Documentation / Documentation Home

Layered Navigation

Created by Unknown User (bondarev), last modified by Poleshchuk, Tatiana on мар 06, 2019

- Installation
    - Set up cron
- Configuration
    - Basic Configuration
    - Filters Management
        - Edit Filter screen
- Notes
- Uninstallation
- Troubleshooting

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

Extension page: http://ecommerce.aheadworks.com/magento-extensions/layered-navigation.html

Optimizes the navigation and lets visitors search by categories, price ranges, color, and other product attributes.
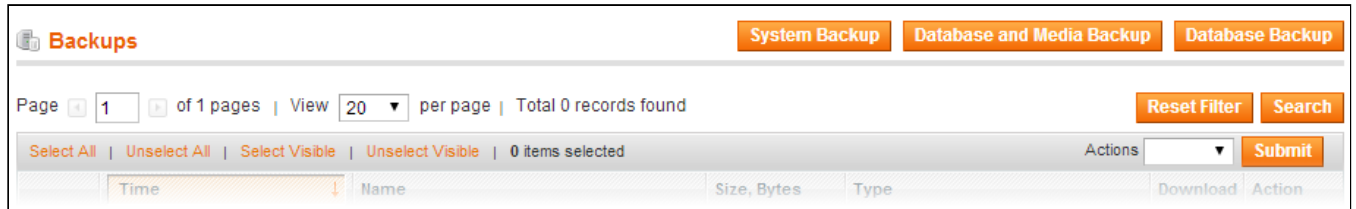
Installation

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
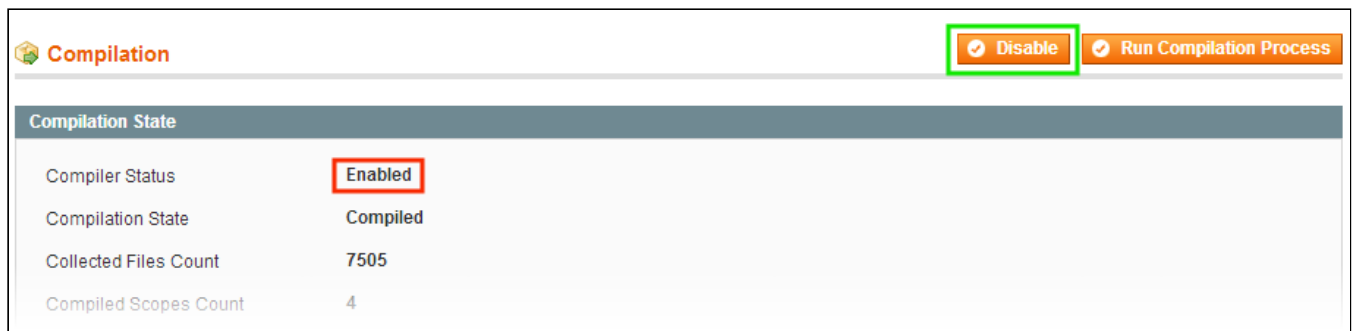

2. Log in to the Magento backend

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.

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"

📦 **Compilation**		✅ Enable	✅ **Run Compilation Process**
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

Cache Storage Management			❌ **Flush Magento Cache**	❌ Flush Cache Storage
Select All | Unselect All | Select Visible | Unselect Visible | 0 items selected			Actions Refresh ▼	Submit
Cache Type	**Description**	**Associated Tags**		**Status**
☐ Configuration	System(config.xml, local.xml) and modules configuration files(config.xml).	CONFIG		ENABLED
☐ Layouts	Layout building instructions.	LAYOUT_GENERAL_CACHE_TAG		ENABLED
☐ 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.
   The extension comes with 2 design packages: **/base/default/** and **/rwd/default/** .

   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.

Set up cron

Automatic Layered Navigation data reindexing is launched by cron. If you have already configured cron jobs for your Magento installation then you can skip this step. Read the following instructions on setting cron job for your Magento store: http://www.magentocommerce.com/wiki/1_-_installation_and_configuration/how_to_setup_a_cron_job

Generally it would be enough to run in SSH console of your server:

```
crontab -e
```

And insert the following line:

```
*/3 * * * * wget -O - -q 'http://your-store.com/cron.php'
```

Don't forget to confirm saving request when exit.

Once cronjob is set up and running, the extension will be saving every action in the log, which can be viewed in **System->Configuration->aheadWorks Extensions->Info->aheadWorks Extensions logging->View log**. If you think that the extension does not work as expected, it is recommended to check the log first. If it is empty or contains very few records, it may indicate that your cronjob is not set or set improperly. Normally, every cron launch should add 1 or more entries. Actual quantity of the entries depends on the number of processed events per given cronjob.

Configuration

Basic Configuration

You can configure the module's basic options in **Catalog-> Layered Navigation-> Configuration** (or **System-> Configuration-> aheadWorks Extensions-> Layered Navigation**).

General Settings tab

- **Enable Layered Navigation** *(Yes/ No)* enables the extension output on frontend;
- **Enable AJAX** *(Yes/No)* - this option defines whether the Layered Navigation block and product list will be dynamically updated via AJAX, or a full page reload will occur;
- **Enable reindex via cron** (*Yes/No*) - enables automatic extension data reindexing (useful if the store has a huge amount of products and attributes).

> Avoid reindexing Layered Navigation manually if you have enabled reindex via cron as it may result in customer session timeout.

Style Settings tab

This tab allows you to configure the style of AJAX progress overlay.

You can specify the **Overlay Color** and **Opacity**, choose progress **Image** and specify its dimensions (**Width** and **Height**)

Style Settings		
Overlay Color	#FBFAF6	[STORE VIEW]
Overlay Opacity	50	[STORE VIEW]
Overlay Image	Choose File No file chosen ☐ Delete Image	[STORE VIEW]
Overlay Image Width (px)	48	[STORE VIEW]
Overlay Image Height (px)	48	[STORE VIEW]

Advanced tab

Synchronize Filters - this option allows to update filters list (see below) in case the automatic update fails for some reason

Advanced
Synchronize Filters

Filters Management

You can access the list of existent Layered Navigation filters in **Catalog-> Layered Navigation-> Manage Filters.**

Manage Filters

Page ◁ 1 ▷ of 1 pages ǀ View 20 ▼ per page ǀ Total 12 records found						Reset Filter	Search

Select All ǀ Unselect All ǀ Select Visible ǀ Unselect Visible ǀ 0 items selected   Actions [▼] Submit

	Filter	Code	Type	Display Type	Image Position	Is Enabled	Is Enabled in Search	Position
Any ▼			▼	▼	▼	▼	▼	From: / To:
☐	Category	category	Category	Checkbox Group	Text Only	Enabled	Disabled	0
☐	capacity	capacity	Option Attribute	Radio Group	Text Only	Disabled	Disabled	0
☐	Color	color	Option Attribute	Radio Group	Text Only	Enabled	Enabled	0
☐	Cover	cover	Option Attribute	Radio Group	Text Only			

> **NOTE:** the filters are created automatically for all the filterable attributes.
>
> In case the filters list does not get automatically updated after an attribute gets saved, you can try using "**Synchronize Filters**" function (in **Catalog-> Layered Navigation-> Configuration-> Advanced**)

> *To enable use of the native Magento product attributes of 'Dropdown' and 'Multiselect'* ***Catalog Input Type for Store Owner*** *with Layered Navigation it will be required to set* ***Use In Layered Navigation*** *option to 'Filterable' (**Catalog-> Attributes-> Manage Attributes-> Edit Attribute-> Properties-> Frontend Properties**).*
> *As an alternative it is also possible to create custom product attribute with the same settings as the native Magento ones, but for the* ***Use In Layered Navigation*** *option.*

Edit Filter screen

General Settings ǀ Options Settings ǀ Not Visible In
In this screen, you can configure the attribute displaying options for the Layered Navigation block (set the Title, ordering position, upload option images, etc).

General Settings tab:

- **Is Enabled** - defines whether the filter is included in the Layered Navigation section in the frontend.
- **Is Enabled in Search Results** - defines whether the filter will be active in the Search Results page specifically
- **Title** - frontend filter name
- **Code** - is an internal identifier, which will also be used in URL parameters.
- **Filter Source** - this field is *not editable*, and contains a link to the associated attribute
- **Position** - sets the displaying order of the filter. The higher the value is, the lower the filter will be displayed in block
- **Display type** - defines the filter frontend representation.

How can a filter look like?

Here's how the available Display Types look like:

- Radio Group
- Checkbox Group
- Range*
- From-To*

Range and *From-To* Display types are only available for **numerical** values

Since 1.1.6 version, *Range* display type is *automatically switched to From-To* for mobile device users.

NOTE: **for Category attribute only**, one extra representation option is available. Category can be displayed as a tree. Switching on this option disables Limit Rows Quantity controls

- **Image Position** - defines the position of the attribute image in the block.

How does it look like?
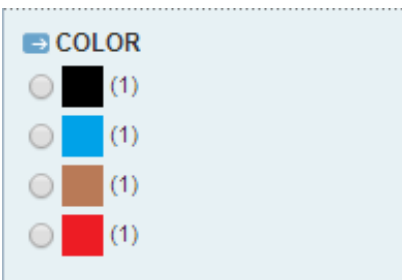There are 4 available combinations:

Text Only

COLOR

- black (1)
- blue (1)
- brown (1)
- red (1)

Left

COLOR

- black (1)
- blue (1)
- brown (1)
- red (1)

Right

COLOR

- black (1)
- blue (1)
- brown (1)
- red (1)

Image Only

COLOR

- (1)
- (1)
- (1)
- (1)

- **Column Layout** - defines how the available attribute options will be arranged within a block. The extension allows displaying 1 to 4 columns

How does it look like?
Here's how a filter looks like with Column Layout = 1 Column:

And here's the same filter with 2 Column Layout:

- **Is Help Enabled**, **Help Type** (Icon or Text) and **Help Content** - these option allow showing an attribute hint.

How does it work?
When you set Is Help Enabled option to "Yes", new input fields will show up.

The specified content will be displayed in a pop-up when the cursor is hovered over the Help Icon or Text. Here's how it looks like:
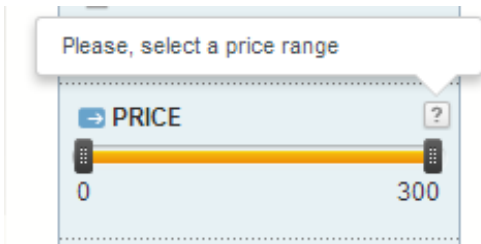
- **Limit Rows Quantity** and **Quantity Of Rows To Display** - these options allow restricting the number of attribute options to be shown in the Layered Navigation block. If an attribute has more options than the **Quantity Of Rows To Display** value, the extension will add **"See more"** link.
E.g., a product comes in 10 colors. If the Quantity Of Rows To Display for the "Color" attribute filter is set to "5", the extension will only show 5 colors in the Layered Navigation block, the rest of the available colors will be displayed after clicking "See more" link.

How does it look like?
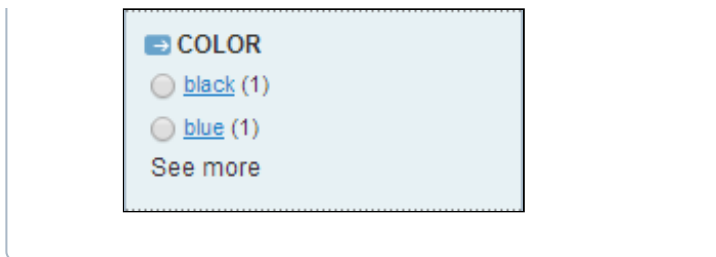
- **Show as Tree** *(Category attribute Specific)* - allows displaying categories with expanding subcategories list.

Options Settings tab:

Through this tab, you can manage the frontend representation of the available attribute values.

You can specify **Title** and **Image** of an attribute value, **Enable / disable** a value.

Sort column allows arranging the values sorting order via simple drag'n'drop procedure.
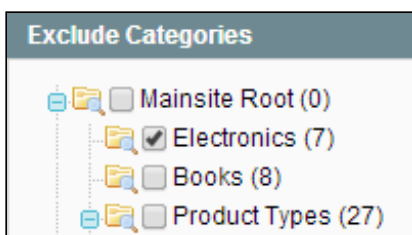
Here's how it looks like in the frontend:

Not Visible In tab:

Since version 1.1.0, the Layered Navigation extension allows to hide a filter from specific categories.

The filter will not be displayed in any category ticked in this tab.

NOTE: subcategories do not inherit the setting of a parent category. If you need to hide the filter from a category and its subcategories, make sure to tick all the necessary entries

Notes

- Since version 1.1.0, the Layered Navigation extension adds its own Index to the Magento. The Index status can be checked in **System-> Index Management** backend page. The Index is normally updated automatically on "save" event, but can also be updated manually if it is required.

- **Range** and **From-To** display types are only available for:

- Attributes with **Catalog Input Type for Store Owner = "Text field"** and **Input Validation for Store Owner = "Integer Number"** or **"Decimal Number"** (see image).

- **Category filter** will only display categories with **Is Anchor** property set to "Yes" (in *Catalog -> Manage Categories -> Edit Category -> Display Settings*)

- The extension adds "**Synchronize with Layered Navigation**" checkboxes to **Edit Category** and **Edit Attribute** screens. In case *Category Name, Attribute Labels, etc* are to be changed, you should check the **Synchronize...** option for the necessary values. Otherwise, the new values will not be imported into the filters.

Uninstallation

If you just need to temporary disable the extension, you can just do the steps 1-6, without deleting the files of the extension.

1. Disable compilation, in case it is enabled.
2. Disable the extension in the backend (**Catalog-> Layered Navigation-> Configuration-> General Settings**)
3. Login to your FTP, navigate to **app/etc/modules/**

4. Open the file **AW_Layerednavigation.xml** and change the following line:

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

to

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

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

```
DROP TABLE IF EXISTS `aw_layerednavigation_reindex_tmp`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_option_yesno_idx`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_option_yesno`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_option_index_idx`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_option_index`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_decimal_index_idx`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_category_index_idx`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_decimal_index`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_category_index`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_option_label`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_option_eav`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_option`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_label`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter_eav`;
DROP TABLE IF EXISTS `aw_layerednavigation_filter`;
```

```
delete from core_resource where code =`aw_layerednavigation_setup`;
delete from core_config_data where `path` LIKE 'aw_layerednavigation%';
```

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 the extension 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.

After the installation System->Configuration->Layered Navigation returns 404 error.

Logout from backend and login back 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.

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 http://www.aheadworks.com

Нравится   Станьте первыми кому понравится   Ни одной
это