

Countdown

- Installation
  - Set up Cron
- Configuration
  - Create/ Edit Timer screen
  - Creating Design Packages
- Inserting the timer block
  - Using CMS or Widget
  - Inserting via XML Layout
  - Inserting product timers into Category View page
- Logging and Statistics
- Uninstallation
- Troubleshooting

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

The Countdown Magento extension displays the countdown block on the home page, product page or any CMS page of your Magento store.

## 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"



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

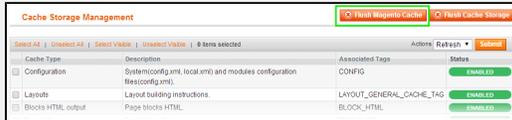
#### 4. Flush store cache

##### 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



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.

## Set up Cron

Countdowns need to be launched by cron for starting / ending. If you have already configured cron jobs for your Magento installation, 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 'http://your-store/cron.php'
```

Don't forget to confirm saving request when exit.

## Configuration

The extension is configured on per-instance basis, i.e., the settings are defined for each of the timer blocks separately.

You can create new and manage the existent timers in Promotions -> Countdown Timers backend page.

## Create/ Edit Timer screen

[General](#) | [Design](#) | [Automation](#) | [Triggers](#) | [Recurring](#)

### General tab

In this tab, you are to define the basic timer parameters.

- Name - defines the timer identifier for backend
- Status - activates / deactivates the rule
- From/To - these fields define the lifetime of the countdown timer block;
- Store view - allows selecting the storeview(s) the timer will appear on
- Customer Groups - defines what customer groups the timer will be displayed for
- After click on timer, customer will be redirected to - you can specify the redirect URL here
- Priority - defines the priority of the timer block. The lower value is, the more priority a timer has.

I.e., if there are several blocks that are to be rendered in the same position, the block with lower Priority value will be displayed

The screenshot shows the 'General' tab configuration for a timer. The fields are as follows:

Name *	sample-timer
Status	Enabled
From: *	03/5/14 10:31 AM
To: *	03/31/14 10:31 AM
Store view *	All Store Views Main Website Main Website Store English
Customer Groups *	NOT LOGGED IN General Wholesale Retailer
After click on timer, customer will be redirected to:	
Priority	0

▲ for timers with non-product page positions

### Design tab

In this tab, you can configure the design of the timer instance.

#### Design section

- Title - the frontend title of the timer
- Design package - you can select the preferred [package](#) here



## Automation tab

Through this tab, you can automate the timer displaying.  
Products subselection section:

Here, you can define the conditions for the products the timer will be applied to. The conditions are based on product attributes.

### Automation section:

- Show for selected products - defines the timer position
  - Inside product page - the timer will automatically appear in the product page
  - Custom - requires inserting the timer instance manually, as a widget
- Disable selected products when countdown is over - works for all the products that meet the conditions set in Product Subselection section above
- Enable selected products on countdown start - works for all the products that meet the conditions set in Product Subselection section above



If you are going to use "enable / disable product" options  
Make sure to set the conditions in the Product Subselection section.  
Otherwise, these options will affect all the store items



"Enable / Disable product" options have "website" scope. It means that if they are enabled, Priority and Customer Groups set for the timer are ignored. As for the Store view selector, the functions will apply for the whole website an assigned Storeview can be associated with.

#### Products subselection

If **ALL** of these order conditions are **TRUE**

SKU **is one of** sku-1, sku-2, sku-8, sku-9 



---

#### Automation

Show for selected products *	<input type="text" value="Inside product page"/>
Disable selected products when countdown is over:	<input type="text" value="No"/>
Enable selected products on countdown start:	<input type="text" value="No"/>

**Be careful with enable and disable options. Note, if you haven't specified any conditions for the timer, turning these options on will enable or disable ALL the products at your store**

## Triggers tab

The extension can be used to trigger (i.e., switch on or off) existent Shopping Cart or Catalog Price rules. This function is configured through this tab.

**Triggers**

[+ Add Trigger](#)

On: <input type="text" value="countdown start"/>	<input type="text" value="Enable"/>	<input type="text" value="Catalog Rule"/>	<a href="#">select</a> -10% if name consists DIGITAL BOOK	<a href="#">Remove</a>
On: <input type="text" value="countdown end"/>	<input type="text" value="Disable"/>	<input type="text" value="Catalog Rule"/>	<a href="#">select</a> -10% if name consists DIGITAL BOOK	<a href="#">Remove</a>

Note, you can create several triggers for each countdown start/end. Triggers are executed by cron.

### Recurring tab

Since 1.1.0 version, the Countdown extension allows setting up recurring timers. I.e., the timer blocks will appear according to the specified schedule.

There are 3 modes:

- Type = "Every day"

The timer will be displayed every day for From-To interval

Include Weekdays field allows restricting the timer to certain weekdays .

**Recurring**

Status:

Type\*:

From:  :  :

To:  :  :

Include Weekdays

- Sunday
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday

- Type = "Every X day(s)"

The timer will be reset every X days (defined in Day Range field), at a time set in the To field.

**Recurring**

[Click here to view example...](#)

Status:

Type\*:

Day Range\*:

To:  :  :

In this example the Countdown block timer will be reset every five days at 18:00 store time.

**Recurring**

Status:

Type\*:

Day Range\*:

To:  :  :

- Type = "Every defined day"

The timer will be displayed on the specified Day of Month, since 00:00 until To time of the target date.

Recurring	
Status	Enabled
Type *	Every defined day
Day of Month *	1
To:	19 : 00 : 00

## Creating Design Packages

The Countdown extension comes with 3 ready-made packages, but you can also create your own ones per instructions below.

1. Create a directory in skin/frontend/default/default/aw\_countdown (use your store's values), e.g. "mypackage"
2. In this new directory, create timer.css or copy this file from any default package and edit it.

Use your package name as postfix in CSS rules.

Example:

```
.aw_countdown_timer_container_mypackage{
  text-decoration:none;
  color: #000;
}
```

3. Go to the add/edit countdown page in the backend. The created package will appear in the Design Package dropdown of the [Design](#) tab.

## Inserting the timer block

Countdown timers can be inserted either automatically (if [Show for Selected Products](#) option is set to "Inside product page") or manually, as per instructions below.

## Using CMS or Widget

Open the static block or CMS page you want countdown block to be displayed at and insert block as widget. Also you can use the following text in the content field:

```
{{block type="awcountdown/countdown" countdownid="8"}}
```

where countdownid is the ID of the countdown that will be inserted

## Inserting via XML Layout

You can insert countdown in any position via XML Layout

```
<block type="awcountdown/countdown" name="awcountdown">
  <action method="setCountdownid"><countdownid>1</countdownid></action>
</block>
```

where countdownid is the ID of the countdown that will be inserted

---

## Inserting product timers into Category View page

Since version 1.1.0, the Countdown extension allows inserting timer boxes into the products listed under a category.

To display the timers like that, make sure to set Display Automatically to "Inside Product Page".

Then, navigate to your FTP and open the following file `app/design/frontend/<your_package>/<your_theme>/template/catalog/product/list.phtml`.

To show the timers in List mode for RWD Theme:

- find the below code string

```
<div class="desc std">
```

- insert the below code BEFORE this string

```
<?php echo
$this->getLayout()->createBlock('awcountdown/countdown')->setProductId($_product->getId())->toHtml();?>
```

To show the timers in List mode for default theme:

- find the below code string

```
<ul class="add-to-links">
```

- insert the below code BEFORE this string

```
<?php echo
$this->getLayout()->createBlock('awcountdown/countdown')->setProductId($_product->getId())->toHtml();?>
```

To show the timers in Grid mode:

- find the below code string

```
<div class="actions">
```

- insert the below code BEFORE this string

```
<?php echo
$this->getLayout()->createBlock('awcountdown/countdown')->setProductId($_product->getId())->toHtml();?>
```

In case you would like to show timers in both modes, you will need to do both adjustments as well

---

## Logging and Statistics

All notifications about starting/ending triggers are saved in AW Core log. You can see this log here: System > Configuration > Info (aheadWorks Extensions section) > aheadWorks extensions logging (group) > View Log (button)

The number of executed triggers is also saved here.

---

## 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_Countdown.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_countdown_index_idx`;  
DROP TABLE IF EXISTS `aw_countdown_index`;  
DROP TABLE IF EXISTS `aw_countdown`;
```

```
delete from core_resource where code = 'countdown_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, you will need to create all rules 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.

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>