

Blog

Created by Unknown User (malynow), last modified by Anton Lashuk on янв 05, 2019

- Installation
 - Upgrading from lazzymonk's original blog
 - Upgrading from 1.0.2, 1.0.1 to 1.0.3 and higher
- Migrating to Blog for Magento 2
 - What one should know before migrating Aheadworks Blog for M1 to Blog for M2?
 - Video guide
 - Migration process
- Configuration
 - Blog Settings
 - Menus and Links
 - Comments
 - ReCAPTCHA
- Managing the Blog
 - Adding new post
 - Post Information
 - Advanced Options
 - Categories
- Using Blog widget
- Uninstallation
- Troubleshooting

Having hard time finding an answer to your question?
Check out our Knowledge Base.

Extension pages:

Community Edition (Free): <http://ecommerce.aheadworks.com/free-stuff/blog.html>

Enterprise edition: <http://ecommerce.aheadworks.com/magento-extensions/blog-enterprise-edition.html>

The Blog solution for Magento. It doesn't require any 3rd party blogging software.

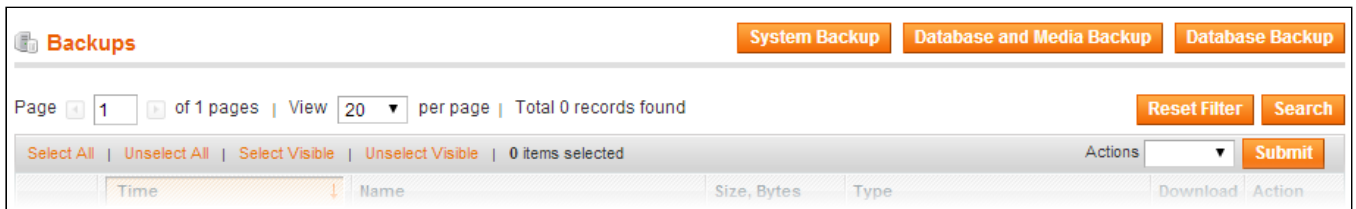
Installation

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.

The screenshot shows the 'Compilation' section with a 'Disable' button highlighted in green. The 'Compilation State' table is as follows:

Compilation State	
Compiler Status	Enabled
Compilation State	Compiled
Collected Files Count	7505
Compiled Scopes Count	4

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"

The screenshot shows the 'Compilation' section with 'Compiler Status' set to 'Disabled' and the 'Run Compilation Process' button highlighted in green. The 'Compilation State' table is as follows:

Compilation State	
Compiler Status	Disabled
Compilation State	Compiled
Collected Files Count	7505
Compiled Scopes Count	4

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

The screenshot shows the 'Cache Storage Management' interface with the 'Flush Magento Cache' button highlighted in green. Below the buttons is a table with the following data:

Cache Type	Description	Associated Tags	Status
<input type="checkbox"/> Configuration	System(config.xml, local.xml) and modules configuration files(config.xml).	CONFIG	ENABLED
<input type="checkbox"/> Layouts	Layout building instructions.	LAYOUT_GENERAL_CACHE_TAG	ENABLED
<input type="checkbox"/> Blocks HTML output	Page blocks HTML.	BLOCK_HTML	ENABLED

- **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

If you are using a custom theme,
read the instructions here.

In case you are using a custom theme, it is recommended to copy the design files to your current theme's folders. In case there are several themes in use at the same store, the design files must be copied to each of them.

- Find this folder in the package: `/step_1/app/design/frontend/base/default/` ; copy its content to `/app/design/frontend/[your_package]/[your_theme]/`

- Find this folder in the package: `/step_1/skin/frontend/base/default/`; copy its content to `/skin/frontend/[your_package]/[your_theme]/`

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.

Upgrading from lazzymonk's original blog

AW_Blog is designed to replace Monk_Blog. So if Monk_Blog is installed on your store and you want to upgrade, you should do the following:

1. Disable Monk_Blog via editing `app/etc/modules/Monk_Blog.xml`: change `true` to `false`:

```
<?xml version="1.0"?>
<config>
  <modules>
    <Monk_Blog>
      <active>true</active>
      <codePool>community</codePool>
    </Monk_Blog>
  </modules>
</config>

<?xml version="1.0"?>
<config>
  <modules>
    <Monk_Blog>
      <active>>false</active>
      <codePool>community</codePool>
    </Monk_Blog>
  </modules>
</config>
```

2. Install the AW_Blog extension in a usual way.

Upgrading from 1.0.2, 1.0.1 to 1.0.3 and higher

If you have had troubles while installing 1.0.1 or 1.0.2, please delete `app/code/community/AW/Blog/sql/blog_setup/mysql4-install-0.2.5.php` before continuing.

Migrating to Blog for Magento 2

What one should know before migrating Aheadworks Blog for M1 to Blog for M2?

Read carefully and consider this information of top priority as otherwise it may result in an accident data loss.

1. **Magento Data Migration Tool is required to be installed prior to starting migration.**
The migration script relies heavily on Magento Data Migration tool. Understanding how it works will prove useful during migration process.

2. **Backup Magento 2 databases before migration to have an opportunity to revert any changes.**
Blog migration may involve rewriting database tables, backup is advised.
3. **Blog post images cannot be migrated automatically and should be migrated manually for best results.**
More information on migrating images can be found in the [Magento instructions on manual migration](#).
4. **All widget references incusing CMS blocks should be migrated manually.**
5. **Pay close attention to <aw_blog_updates> tag value you enter if you already have posts on your Magento 2 store and want to keep them.**
"0" value set by default will overwrite all existing Magento 2 Blog posts, "1" value will add new post entries to the existing ones.

Video guide

Migration process is highlighted in the education **Video Guide** at [Aheadworks Youtube channel](#). However, we still advise to read through the instructions below to make sure you haven't missed anything.

Don't forget to enable subtitles in the video to follow the data migration process.

Migration process

Down below is a written migration guide with more detailed steps description.

1. Log into your Magento server as file owner and navigate to the directory you have your Magento 2 store installed
[Click to view details](#)
For example:

```
cd /var/www/my_magento2_store/
```

2. Install **Magento Data Migration Tool**

[Click to view details](#)

Installation instructions for Magento Data Migration Tool along with basic information can be found here:
<http://devdocs.magento.com/guides/v2.0/migration/migration-tool-install.html>

Magento provides two ways of installing Magento Data Migration Tool:

- Install the tool from GitHub
- Install the tool from repo.magento.com

For the demo video purposes we have been installing Data Migration Tool from GitHub.

After you may want to navigate to `vendor/magento/` to make sure the Migration Tool has been installed successfully.

3. **Download Blog migration distributive** and unpack it to the desired location

[Click to view details](#)

Blog migration distributive can be downloaded by the link here:
https://s3.amazonaws.com/aheadworks_cdn/Blog/data-migration-tool.zip

Unpack the contents of the distributive to find `/etc` and `/scr` folders inside.

4. Copy the contents of Blog migration distributive to Magento Data Migration Tool main folder

[Click to view details](#)

Copy `/etc` and `/scr` folders found in the Blog migration package to: `vendor/magento/data-migration-tool/`
Rewrite on prompt.

5. Edit `config.xml` file of the Magento Data Migration Tool

[Click to view details](#)

Navigate to **vendor/magento/data-migration-tool/etc/aw_blog/** and open **config.xml** for edit. More information on the tag values and what one should put inside can be found in the **Magento Data Migration Tool configuration instructions**.

Pay close attention to <aw_blog_updates> tag value you enter if you already have posts on your Magento 2 store and want to keep them. "0" value set by default will overwrite all existing Magento 2 Blog posts, "1" value will add new post entries to the existing ones.

6. Run migration script in the console

[Click to view details](#)

Run the following command considering path to the **config.xml** to start migration:

```
bin/magento migrate:data [-r|--reset] {<path to config.xml>}
```

In the [demo video](#) we are using the following command considering path to config.xml:

```
bin/magento migrate:data -r vendor/magento/data-migration-tool/etc/aw_blog/config.xml
```

7. Data migration is complete. Make sure to migrate images and widget related items manually

Configuration

General Configuration Options can be found under **System > Configuration > AheadWorks Extensions > Blog**

Blog Settings

- **Enable** - defines if the Blog is enabled or disabled;
- **Route to Blog** - allows defining the route to the blog: `yourstore/route_to_blog`;

Route to Blog must NOT contain any slashes.

This is a correct route:

Route to Blog [STORE VIEW]

This is an incorrect route:

Route to Blog [STORE VIEW]

*Please note: Blog will not be available if incorrect **Route to Blog** is specified.*

- **Blog Title** - allows defining the blog title;
- **Meta Keywords** - specify the meta keywords for a precise appeal to the the search engines;
- **Meta Description** - specify the description of the blog posts for better SERP (search engine results page) appeal;
- **Image** - since 1.4.0 version Blog supports Open Graph tag. 'Image' field allows uploading an image that will be used as a thumbnail when sharing on social networks. Image min size 200x200, max 1600x1600. Image aspect should be 3:1, otherwise it may be displayed improperly. If no image uploaded

store default logo will be used instead.

- **Page Layout** - defines the blog pages layout (same terms as for Magento default layout);
- **Date Format** - defines the date format to be displayed under the blog posts:
 - Short;
 - Medium;
 - Long;
 - Full;
- **Show Breadcrumbs** - defines if the breadcrumbs should be displayed (useful for navigation);
- **Summary Character Count** - defines the number of characters in a post to be displayed before it is truncated and the **Read More** link is added (0 for unlimited);
- **Use Short Content Field** - defines if the **Short Content** field (if filled) should be used instead of truncating the post;
- **Parse CMS Tags in Post Contents** - defines if the CMS Tags (if used) should be parsed into the post's body;
- **Posts per Page** - defines how many blog posts should be displayed at a single blog page;
- **Use Categories ID in Posts URLs** - defines if the Categories ID should be included in the posts's URLs;
- **Show Bookmarks in Lists** - defines if the bookmarks (Digg.com, Stumbleupon.com, Del.icio.us, Facebook, Reddit.com) should be displayed at the blog page;
- **Show Bookmarks in Posts** - defines if the bookmarks (Digg.com, Stumbleupon.com, Del.icio.us, Facebook, Reddit.com) should be displayed at the post page;
- **Default Posts Sorting** - defines the way the blog posts should be sorted:
 - Newest first;
 - Oldest first.

Menus and Links

- **Enable left Menu** - defines if the Blog menu should be introduced to the left column (wherever columns are available):
 - No;
 - Yes, all pages;
 - Yes, only blog pages;
- **Enable right Menu** - defines if the Blog menu should be introduced to the right column (wherever columns are available):
 - No;
 - Yes, all pages;
 - Yes, only blog pages;
- **Enable Footer Link** - defines if the link to the blog should be available in the store footer;
- **Enable Top Link** - defines if the link to the blog should be available in the top section of the store;
- **Recent Posts** - defines the number of the most recent posts to be displayed in the blog menu (if enabled);
- **Enable Categories Menu** - defines if the blog categories can be listed in the blog menu (if enabled);
- **Tags to Display** - defines if the tags should be displayed in the blog menu (if enabled).

Menus and Links		
Enable Left Menu	<input type="text" value="No"/>	[STORE VIEW]
Enable Right Menu	<input type="text" value="No"/>	[STORE VIEW]
Enable Footer Link	<input type="text" value="No"/>	[STORE VIEW]
Enable Top Link	<input type="text" value="No"/>	[STORE VIEW]
Recent Posts	<input type="text" value=""/>	[STORE VIEW]
	<small>▲ The number of recent post to be displayed in the menu. 0 to disable.</small>	
Enable Categories Menu	<input type="text" value="Yes"/>	[STORE VIEW]
Tags to Display	<input type="text" value="10"/>	[STORE VIEW]
	<small>▲ The number of tags to display in tag cloud. 0 disables tag cloud</small>	

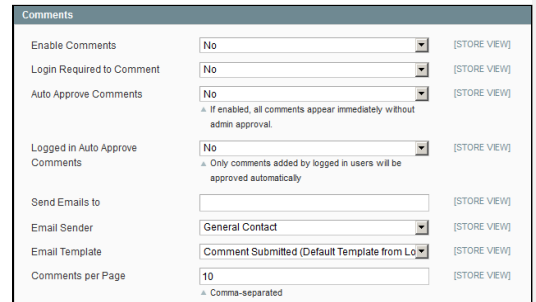
How does the menu look like?

The blog menu one can introduce to any section of the store (via widget) has the following look:



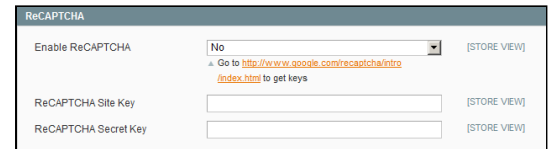
Comments

- **Enable Comments** - defines if the comments to posts should be enabled;
- **Login Required to Comment** - defines if all customers can comment on blog posts or only logged in customers;
- **Auto Approve Comments** - defines if all comments should be approved automatically or manually by the store administrator;
- **Logged in Auto Approve Comments** - defines if comments by logged in customers should be approved automatically;
- **Send Emails to** - if email address specified - notifications of the new comments will be sent to this particular email address;
- **Email Sender** - defines the notification emails sender;
- **Email Template** - defines the notification emails template (default template can be found under: `app/locale/en_US/template/email/aw_blog/`);
- **Comments per Page** - defines maximum number of comments to be displayed per blog page.



ReCAPTCHA

- **Enable ReCAPTCHA** - defines if the ReCAPTCHA should be enabled in the comments form (if enabled);
- **ReCAPTCHA Site Key** - acquired at Google ReCAPTCHA;
- **ReCAPTCHA Secret Key** - acquired at Google ReCAPTCHA.



How to get ReCAPTCHA key and secret key?

To get the required ReCAPTCHA credentials form Google navigate to:

<http://www.google.com/recaptcha/intro/index.html>

And follow two steps to register your store domain for ReCAPTCHA service.

Once registered you will only need Site Key and Secret Key. No additional manipulations to the HTML code required!

Once keys are submitted in the store backend the ReCAPTCHA will be available at the comments form (providing comments are enabled).

Submit Comment

Name *

Email *

Comment *

RECAPTCHA

Managing the Blog

All the Blog managing tools can be found under the Blog dashboard section of the store backend.

Blog section is represented with the following:

- **Add Post** - allows quickly create a blog post by redirecting to the [Add New Post](#) form;
- **Posts** - blog posts managing table - allows adding new blog posts, editing the existing posts and removing posts required;
- **Comments** - comments managing tool - allows quickly approving/disapproving comments in a bulk or removing required;
- **Categories** - allows adding new or editing existing categories;
- **Settings** - navigates the administrator to the blog [Configuration](#) section.

Adding new post

Post Information

Store administrator is redirected to the following form when **Add Post** button is clicked:

- **Title** - defines the post's title displayed at the frontend;
- **Identifier** - unique post identifier for internal use;
- **Store View** - defines the store view the post will be viewable at;
- **Category** - defines the Blog categories the post should be available at;
- **Status** - defines if the post is enabled, disabled or hidden (only available by direct link);
- **Enable Comments** - defines if the customers can comment this post;
- **Tags** - allows defining any tags for navigation convenience;
- **Short Content** - short preview of the post at the blog main page;
- **Content** - post's content.

How do these settings affect the post's look?

Post information

Title *

Identifier * e.g. domain.com/blog

Store View * English, French, German

Category * News, test

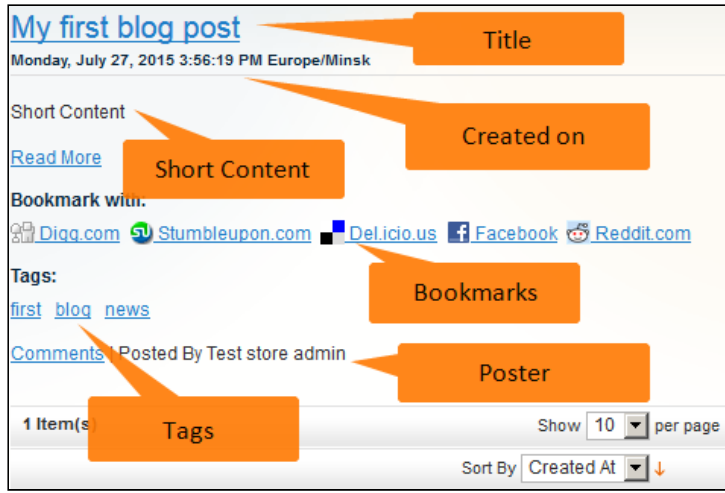
Status Enabled Hidden pages won't be indexed

Enable Comments Enabled Disabling will close comments

Tags Use comma as separator

Short Content Show / Hide Editor

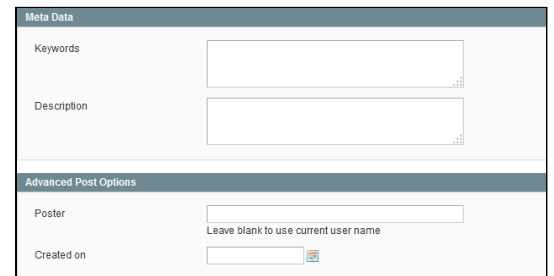
Content Show / Hide Editor



Advanced Options

The second section of the **Add Post**:

- **Keywords** - post specific SERP keywords;
- **Description** - post specific SERP description;
- **Poster** - allows defining the poster name to be displayed at the store frontend;
- **Created on** - allows defining the post created at date.



How do these settings affect the post's look?



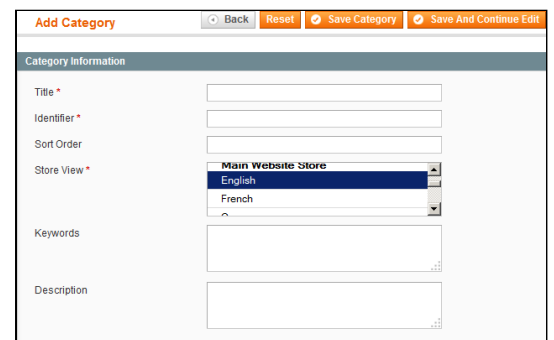
Categories

Categories are used to introduces a necessary thematic structure to the blog posts.

Blog extension once installed comes with pre-defined 'News' category one can refer to as an example.

Once can manage and created new categories at **Blog > Categories** page:

- **Title** - defines the category name that should be displayed at the store frontend;
- **Identifier** - unique category identifier for internal use;
- **Sort Order** - category sorting order helps build a category list by any relevance necessary;
- **Store View** - defines the store view the category should be available at;
- **Keywords** - category specific SERP keywords;
- **Description** - category description.



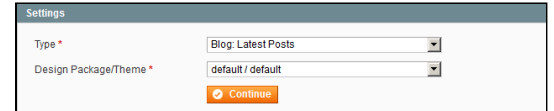
Using Blog widget

Blog extension comes with the **Latest Posts** widget one may want to use to insert a blog reference to any location of the store.

The widget comes with the following widget-specific options:

- **How many blocks to show** - defines how many posts the widget should display at the store frontend;
- **Categories to show** - defines posts from which blog categories should be displayed.

With **Categories to show** option you may want to place product category specific widgets that will display recent posts from the most relevant blog categories to let the blog advertise the products for you.



Uninstallation

If you just need to temporary disable the extension, you can just do the steps 1-5, without deleting the files of the extension.

- Disable compilation, in case it is enabled.
- Login to your FTP, navigate to **app/etc/modules/**
- Open the file AW_Blog.xml and change the following line:

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

to

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

Now your Magento is unaware of the existence of this module.

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

```
DROP TABLE IF EXISTS `aw_blog`;
DROP TABLE IF EXISTS `aw_blog_cat`;
DROP TABLE IF EXISTS `aw_blog_cat_store`;
DROP TABLE IF EXISTS `aw_blog_comment`;
DROP TABLE IF EXISTS `aw_blog_post_cat`;
DROP TABLE IF EXISTS `aw_blog_store`;
DROP TABLE IF EXISTS `aw_blog_tags`;
```

```
delete from core_resource where code = 'blog_setup';
```

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, all the extension settings, posts and user comments will be lost.

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 undefined likes this

Ни одной