

Price Match

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

- Installation
- Configuration
 - General Settings
 - Appearance
 - Notifications
 - Integration with AW Facebook Link extension
 - Discounts
 - Custom positioning
 - Inserting Price Match form (via CMS)
 - Inserting Price Match button
- Managing Price Match requests
- Uninstallation
- Troubleshooting

Extension page: <http://ecommerce.aheadworks.com/magento-extensions/price-match.html>

The Price Match extension allows you to be ahead of competitors with the help of your customers, reporting about lower prices on your products in other stores.

The extension supports **Simple**, **Virtual** and **Downloadable** product types.

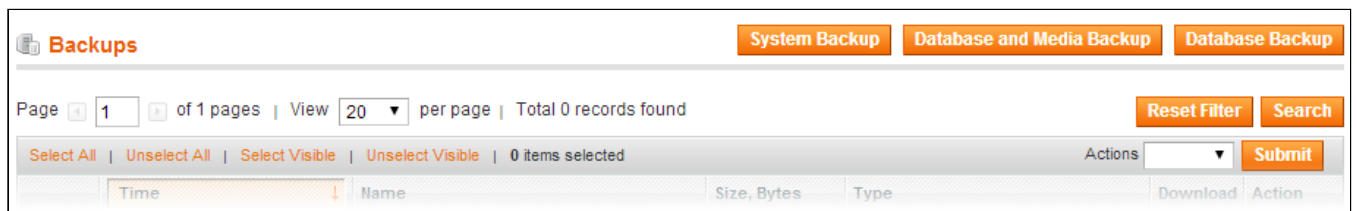
Installation

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.

Compilation Disable Run Compilation Process

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"

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
<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.

Configuration

General Settings

To configure the extension, go to the **System > Configuration > aheadWorks Extensions > Price Match** or **Catalog > Price Match Requests > Settings** backend page.

Appearance

Through this tab you can configure the appearance of the form (a popup, a separate page, or a custom page position). Also you can optionally set various text (html allow) before and after form.

- **Request Form Type** - this selector allows choosing how the frontend request form will be represented. The available option are:
 - **Popup** - the form will be displayed in a pop-up window
 - **Separate Page** - the form will be displayed on a dedicated page
 - **Custom (Manual Injection)** - the form will not be show automatically; it must be inserted to the product page via CMS (see instructions [here](#))

If you set the form type to "Custom" , the "#pricematch" anchor will be added to the "Price Match" link. After clicking the "Price Match" button, the page will be scrolled to where the "Price Match Form" is actually located.

Also make sure `pmatch/form` is added and allowed in **System -> Permissions -> Blocks**

- **Insert request button automatically on product page** - automatically adds "Price Match" button into the product pages. If this option is set to "No", the button must be inserted manually (see instructions [here](#))
- **Request button label** - here you can define the text of the Price Match button
- **Text before / after form** - these two fields allow to add desired content to the header and footer sections of the Price Match form

- **Show option "I'd like to pay for it"** - enables / disables an additional field where a customer can specify the desired price

How does the form look like?
The below example shows a pop-up form.
Highlighted are the following elements:

1. Text before form
2. "I'd like to pay for it" optional field
3. Text after form

The screenshot shows a 'Price Match' form with the following elements highlighted:

- A yellow box highlights the text: "If you find a lower price on any our product, we will match the price and get back the difference."
- An orange box highlights the "Price Match" title.
- An orange box highlights the "I'd like to pay for it" input field.
- An orange box highlights the "Thank you!" text at the bottom left.

The form includes fields for: Your Name *, Email *, Telephone, Product Name * (filled with "Sony VAIO VGN-TXN27N/B 11.1"), Competitor's product link *, Competitor's price *, and an Information text area. It also has "Cancel" and "Submit" buttons, and a "* Required Fields" note.

Notifications

Here you can configure the email notifications sent out by the extension



- **Send To** - defines the recipients of the notification:
 - Both Customer & Admin
 - Admin Only
 - Customer Only
 - Disable
- **Sender** - defines the Sender of the outgoing emails
- **Recipient** - for admin-side notifications

The extension sends notifications on new incoming requests, approval or rejection of the existent ones. You can adjust the text of these emails via modifying the below templates:

- */app/locale/en_US/template/email/pmatch_email_approved_email.html*
- */app/locale/en_US/template/email/pmatch_notify_to_customer.html*
- */app/locale/en_US/template/email/pmatch_notify_to_admin.html*
- */app/locale/en_US/template/email/pmatch_email_rejected_email.html*

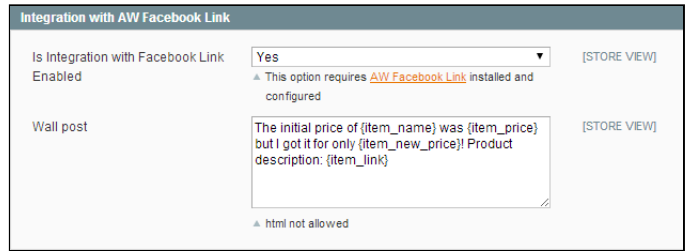
Email Template Variables

The extension has a number of special variables that can be used in notification **Email templates**:

Code	Description
<code>{{var request.customerName}}</code>	customer name
<code>{{var request.customerEmail}}</code>	customer email
<code>{{var request.productName}}</code>	product name
<code>{{var request.competitorLink}}</code>	link to competitor's site
<code>{{var request.competitorPrice}}</code>	competitor's price
<code>{{var request.info}}</code>	request information
<code>{{var request.customerId}}</code>	empty if question from guest
<code>{{var request.productId}}</code>	product id
<code>{{var request.productPrice}}</code>	price of the product
<code>{{var request.requestId}}</code>	request id
<code>{{var request.storeId}}</code>	store id
<code>{{var request.createdTime}}</code>	creation time
<code>{{var request.updateTime}}</code>	update time
<code>{{var request.productUrl}}</code>	link to product page
<code>{{var request.productEdit}}</code>	link to product in admin panel
<code>{{var discountUseBefore}}</code>	use a discount of up to
<code>{{var discountCode}}</code>	discount code for guest Price Match
<code>{{var linkToRequest}}</code>	link to request in admin panel
<code>{{var customerPhone}}</code>	customer phone number
<code>{{var discount}}</code>	discount

Integration with AW Facebook Link extension

This functionality becomes available if the [Facebook Link](#) module is installed at the store. Through this tab, you can enable the automated wall posts and adjust the posts text.



Discounts

This tab allows controlling how the extension-generated discounts are applied.

- **Apply discount coupons automatically** (works for registered customers only) - enables / disables automated discounts usage. If enabled, the price of an item will be automatically adjusted for the lower price reporter; otherwise, the user will have to submit the discount code to get the bonus.
- **Block other coupons, if Price Match coupon was applied** - enabling this option will invalidate all the other promo codes a customer may have in case they apply a price match discount.
- **Automatically accept requests** - enables / disables the need for a store admin to approve the incoming PM requests. If enabled, all the requests that do not exceed the **Price interval from original price, %** value set below, will be automatically approved



Auto-approval restrictions

If you enable the Auto-approval option, a given customer will be limited to **1 request per product**.

I.e., if a customer has already submitted a price match request for a product, the extension will not accept any more requests for this particular item from that customer. The Price Match button will be hidden.

Custom positioning

Inserting Price Match form (via CMS)

You can insert price match request form at any page, using the following code:

```
{{block type="pmatch/form" product_id="164"}}
```

where **product_id** is the id of the product (optional), which name and price will be included into the form.

Do not insert this line in .phtml files directly, as it will work only if inserted into CMS block or page.

Inserting Price Match button

In case you have set **Insert request button automatically on product page** option to **"No"** (in **System->Configuration->aheadWorks Extensions->Price Match->Appearance**), you will need to call the button manually, through a page template file.

The button can be inserted into templates using the following code:

```
<div class="pmatch-button"><?php echo Mage::helper('pmatch')->getPMatchButton()?></div>
```

Managing Price Match requests

All the incoming price match requests are listed in **Catalog -> Price Match Requests -> All Requests** grid (a quick shortcut for unprocessed ones is also available, **Catalog -> Price Match Requests -> Pending Requests**).

After opening a pending request, you will land on the Request Information screen:

Status - Pending / Approved / Rejected

Changing Status to "Approved" and saving the request will activate the discount.

Below is a block of information-only non-editable fields:

- **Created on** - the date of the request submission
- **Store** - the origin of the request (storeview)
- **Product** - the link to backend edit product page
- **Customer** - the link to backend Customer Account page

The next block contains price-related values:

- **Current price for customer** - the actual price the item is currently available at to the specific customer (Price Rules, Group pricing, etc. are applied)
- **Product original price** - the list price of the item
- **Requested price** - the value specified by the customer in "I'd Like To Pay" field

If the correspondent frontend field is disabled in the **General Settings**, this field will still be present, but will have \$0 value

- **Competitor's price** - the reported price at the competitor site
- **Difference between competitor's price and current price** and **Difference between requested price and current price** fields - automatically calculated differences, represented in both absolute (\$) and relative (%) values

Request Information	
Status	<input type="text" value="Pending"/>
Created on	Mar 5, 2014 4:13:35 AM
Store	English
Product	Nokia 2610 Phone
Customer	John Smith < test.ahead@vandex.ru >
Current price for customer	\$149.99
Product original price	\$198.99
Requested price	\$139.00
Competitor's price, USD	<input type="text" value="139"/>
Difference between competitor's price and current price	\$10.99 (7.33 %)
Difference between requested price and current price	\$10.99 (7.33 %)
Discount, USD	<input type="text" value="0.00"/>
Discount active to *	<input type="text"/>
Discount usage	<input type="text" value="Only once"/>
Link	http://ecommerce.aheadworks.com/magento-extensions/price-match.html
Phone Number	<input type="text" value="123456789"/>
Additional information	<input type="button" value="Show / Hide Editor"/> <input type="button" value="Insert Widget..."/> <input type="button" value="Insert Variable..."/>
	<div style="border: 1px solid #ccc; height: 40px;"></div>

In the block below, you can adjust your offer:

- **Discount, USD** - here you can specify the discount amount (in the store's base currency, absolute value)
- **Discount active to** - here you are to set the time restriction
- **Discount usage** - Unlimited / Only Once / Used

Below you can find the additional details provided by the customer:

- **Link** - the link to the competitor's store
 - **Phone Number** - customer's phone number
 - **Additional Information** - The content of Information field submitted by the customer
-

Uninstallation

1. Disable compilation, in case it is enabled.
2. Since Magento doesn't give an opportunity to perform an automatic uninstall of the modules, you need to run the following query on your database manually:

```
DROP TABLE `aw_pmatch_requests` ;
```

This will delete the extension's attributes from the database. After that, you can safely remove the extension's files from your store:

3. Login to your FTP, navigate to **app/etc/modules/**
4. Open the file AW_Pmatch.xml and change the following line:

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

to

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

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.

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 <http://www.aheadworks.com>

Нравится
это

Станьте первыми кому понравится

Ни одной