extract the flashTweets.zip file that you downloaded. You will find a file called flashTweets.mxp. Double click on this file in order to install the component using Extension Manager.

flashTweets should now be installed in your Flash Components Panel.

Getting started

1. Having installed flashTweets using the Adobe Extension Manager, start a new Flash ActionScript 3.0 file and save it.
2. Locate the **Flashloaded** folder in the Components panel and double click on it to expand it. You will find the **flashTweets** component inside this folder.
3. Drag and drop the **flashTweets** component onto the stage. The component initially has a default skin containing all of the fields. Please refer to the [skinning](#) section for instructions on skinning the component and removing information that you don't wish to display.
4. Click on the flashTweets component that's on the stage and open the Component Inspector panel (shift +F7).
5. Double click on the **Feeds** value in order to add the Twitter feeds that you wish to display. You can choose to display one or multiple feeds. Enter one Twitter name per line (this is the *@twitter* name, without the @ symbol). Press the + button to add each new feed. The order does not matter as the combined feeds will be displayed in chronological order. Press OK when done. The feeds can also be entered through an external XML file (instead of hard coded in the Component Inspector). Please see the [XML](#) section for instructions on using this option.

Note: You cannot display the feeds of accounts which have their updates protected.

6. Two sets of two proxy files, a crossdomain file and two empty folders are included with the component download - one set for servers that support PHP 4.0 or later (with support for CURL) and one for servers that support asp.net. Upload either of the following sets of proxy files to your webserver, depending on whether your server supports PHP or asp.net:

PHP:

flashtweetsproxy.php
imp.php
crossdomain.xml
cache (empty folder)
thumbnails (empty folder)

asp.net:

flashtweetsproxy.aspx
imp.aspx
crossdomain.xml
cache (empty folder)
thumbnails (empty folder)

The files should be uploaded to a location that is accessible through a browser (where you would normally upload HTML, php or JavaScript files). As a *crossdomain.xml* file is included for the exclusive usage of this component, we recommend creating a separate folder on your webserver for the proxy

files and this crossdomain.xml file. This will prevent overwriting your own crossdomain.xml file.

7. After uploading the files and folders in step 6, set the permissions on the **cache** and **thumbnails** folders to 777 (read, write access for everyone).
8. Enter the URL of the *flashtweetsproxy.php* or *flashtweetsproxy.aspx* proxy file that's on your server in the **Proxy file URL** parameter.

Note: If you choose to specify the full URL, you should have your own [crossdomain.xml](#) file on your server.

9. At this stage, you can already test flashTweets with the default parameters, to ensure that you have set it up correctly. Press Ctrl+Enter (win) or Cmnd+Enter (mac) to test your project.
10. You can change the various parameter settings in the Component Inspector to obtain the desired look and feel. Please see the [Component Inspector parameters](#) section for a description on each setting.

Component Inspector parameters

Parameter	Description	Example
Feeds	<p>The Twitter feeds that you wish to display. You can choose to display one or multiple feeds. Enter one Twitter name per line (this is the <i>@twitter</i> name, without the @ symbol). Press the + button to add each new feed. The order does not matter as the combined feeds will be displayed in chronological order. Press OK when done.</p> <p>The feeds can also be entered through an XML file using the Feeds XML parameter.</p>	designerdepot, mashable, techcrunch
Feeds XML	The optional XML file name containing the list of Twitter feeds to display. Please see the XML section for instructions on creating the XML file.	feeds.xml
Proxy file URL	<p>The URL of the flashtweetsproxy.php or flashtweetsproxy.aspx file that you uploaded to your webserver.</p> <p><i>Note: If you specify the full URL, you should have a crossdomain.xml file on your server.</i></p>	http://www.domain.com/flashtweetsproxy.php
Number of tweets	The total number of tweets to display when the component initializes. This total number for the combined feeds. If the <i>more</i> button is enabled, each press of the more button will add the same number of tweets to the list.	10
Autosize tweets	Sets whether to resize the height of the tweets automatically based on the length of their content or not.	true
Show more	Sets whether to display the more tweets button below the last tweet.	true
More cache	The number of sets of tweets to download at a time in cache. For example, setting this to 3 would allow for the more button to be pressed 3 times before having to make another request to the Twitter server for more content. See the refresh interval section for more information on caching.	3

Parameter	Description	Example
Refresh interval	The time interval (in minutes) to refresh the Twitter feeds. See the refresh interval section.	5
Auto scroll	If set to <i>true</i> , the list of tweets can be scrolled by moving the mouse up and down. <i>Note: This option will only work if Scrollbar is set to false.</i>	false
Scrollbar	Sets whether to display a scrollbar to scroll the list of tweets or not.	true
Spacing	The gap (in pixels) between each tweet.	5
Link	The color of links that appear in tweets.	#0000FF
Link Over	The roll over color for links.	#0099CC
Link Visited	The visited color for links.	#993366
Link Over Underline	Sets whether to underline the links on mouse over or not.	true
Link Bold	Sets whether to display the links in bold or not. <i>Note: When using this option, if you are using embedded fonts in the tweet_content textfield, the bold style of the font must also be embedded in the library of your fla and set to "Export for ActionScript" and "Export in First Frame".</i>	true

XML

The list of feeds that you wish to display in the flashTweets component can be specified using an XML file. By defining the feeds in an external XML file, you can publish the SWF file once and change the feeds whenever you wish.

1. Open your favorite plain text editor (for example Notepad on Windows or TextEdit on Mac) and start a new file. *Note: If you are using TextEdit on Mac, choose Format > Make Plain Text*
2. Begin your file with the following line:

```
<?xml version="1.0"?>
```

This is the standard xml declaration.

3. Add the following lines to your xml file (the bold lines are the new additions)

```
<?xml version="1.0"?>  
<feeds>  
</feeds>
```

4. Add the feed tags to your XML file. Each feed name is defined using between the <feed></feed> tags (the bold lines are the new additions).

```
<?xml version="1.0"?>  
<feeds>  
  <feed>mashable</feed>  
  <feed>designerdepot</feed>  
  <feed>techcrunch</feed>  
</feeds>
```

5. Save the XML file to the same folder as your Flash file. In this example, we have given the XML file the name: *feeds.xml*

6. Return to your Flash file. Enter the name and path to the XML file that you just created in the **Feeds XML** parameter of the flashTweets that's on the stage.

Note: If your .swf file will be in a different folder to the HTML file in which it is embedded, you should enter the path to the XML file, relative to the location of the .html file.

7. Press Ctrl+Enter (Win) or Cmd+Enter (Mac) to test your movie.

Refresh interval

Twitter currently allows a maximum of 150 requests per hour. One request is made for each feed at every refresh interval.

For example, if you have set the component to display 10 feeds, with a refresh interval of 5 minutes, you will be making 120 requests per hour.

Note that clicking on an @twitter account name for a feed that is not loaded in the component (i.e. an account name is that is included in the content of a tweet), will count as an additional request.

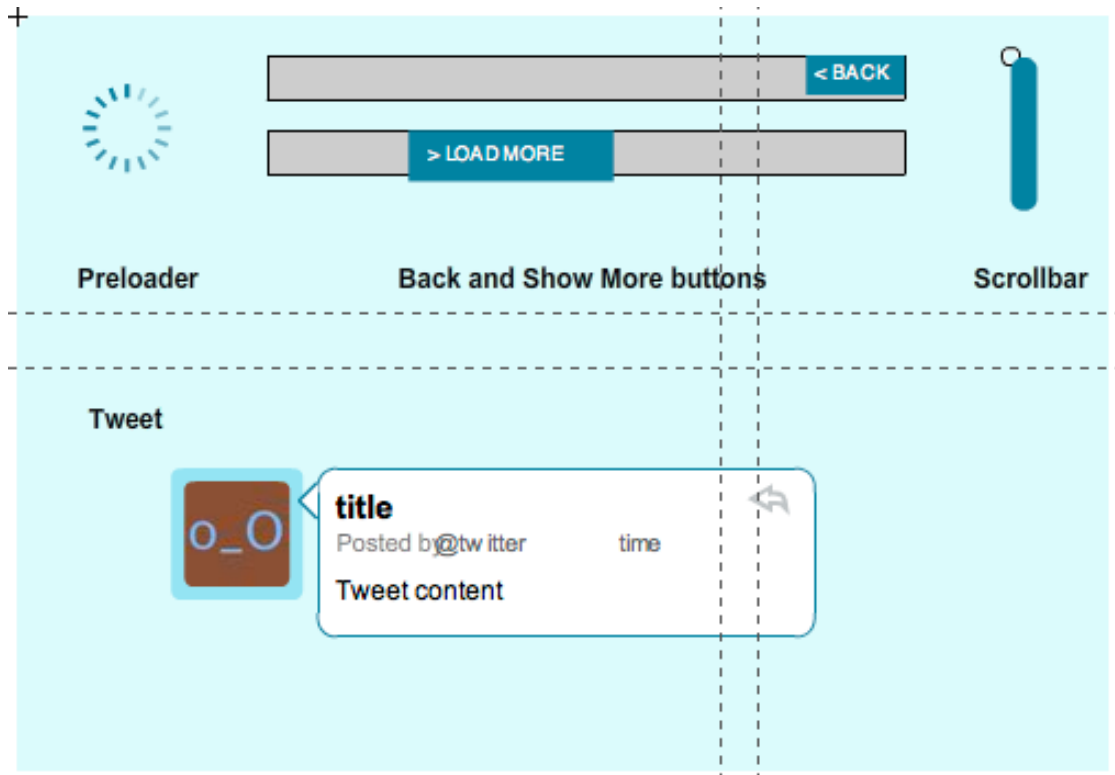
You can set the **more cache** parameter to change the number of times that the more button can be pressed without the server having to make another request. Setting this to a lower number will cause the data to load quicker, while setting it too high may cause excessive load time. We recommend setting this to a value of 3.

The proxy files automatically create cache files of the data from Twitter, to prevent excessive requests made to the server. The last 24 hours of cache files will be stored on your server.

Skinning

The entire look of the way in which the tweets are displayed in flashTweeter can be changed to suit your own design requirements. You can also delete information or elements that you do not wish to display.

To start skinning, double click on the flashTweets component that's on the stage. You will now see all of the elements that make up the display of this component. They are all located in a layer called "skins". Ensure that this layer is unlocked in order to edit any of these elements.



Skinning the appearance of tweets

The dynamic textfields in the default skin do not use embedded fonts. If you are using a font that is not installed on all computers, you should choose to embed the font in the textfield properties. Double click on the Tweet movie clip. Inside this movie clip you will find the following elements:

tweetTitleBtn (button)

This is the name of the Twitter feed that the tweet is from. Double click on this button in order to edit the textfield inside to change the font, size and color. You will see two frames, Up and Over, in which you can change the look of the textfield for the mouse up and mouse over positions.

Note: You may wish to delete the tweetTitleBtn button from the skin if you are only displaying feeds from one Twitter account and have no need to differentiate between them.

twitterLinkBtn (button)

This is the Twitter account name that the tweet is from. Double click on this button in order to edit the textfield inside to change the font, size and color. You will see two frames, Up and Over, in which you can change the look of the textfield for the mouse up and mouse over positions.

Note: You may wish to delete the twitterLinkBtn button from the skin if you are only displaying feeds from one Twitter account and have no need to differentiate between them.

tweet_content (textfield)

This textfield contains the content of the tweets. You can change all of the properties of the textfield as well as the start position and width. It will expand in height automatically based on the length of the tweets.

tweet_time (textfield)

This textfield contains the time that the tweet was posted. You can change all of the properties of the textfield as well as the position.

retweet_icon (button)

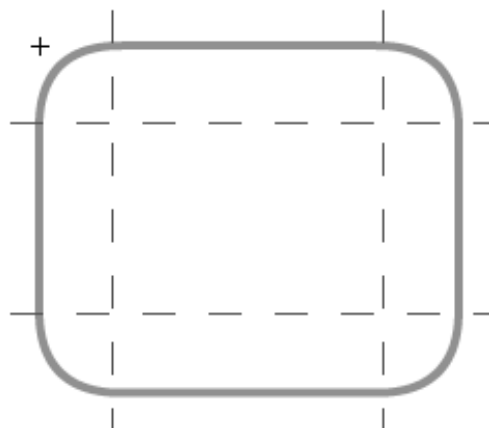
This button contain the graphic for the retweet button. Double click on this button in order to edit the look or color of the icon. You will see two frames, Up and Over, in which you can change the look of the graphic for the mouse up and mouse over positions.

ProfileImage (movie clip)

This movie clip contains the profile image (Twitter account avatar) movie clip and the background that appears behind the profile image. The profile image can also be masked inside this movie clip. You may wish to delete this movie clip from the skin if you are only displaying feeds from one Twitter account.

tweet_bg (movie clip)

This movie clip contains the background that appears behind each tweet. To edit it, start by double clicking on the tweet_bg movie clip. You will see two frames on the timeline. Frame 1 is for the mouse up position and frame 2 is for the mouse over position. You will see a grid, indicating that this is a 9-slice movie clip. This means that whatever is in the 4 corners will not resize and the pieces between the corners and the middle will resize, depending on the length of the tweet.



tweet_arrow (movie clip)

This movie clip contains the arrow which points the tweet to the profile image. Double click on it to edit the look. It will automatically be positioned towards the top of each tweet.

Skinning the back (show all) and more buttons

To change the look or horizontal position of the *back (show all)* or *more* buttons, double click on the **showallPositionMc** or **morePositionMc** movie clips respectively. You will see the show all or more buttons as well as a background movie clip called widthLineMc. This background movie clip is used as a guide only in order to set the horizontal position of the show all and more buttons. Set this movie clip to match width of your feeds and position the show all and more buttons on top of it accordingly.

Double click on the showall_btn or more_btn buttons in order to edit the textfield inside them to change the font, size and color. You will see two frames, Up and Over, in which you can change the look of the textfield for the mouse up and mouse over positions.

Skinning the scrollbar

The scrollbar consists of 2 movie clips - **scrollTrack** and **dragBar**. Double click on each of these movie clips to edit the look of the scroll track or drag bar. Both of these movie clips use 9-slice scaling. This means that whatever is in the 4 corners will not resize and the pieces between the corners and the middle will resize, depending on the length of the scrollbar.

Skinning the preloader

To change the look of the preloader, double click on the preloader_mc movie clip. You can replace the contents of this movie clip with other animation or textfield.

Crossdomain file

The Flash Player includes an additional security feature that does not allow scripts to be read across domains. It also does not recognize two URLs for the same site, with and without "www", as being on the same domain. For example, *http://www.yourdomain.com* or not recognized as being the same as *http://yourdomain.com*

If you chose to specify the full URL to the proxy file, for example *http://www.yourdomain.com/flashtweetsproxy.php*, then you would need a *crossdomain.xml* policy file on your server. Without it, visitors who enter your site without the "www" in the URL will not see the feeds. You might already have a *crossdomain.xml* file on your server.

This is how you would create a *crossdomain.xml* file:

1. Open a new document in a text editor, such as Notepad on Windows or TextEdit on Mac (if you are using TextEdit, select Format > Make Plain Text).
2. Paste the following code into the document:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.macromedia.com/xml/dtds/
cross-domain-policy.dtd">
  <cross-domain-policy>
    <allow-access-from domain="yourdomain.com" />
    <allow-access-from domain="www.yourdomain.com" />
  </cross-domain-policy>
```

3. Replace "yourdomain.com" with your own domain (both with and without the www).
4. Save this file under the name: ***crossdomain.xml***
5. Upload this file to the root of your webserver (usually the same place where the HTML files reside).

For more information on the cross-domain policy file, please see the following Adobe article: http://kb2.adobe.com/cps/142/tn_14213.html

ActionScript methods

The following ActionScript methods are available:

LoadXml

Availability

Flash Player 9

Description

Method; sets the XML file to load into the flashTweets component.

Example

```
flashTweetsInstance.LoadXML("feeds.xml");
```

ResetFilter

Availability

Flash Player 9

Description

Method; calls the function of the back button in the component.

Example

```
flashTweetsInstance.ResetFiler();
```

ShowMore

Availability

Flash Player 9

Description

Method; calls the function of the "more" button in the component.

Example

```
flashTweetsInstance.ShowMore();
```

Help

This component is fully supported by the Flashloaded support team through our support forum. You will also find tips and additional information in the forum as well as announcements for version updates:
[flashTweets Support Forum](#)

Note: In order to post a question in the forum, you will need to [register](#) by creating a username and password. This registration differs from your account login.