

# Group Deals

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

- Installation
  - Set up cron
  - Setting up permissions
- Creating new deal
  - General Section
  - Details
  - Coupons
  - Orders
  - Information
- Configuration
  - General settings
  - Deals tabs
  - Notifications
  - Email variables
- Customer account
- Troubleshooting

Extension page: <http://ecommerce.aheadworks.com/magento-extensions/group-deals.html>

Group Deals is a great choice for those online merchants who want to organize collective purchases process.

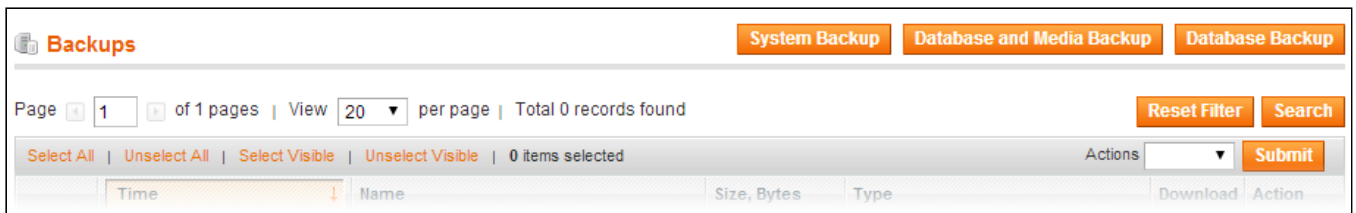
## Installation

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.  
 The extension comes with 2 design packages: **/base/default/** and **/pro/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

In order to send emails extension needs to be 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: [Magento user guide](#)

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.

## Setting up permissions

**Group Deals** extension allows uploading custom images for the deals instead of the product ones. For uploading images folder **'media/aw\_collpur'** must be set to **777** permissions.

Except this folder all other folders should be set to required permissions.

## Creating new deal

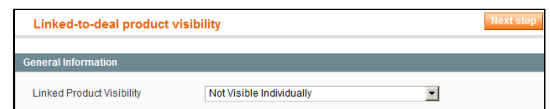
For the convenience purposes of the readme, first process of creating new group deal will be covered.

To create a new deal navigate to **Group Deals > New Deal** and click **Add New** button.

At this stage it will be suggested to select the product that one wants to create a group deal for.

Group Deals extension works with all product types except: bundle, grouped and gift card.

Once the product is selected it will be suggested to specify **Linked Product Visibility** for this product:



- Not Visible Individually;
- Catalog;
- Search;
- Catalog, Search.

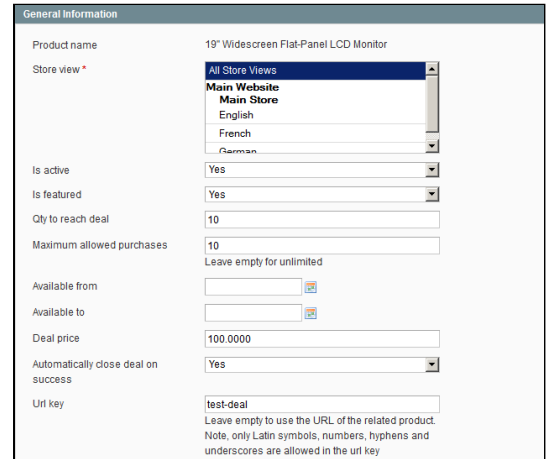
These represent native Magento visibility settings. Visibility settings can prove useful if you do not want the product to be available for purchase along with the deal.

Hit **Next step** button to proceed to the **General Section** of the deal settings.

## General Section

**General Section** contains most important group deal settings:

- **Product name** - represents the name of the product the deal has been created for;
- **Store view** - allows defining the store view(s) the deal should be available at;
- **Is active** - allows enabling/disabling the deal;
- **Is featured** - defines if the deal should be considered featured and represented in the **Featured Tab** at the frontend of the store;
- **Qty to reach deal** - defines how many purchases should be made for the deal to be considered successful;
- **Maximum allowed purchases** - defines if the deal should be available only for a limited number of purchases;
- **Available from** - defines the deal start date. If the start date is set to any date in the future the deal will be considered **Upcoming** and displayed in a **Upcoming deals tab**;
- **Available to** - defines the deal end date;
- **Deal price** - allows specifying the price for the product on the deal;
- **Automatically close deal on success** - defines if the deal should be closed automatically once the product has been purchased for **Qty to reach deal** times;
- **Url key** - allows adjusting the deal unique URL.



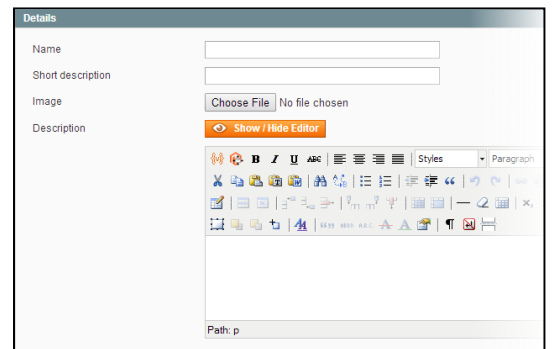
## Details

The **Details** section responds for the deal content:

- **Name** - allows setting up the deal name;
- **Short description** - short description of the deal displayed below the deal name at the deal page;
- **Image** - allows uploading image for the deal;

For proper image upload folder **'media/aw\_collpur'** must be set to **777** permissions.

- **Description** - deal description available under the main deal block.



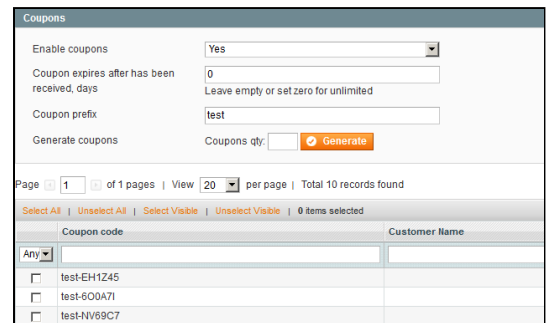
How does these options affect the deal look?



## Coupons

**Coupons** section of the deal configuration has been implemented to cover the needs of store administrators that require physical coupons printed for customers:

- **Enable coupons** - defines if the coupons should be available for this deal;
- **Coupon expires after has been received, days** - defines the period the coupon will be valid for. If the expiration date set, an email with the coupon code sent will have a coupon expiration date;
- **Coupon prefix** - allows defining the coupon code prefixes;
- **Generate coupons** - allows generating set amount of the coupon codes for the deal. When generating coupon codes please make sure the number of the coupon codes generated corresponds with the **Maximum allowed purchases** value, to prevent any issues with customers not receiving coupon codes.



The coupons generated with the Group Deals extension are meant for printing only and are not available for redeeming at the Magento store.

If the coupons enabled and generated, customers purchasing the deal will receive an email with the Group Deals coupon generated. The email will suggest printing this coupon and coming to redeem it to the store.

Customers can also look through the coupon codes for the deals purchased under the **My Deals** section of the customer account.

## Orders

The Orders section of the deal represents an order grid with all the orders placed for the deal listed. The grid comes with the following columns:

- Order#;
- Purchased From (Store);
- Purchased On;

- Bill to Name;
- Ship to Name;
- G.T. (Base);
- G.T. (Purchased);
- Status;
- Action.

Edit Deal 'Classic Vacations - Tahiti'			
Page 1 of 1 pages   View 20 per page   Total 1 records found			
Order #	Purchased From (Store)	Purchased On	Bill
10000016	Main Website Main Store English	Dec 23, 2013 4:52:55 AM	Job
10000017	Main Website Main Store	Dec 23, 2013 5:02:17 AM	Ma

## Information

The Information section of the deal details represents overall report on the deal:

- Product name;
- Product price;
- Condition;
- Time to start (if set);
- Time to finish (if set);
- Purchases to complete.

Details	
Product name	Sony Walkman
Product price	\$50.00
Condition	Not Running
Time to start	3 days 23:48
Purchases to complete	5

## Configuration

**Group Deals** general and notification settings can be found under **System > Configuration > AheadWorks extensions > Group deals**.

Along with the general settings the configuration represents different **Group Deals** tabs e.g.: Active deals tab, Closed deals tab. These tabs share common set of settings, for convenience purposes this readme will cover all four (4) tabs in a single section.

Configuration	Group deals
AHEADWORKS EXTENSIONS	General settings
Group deals	Active deals tab settings
aheadWorks	Upcoming deals tab settings
ADD-ONS STORE	Closed deals tab settings
Info	Featured deal tab settings
	Notifications

## General settings

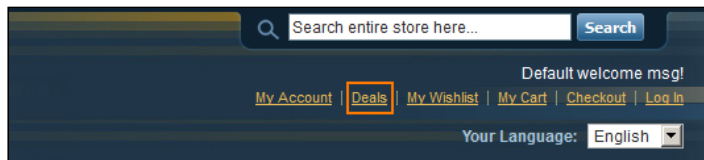
**General settings** section controls the most basic aspects of the extension's functionality:

- **Enabled** - defines if the extension is enabled or disabled (same as **System > Configuration > Advanced > Advanced > AW\_Collpur**);
- **Show deal menu in top links** - defines if the 'Deals' menu should be available at the top links along with 'My Account', 'Log In' and other links;

General settings	
Enabled	Yes [STORE VIEW] Enable or disable module frontend functionality
Show deal menu in top links	Yes [STORE VIEW]
Thumbnail size, px	325 [STORE VIEW]
Deal view image size, px	669 [STORE VIEW]

How does the top link look like?

Customers can click the **'Deals'** link to proceed to the Group Deals dedicated category.



**Please note:** if you decide to change the **Deals** link position or rename it remember that the link should always refer to the **base\_url/deals/**

- **Thumbnail size, px** - defines the deal thumbnail width for catalog (thumbnail height is adjusted automatically);
- **Deal view image size, px** - defines the deal image width for the **Deals** page (image height is adjusted automatically).

## Deals tabs

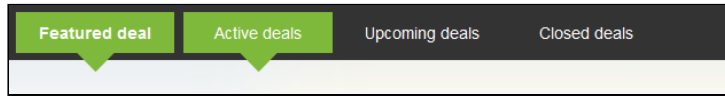
The extension introduces several Deals 'categories'/tabs that can be accessed by clicking the Deals in the top links menu. There are four (4) tabs each having a predefined role:

- **Active Deals** - in this tab only currently active and featured deals are displayed;
- **Upcoming Deals** - in this tab that only the deals scheduled for a particular date are displayed;
- **Closed Deals** - in this tab the deals that have been successfully closed or reached the end date are displayed;
- **Featured Deals** - in this tab the deals that have been marked as Featured are displayed.

Active deals tab settings	
Enabled	<input type="checkbox"/> Yes <small>[STORE VIEW]</small>
Sort order	10 <small>[STORE VIEW]</small>
Title	Active deals <small>[STORE VIEW]</small>
Upcoming deals tab settings	
Closed deals tab settings	
Featured deal tab settings	

### How does the deal tabs look like?

When navigating to Deals section, depending on what tabs have been enabled, customers will be presented with the following menu:



The tabs share the same set of settings:

- **Enabled** - defines if a tab should be enabled on the frontend;
- **Sort order** - defines the tab position (the tab with the lower **Sort order** value is the leftmost);
- **Title** - allows adjusting the tab title.

## Notifications

The Notifications configuration sections represents a set of email templates that (if enabled) are fired to store administrator/customers on a particular extension related event:

- **Enable** - defines if the email notifications are enabled;
- **Email sender** - allows defining the default sender of the notifications that will displayed in the *From*: field;
- **Administrator email** - allows defining the email address all admin notifications should be sent to. Leave empty to send notifications to the email address specified in **General Contact**;
- **Deal succeed template for admin** - defines the email template that should be sent to the store admin upon deal hitting 'successful' status;
- **Deal succeed template for customer** - defines the email template that should be sent to the customer upon deal hitting 'successful' status;

Notifications	
Enable	<input type="checkbox"/> No <small>[STORE VIEW]</small>
Email sender	General Contact <small>[STORE VIEW]</small>
Administrator email	<input type="text"/> <small>[STORE VIEW]</small> <small>* If empty, emails will be sent to General Contact</small>
Deal succeed template for admin	Admin deal succeed (Default Template from Lo... <small>[STORE VIEW]</small>
Deal succeed template for customer	Customer deal closed and succeed (Default Te... <small>[STORE VIEW]</small>
Deal failed template for admin	Admin deal expired (Default Template from Loc... <small>[STORE VIEW]</small>
Deal failed template for customer	Customer deal expired (Default Template from I... <small>[STORE VIEW]</small>
Notify admin before deal expires, days	<input type="text"/> <small>[STORE VIEW]</small>
Notify about deal expiration template	Deal expires before (Default Template from Loc... <small>[STORE VIEW]</small>

### How does Deal succeed notifications work?

After deal has reached the **Qty to reach deal** limit (the number of invoiced purchases), it is considered to be successful and on the next cron run email notifications (if enabled) will be sent to store administrator and customers who paid for the deal. All other notifications will be sent automatically on invoice process. If coupons are enabled but quantity generated is not enough, customer(s) left without the coupon codes will only receive an email notification upon proper number of coupon codes generated. In such cases notification emails are sent right after coupon generation process.

- **Deal failed template for admin** - defines the email template that should be sent to the store admin upon deal hitting 'failed' status;
- **Deal failed template for customer** - defines the email template that should be sent to the customer upon deal hitting 'failed' status;

#### How does Deal failed notifications work?

Deal considered to be failed if:

- Administrator manually closed it as failed by clicking the "Close as Failed" button in the edit mode of not yet succeed deal;
- Deal has not reached the **Qty to reach deal** limit in the predefined time limit.

Failed deals are processed by cron.

- **Notify admin before deal expires, days** - works in conjunction with the **Notify about deal expiration template** setting; defines how many days before the deal expiration the notification email should be sent to the store administrator;
- **Notify about deal expiration template** - defines the email template that should be sent to the store administrator before the set number of days before the deal expiration.

**Administrator notifications** are related to store 0, i.e affected by default Magento store settings (Current Configuration Scope: Default Config).

**Customer notifications** are related to specific store, i.e. the store where the deal has been purchased.

Thus, if required notifications can be configured separately for customers and administrators considering the store view the deal has been purchased at.

*Email notifications will NOT be sent if the deal is/has been disabled.*

*Email notifications will NOT be sent to customers if at the moment of purchase notifications were disabled and enabled later.*

## Email variables

The notification email templates can be found and edited only at the store FTP at [app/locale/en\\_US/template/email/aw\\_collpur/](app/locale/en_US/template/email/aw_collpur/)

All administrator templates receive the whole deals model object and can contain the following variables:

Code	Description
<code>{{var deal.getName()}}</code>	deal name
<code>{{var deal.getProductname()}}</code>	related to deal lined product name
<code>{{var deal.getEditPage()}}</code>	link that leads to the deal edit page
<code>{{var deal.getAvailableFrom()}}</code>	GMT date time stored in DB format (Y-m-d h:i:s)
<code>{{var deal.getAvailableTo()}}</code>	GMT date time stored in DB format (Y-m-d h:i:s)
<code>{{var deal.getDescription()}}</code>	short deal description



Code	Description
<code>{{var deal.getFullDescription()}}</code>	full deal description
<code>{{var deal.getQtyToReachDeal()}}</code>	qty needed for deal to be considered successful
<code>{{var deal.getMaximumAllowedPurchases()}}</code>	the maximum allowed purchases
<code>{{var deal.getPrice()}}</code>	deal price
<code>{{var deal.getIsFeatured()}}</code>	weather deal is featured (1 - featured, 0 - not featu
<code>{{var deal.getProductId()}}</code>	linked product id

All customer templates receive deals model object (see administrator variables) and complete Mage\_Sale\_Model\_Order n object. Meaning that in addition to the variables above, customer templates receives the following variables:

Code	Description
<code>{{var order.getAccountLink()}}</code>	link to customer account page
<code>{{var order.getCustomerFirstname()}}</code>	customer first name
<code>{{var order.getCustomerLastname()}}</code>	customer last name
<code>{{var order.getCouponCode().format('html')}}}</code>	coupon code(s) if enabled
<code>{{var order.getCouponExpireDate()}}</code>	coupon expiration date if available
<code>{{var order.getCustomerEmail()}}</code>	customer email
<code>{{var order.getStoreName()}}</code>	store name

## Customer account

Group Deals extension introduces the **Deals** category to the top links of the store and **My Deals** customer account section.

My Deals section represents the most information on the all deal the customer has purchased.

By clicking **view** link of a particular deal customer is redirected to the deal details screen.

Here all the deal related coupon codes can be found and printed.

Deal information					
2 Item(s)					Show 10 per page
ID	Deal name	Status	Purchased	Expires	Action
1	Night Eff LVL 80	Paid	2015-06-24 11:39:54	--	<a href="#">view</a>
2	Great Gatsby deal	Paid	2015-06-25 11:45:34	--	<a href="#">view</a>
2 Item(s)					Show 10 per page

### How does the deal details screen look like?

**Purchase Information**

ID: 2	Deal name: <a href="#">Great Gatsby deal</a>
Price paid: \$35.00	Deal status: Succeed
Payment status: Paid	Purchased at: 2015-06-25 11:45:34
Expires at: --	

**Available Coupons**

1 Item(s) Show 10 per page

Coupon code	Status	Expires at	Action
gatsby-42CF5Q	Not used	--	<a href="#">print</a>

1 Item(s) Show 10 per page

## 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    Станьте первыми кому понравится это

Ни одной