

Twitter Integration

Created by Unknown User (malynow), last modified by Anton Lashuk on янв 06, 2019

- Installation
 - Installing Twitter Integration extension
 - Registering application on Twitter
- Configuration
 - Twitter Connect
 - General Options
 - Content Options
 - Twitter Follow Button
 - General Options
 - Content Options
 - Twitter Tweet Button
 - General Options
 - Content Options
- Customer Account section
- Custom positions
 - Layout Update
 - CMS Pages
- Uninstallation
 - Troubleshooting

Extension page: <http://ecommerce.aheadworks.com/magento-extensions/twitter-integration.html>

Installation

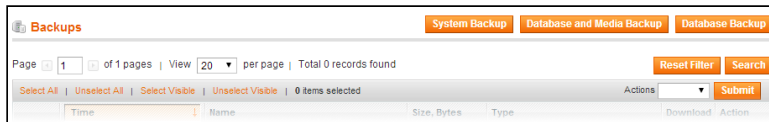
Installing Twitter Integration extension

1. Backup your web directory and store database.

[Click to view details](#)

You can make backup copies with any tool you find appropriate

If you are going to use the native Magento backup function, navigate to **System -> Tools -> Backups** and perform **System** and **Database** backups



2. Log in to the Magento backend

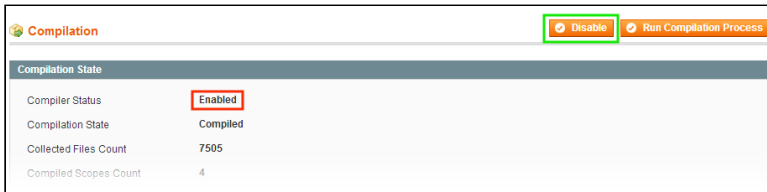
3. Disable compilation

[Click to view details](#)

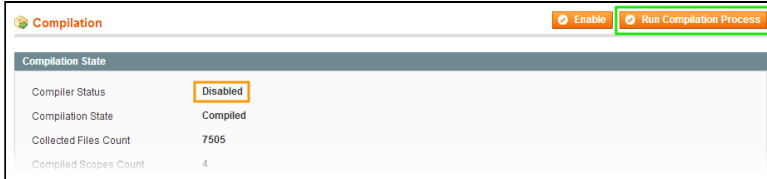
Navigate to **System -> Tools -> Compilation**.

If Compiler status is Disabled, you can skip to the next step

If Compiler is enabled, disable it.



IMPORTANT: after the extension is installed, you can enable the compilation again; **IT IS CRUCIAL** that you use "Run Compilation Process" function, not just "Enable button"



Installing an extension with the Compilation enabled will result in store downtime.

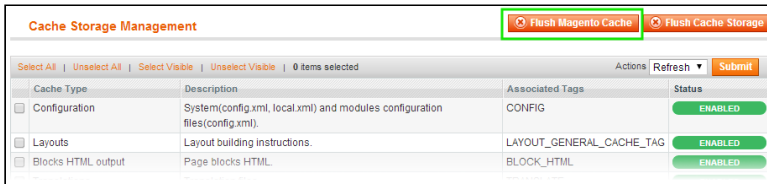
4. Flush store cache

Click to view details

You can flush the store cache in 2 ways:

- **via the backend:**

Navigate to **System -> Cache Management** menu, and click **Flush Magento Cache** button



- **via filesystem:**

On your server, navigate to Magento root folder, then proceed to **/var/cache/**; delete all the content there.

5. Download the extension package from your account and extract the downloaded archive

6. Copy the content of **/Step_1/** folder to your store's root directory

7. Copy the content of **/Step_2/** folder to your store's root directory

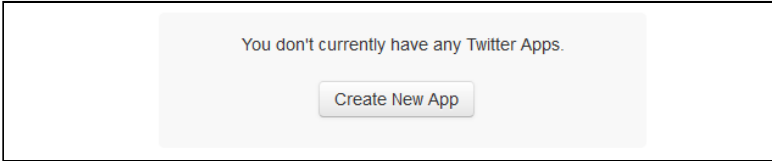
8. Flush store cache again; log out from the backend and log in again.

Registering application on Twitter

To set up the extension navigate to the **Twitter Apps** page. Here it will be required to create a new Twitter App for later use with the Twitter

Integration extension.

1. Add new App;
Click to view details
Click **Create New App** button.



2. Fill in application details;
Click to view details
Name the application, add description and specify the website and Callback URL you want to use the Twitter Integration extension at.

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Once application details specified you will be suggested to read and accept Developer Agreement.

After you are done with the agreement, click **Create your Twitter application** button.

Your application has been created. Please take a moment to review and adjust your application's settings.

3. Configure the app;
Click to view details
Under Settings tab tick the **Allow this application to be used to Sign in with Twitter** checkbox.

Allow this application to be used to [Sign in with Twitter](#)


Proceed to the Permissions tab of the application settings and select **Read, Write and Access direct messages** radio button.

Details Settings Keys and Access Tokens Permissions

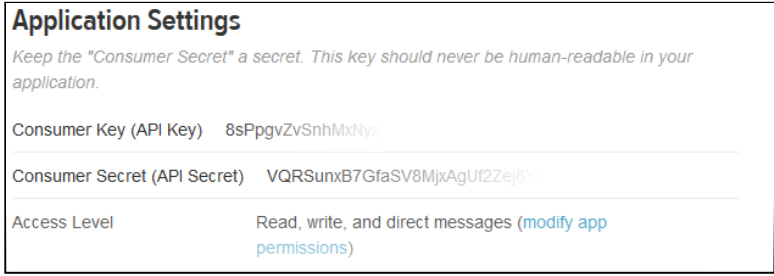
Access

What type of access does your application need?
[Read more about our Application Permission Model.](#)

Read only
 Read and Write
 Read, Write and Access direct messages

At the bottom of the screen click  button.

4. Get Application Consumer Key and Consumer Secret;
[Click to view details](#)
 Under the **Keys and Access Tokens** copy **Consumer Key** and **Consumer secret** and paste these into appropriate fields of the extension's settings in your store backend.



Configuration

All the extension configuration options can be found under **System > Configuration > AheadWorks extensions**.

As the extension introduces several aspects of the Twitter integration functionality all settings are splint into three (3) sections:

- Twitter Connect;
- Twitter Follow Button;
- Twitter Tweet Button.

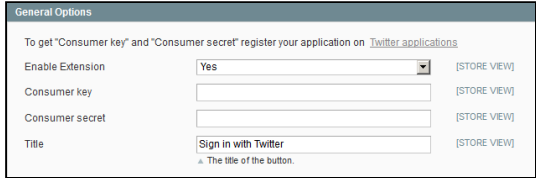
Twitter Connect

Twitter Connect settings section allows managing basic extension configuration.

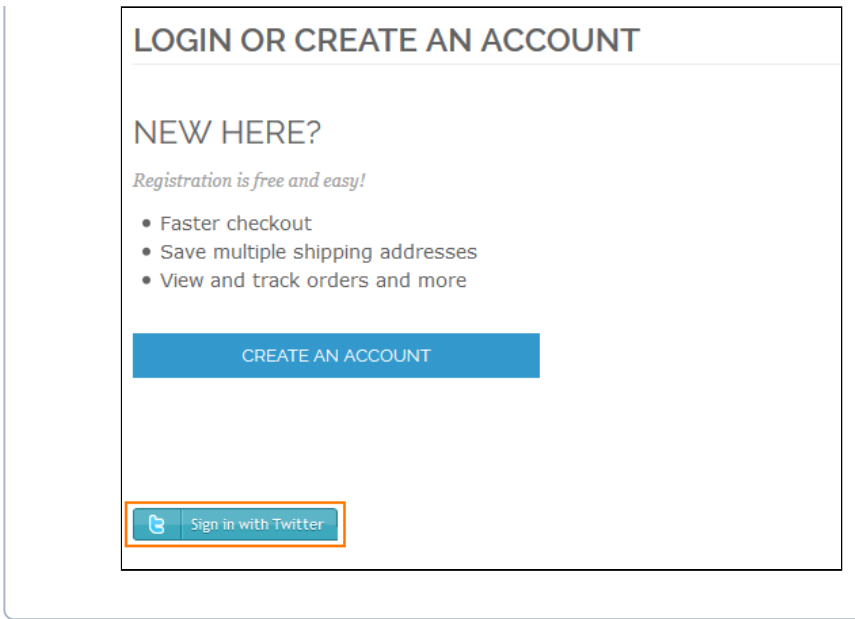
General Options

In the **Twitter Connect** settings section one is suggested to specify the Twitter Consumer Key and Consumer Secret.

- **Enable Extension** - allows enabling/disabling extension output;
- **Consumer key** - Twitter Consumer Key;
- **Consumer secret** - Twitter Consumer secret;
- **Title** - defines the title of the Twitter **Sign In** button.

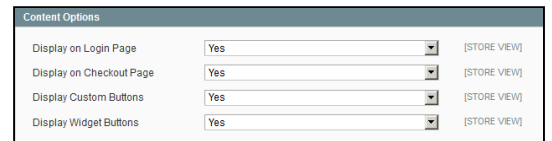


How does Twitter sign in button look like?



Content Options

- **Display on Login Page** - defines if the Twitter **Sign In** button should be displayed at the Login page;
- **Display on Checkout Page** - defines if the Twitter **Sign In** button should be displayed at the Checkout page;
- **Display Custom Buttons** - allows disabling/enabling all custom buttons placed via **CMS > Widget**;
- **Display Widget Buttons** - allows disabling/enabling **ALL** Twitter Integration related buttons.

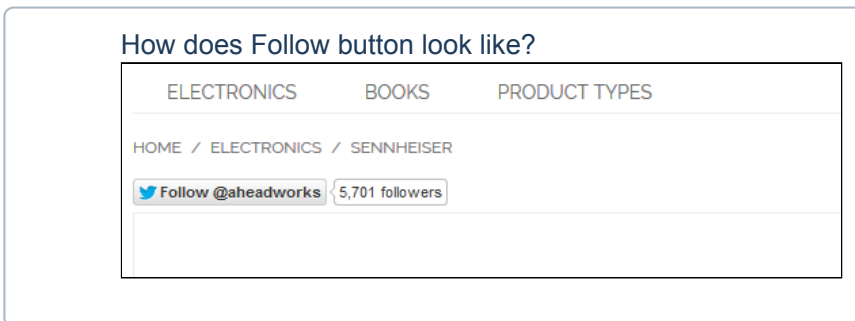
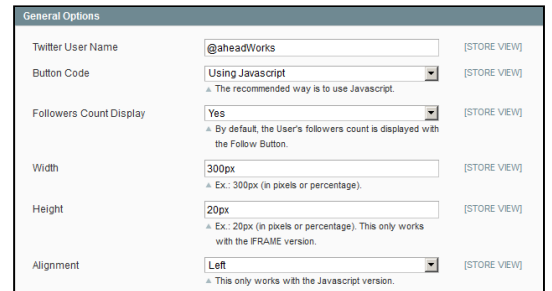


Twitter Follow Button

Twitter Follow Button settings allow managing **Follow** button related settings.

General Options

- **Twitter User Name** - defines the store Twitter user name;
- **Button Code** - defines the code that should be used for the **Follow** button functionality (leave the default value if you are not sure if you want to change it);
- **Followers Count Display** - defines if the number of followers should be displayed along with the **Follow** button;



- **Width - Follow** button block width in pixels or %;
- **Height - Follow** button block height in pixels or %;
- **Alignment** - defines the Follow button position:
 - Left;
 - Right.

Content Options

- **Display on CMS pages** - defines if the Twitter **Follow** button should be displayed at the CMS pages;
- **Display on Home Page** - defines if the Twitter **Follow** button should be displayed at the Home page;
- **Display on categories page** - defines if the Twitter **Follow** button should be displayed at the category pages;
- **Display on products page** - defines if the Twitter **Follow** button should be displayed at the product pages;
- **Display Custom Buttons** - allows disabling/enabling all custom buttons placed via **CMS > Widget**;
- **Display Widget Buttons** - allows disabling/enabling **ALL** Twitter Integration related buttons.

Content Options		
Display on CMS pages	<input type="text" value="No"/>	[STORE VIEW]
Display on Home page	<input type="text" value="No"/>	[STORE VIEW]
Display on categories page	<input type="text" value="No"/>	[STORE VIEW]
Display on products page	<input type="text" value="No"/>	[STORE VIEW]
Display Custom Buttons	<input type="text" value="No"/>	[STORE VIEW]
Display Widget Buttons	<input type="text" value="No"/>	[STORE VIEW]

Twitter Tweet Button

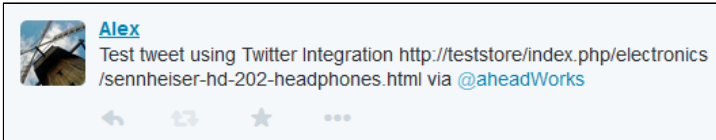
Twitter Tweet Button settings allows managing **Tweet** button related settings.

General Options

- **Button Code** - defines the code that should be used for the **Follow** button functionality (leave the default value if you are not sure if you want to change it);
- **Twitter User Name** - defines the store Twitter user name;
- **Tweet text** - allows editing the text to be displayed in the customer's tweet;
- **Recommended accounts** - allows suggesting other Twitter accounts for customers to follow;
- **Count box position** - defines the position of the count box and if it should be displayed:
 - None;
 - Horizontal;
 - Vertical;
- **The URL to which your shared URL resolves to** - defines the URL that will be tweet related e.g. the store home page;
- **Width** - **Tweet** button block width in pixels or %;
- **Height** - **Tweet** button block height in pixels or %;

General Options		
Button Code	<input type="text" value="Using Javascript"/> <small>▲ The recommended way is to use Javascript.</small>	[STORE VIEW]
Twitter User Name	<input type="text" value="aheadWorks"/>	[STORE VIEW]
Tweet text	<input type="text"/>	[STORE VIEW]
Recommended accounts	<input type="text"/> <small>▲ Using this field you can suggest accounts for a user to follow once they have sent a Tweet using your Tweet Button. These suggested accounts and their basic information are shown on the last page of the Share Box flow. You can provide multiple related accounts by comma separating entries in the field. Ex.: anywhere:The Javascript APIs:streams.twitter:The official account</small>	[STORE VIEW]
Count box position	<input type="text" value="Horizontal"/>	[STORE VIEW]
The URL to which your shared URL resolves to	<input type="text"/>	[STORE VIEW]
Width	<input type="text" value="130px"/> <small>▲ Ex.: 130px (in pixels or percentage). This only works with the #FRAME version.</small>	[STORE VIEW]
Height	<input type="text" value="50px"/> <small>▲ Ex.: 50px (in pixels or percentage). This only works with the #FRAME version.</small>	[STORE VIEW]

How does the tweet look like?



Content Options

- **Display on CMS pages** - defines if the Twitter **Tweet** button should be displayed at the CMS pages;
- **Display on Home Page** - defines if the Twitter **Tweet** button should be displayed at the Home page;
- **Display on categories page** - defines if the Twitter **Tweet** button should be displayed at the category pages;
- **Display on products page** - defines if the Twitter **Tweet** button should be displayed at the product pages;
- **Display on Checkout Success page** - defines if the Twitter **Tweet** button should be displayed at the category pages;
- **Display Custom Buttons** - allows disabling/enabling all custom buttons placed via **CMS > Widget**;

Content Options		
Display on CMS pages	<input type="text" value="No"/>	[STORE VIEW]
Display on Home page	<input type="text" value="No"/>	[STORE VIEW]
Display on categories page	<input type="text" value="No"/>	[STORE VIEW]
Display on products page	<input type="text" value="No"/>	[STORE VIEW]
Display on Checkout Success page	<input type="text" value="No"/>	[STORE VIEW]
Display Custom Buttons	<input type="text" value="No"/>	[STORE VIEW]
Display Widget Buttons	<input type="text" value="No"/>	[STORE VIEW]

- **Display Widget Buttons** - allows disabling/enabling **ALL** Twitter Integration related buttons.

Customer Account section

Twitter Integration extension introduces new customer account section **Twitter Connect**.

The section allows customer decide if he wants the orders to be tweeted.

Post order details to Twitter setting is enabled by default.

Custom positions

Twitter Integration extension allows placing it's buttons and blocks using widget **Layout Updates** or **CMS pages** widgets.

The extension comes with the following widgets available:

- Twitter "Follow" Button;
- Twitter "Tweet" Button;
- Twitter Connect Button.

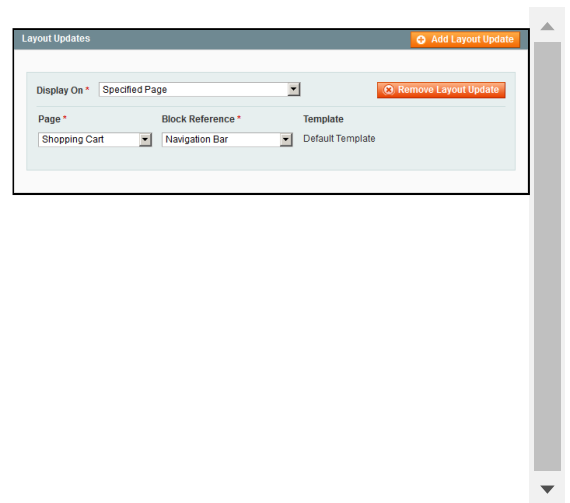
Each widget represents related functional aspect of the extension.

Layout Update

Custom Twitter button widgets can be introduced to different store pages from **CMS > Widget** section.

1. Click **Add New Widget Instance**;
2. Specify required Twitter widget in the **Type** dropdown list;
3. Specify required **Theme** you want the widget to be displayed at;
4. Click **Continue** button;
5. Set required widget parameters in the Frontend Properties section;
6. In the Layout Updates section by clicking **Add Layout Update** button specify the store pages you want the widget to be displayed at.

Once the widget is saved required Twitter widget will be displayed at the set store pages.

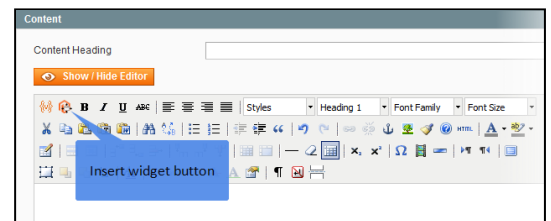


CMS Pages

Twitter Integration widgets can be inserted in the CMS pages from **CMS > Pages > Manage Content** section.

1. Select existing or create new CMS page;
2. In the Content block click **Insert Widget** button of the WYSIWYG editor;
3. Select required Twitter Integration widget.

The widget will be automatically inserted in the CMS page.



Uninstallation

1. Disable compilation, in case it is enabled.
2. Login to your FTP, navigate to **app/etc/modules/**
3. Open the files: AW_Twitter_Connect.xml, AW_Twitter_Follow.xml, AW_Twitter_Tweet.xml and change the following line:

```
<active>>true</active>
```

to

```
<active>>false</active>
```

Now your Magento is unaware of the existence of the extension

4. Clear the cache under var/cache
5. Make sure that the site is working properly, otherwise roll back the changes and apply to our technical support.
6. If everything works fine, it is safe to delete the files of the extension.
7. In case you need to clean the database, **make a backup** and then run the following query in MySQL:

```
DROP TABLE 'twitterconnect_synch';
```

Note, if you are using the table with prefixes, you must specify them in all table names.

If you are not sure how to do that or expect any troubles with it, please contact your server administrator regarding the matter.

If you remove the tables as it is described above, you will need to configure the extension again after it is reinstalled.

Troubleshooting

After the extension installation the store gives an error, or blank page, or suggests to start Magento installation procedure.

Change the owner of the extracted extension files to the web server user and set 775 permissions on them. Clear the store cache and try again.

There is no *aheadWorks extensions* under my configuration section, or having the extension tab clicked I get a blank page, or *Access Denied* error.

Clear the store cache, browser cookies, logout and login again.

I've set up everything correctly, inserted the HTML code but there is nothing on that page.

Clear the store cache, clear your browser cache and domain cookies and refresh the page.

My configuration changes do not appear on the store.

Clear the store cache, clear your browser cache and domain cookies and refresh the page.

You can always find the latest version of the software, full documentation, demos, screenshots and reviews at <http://ecommerce.aheadworks.com>

License agreement: <http://ecommerce.aheadworks.com/LICENSE-M1.txt>

Contact us: <http://ecommerce.aheadworks.com/contacts/>

Copyright © 2019 aheadWorks Co. <http://www.aheadworks.com>

Нравится
это

Станьте первыми кому понравится

Ни одной