Documentation / Documentation Home

Follow Up Email

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

Having hard time finding an answer to your question?

Check out our Knowledge Base.

Magento version compatibility	
Community	1.4.1.1 - 1.9.2.0
Enterprise	1.12.0.0 -1.14.2.0

Follow Up Email

Extension page: **http://ecommerce.aheadworks.com/magento-extensions/follow-up-email.html**

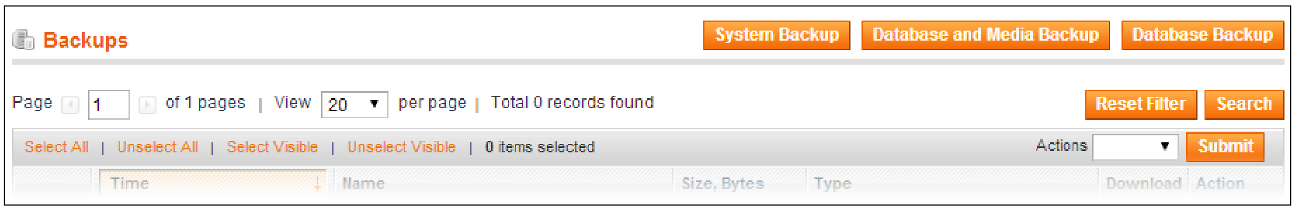
Follow Up Email is a powerful tool for any ecommerce site. It gives a comfortable and flexible way to automatically reach your customers on day X after or before some event occurs.

Never lose a customer after a sale, stay in contact with people who are most likely to be interested in another purchase. Use follow up emails to get opinion about the products a customer has purchased, request a feedback, send attractive discounts and promotions, and anything else you could imagine. People like seeing positive and careful attitude towards them, let them know that they are not abandoned after a purchase and are welcome to come back with a question, review or a new purchase. Establish stronger relationships, increase customers' loyalty and as a result get more consequent sales with Follow Up Email extension brought to you by aheadWorks Co.

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



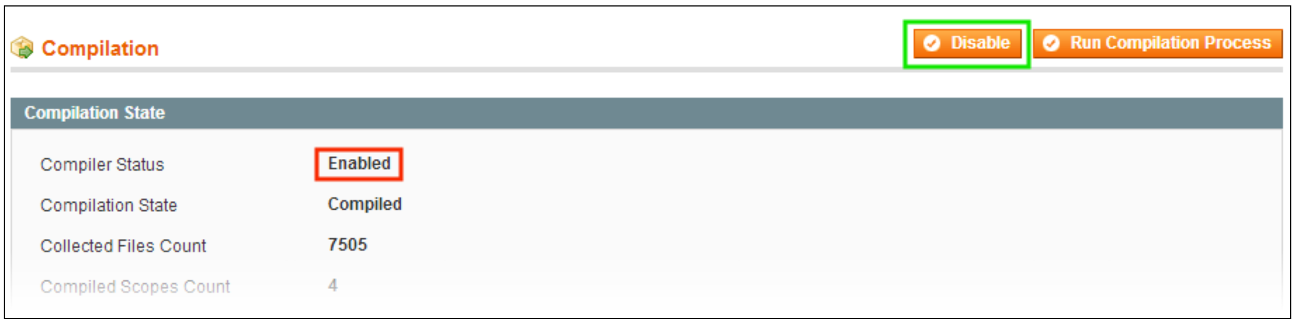
Backups
System Backup
Database and Media Backup
Database Backup
Page 1 of 1 pages | View 20 per page | Total 0 records found
Reset Filter
Search
Select All | Unselect All | Select Visible | Unselect Visible | 0 items selected
Actions
Submit
Time
Name
Size, Bytes
Type
Download
Action

2. Log in to the Magento backend

3. Disable compilation
   Click to view details
   Navigate to **System -> Tools -> Compilation**.

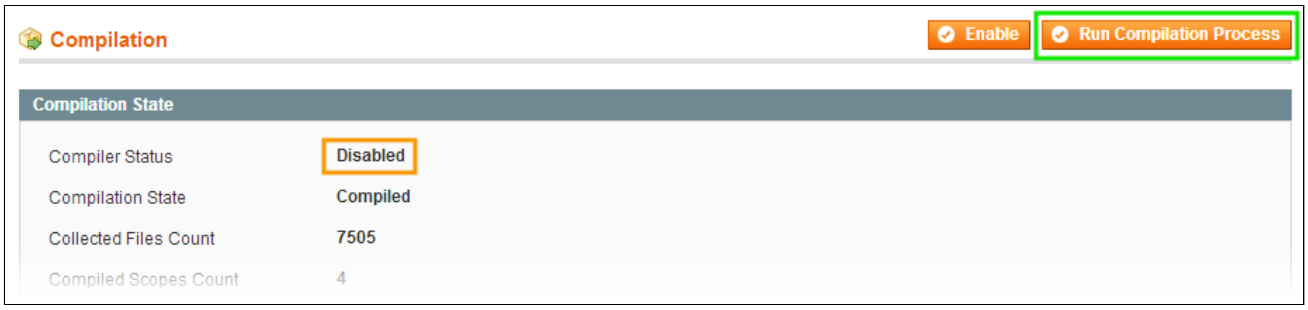
   If Compiler status is Disabled, you can skip to the next step

   If Compiler is enabled, disable it.



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 State

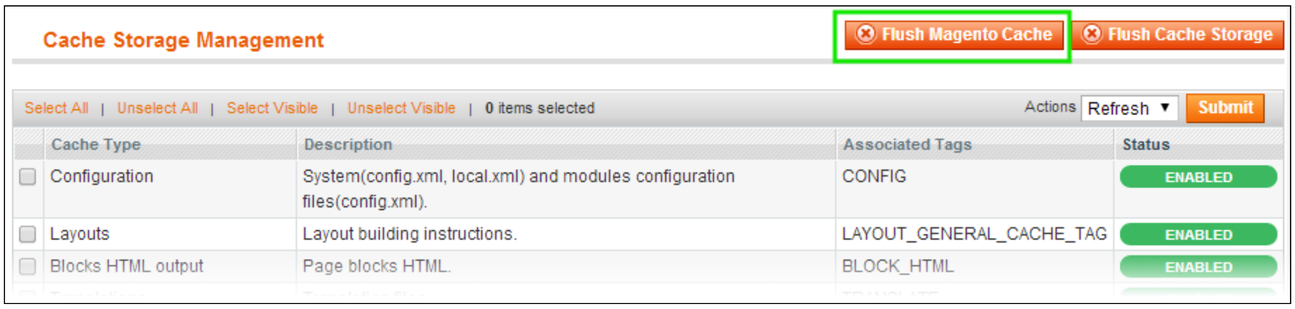
Compiler Status	Disabled
Compilation State	Compiled
Collected Files Count	7505
Compiled Scopes Count	4

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 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

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.

Enable System Log

The most recent versions of Magento Community (starting from 1.9.2.0) and Enterprise (starting from 1.14.2.0) introduced the customer activity logging.

Follow Up Email requires the logging to be enabled for proper **Customer Last Activity** rule functioning. To enable the logging follow the instructions below:

1. Navigate to **System > Configuration > Advanced > System > Log**
2. Set **Enable Log** field to 'Yes'
3. Click 'Save Config' button.

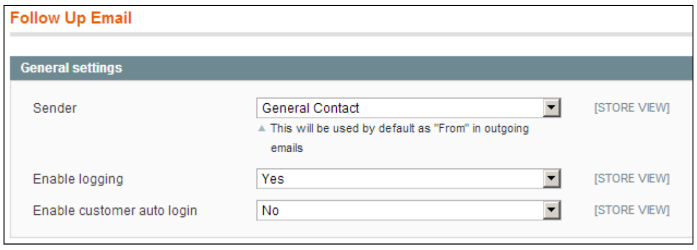
You have now enabled the customers logging.

Configuration

General Configuration

You can configure the extension's general settings in **Admin > System > Configuration > aheadWorks Extensions > Follow Up Email** (or **Follow Up Email-> Settings**)

- **Sender** - defines the default Sender for extension's emails. Note, this option can be overriden in the Rule Settings
- **Enable Logging** - adds the track of modules activity to the aheadWorks general log
- **Enable Customer Auto Login** - this options allows the customers to automatically log in to their accounts. Note, this function only works for the emails with included resume session link.



Sender	General Contact This will be used by default as "From" in outgoing emails	[STORE VIEW]
Enable logging	Yes	[STORE VIEW]
Enable customer auto login	No	[STORE VIEW]

Setting up rules

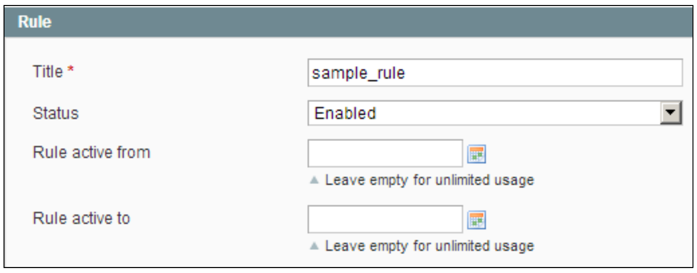
The extension's rules can be managed through **Admin > Follow Up Email > Manage Rules** backend tab. Define rule conditions, then add several tiers to your rule chain. Each tier allows you to choose email template and the number of days for system to wait before sending (for the *Customer birthday* event you may input negative values, pointing that the corresponding email will be generated and sent before the birthday happen).

There are 3 sample rules available after the first-time extension installation only - customer birthday, review request and abandoned cart.

Basic parameters

General Parameters | Events | Email Chain | Conditions

General parameters include Rule **Title** (obligatory), **Status**, **Active From-To** and **Customer Groups** (obligatory)

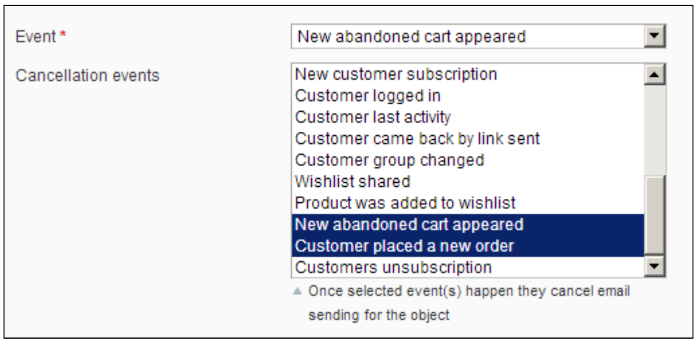


Title *	sample_rule
Status	Enabled
Rule active from	Leave empty for unlimited usage
Rule active to	Leave empty for unlimited usage

Events - this is the most important part of the rule configuration. In the Event selector, you should select the event that will trigger the rule.

You can also select Cancellation event (multiple select is allowed). If any of the specified Cancellation Events occurs, all the currently scheduled emails for this given sendout instance will be removed from the queue.

Below is the list of all the supported events. Each of them can be used as a trigger and for cancellation.


Event * : New abandoned cart appeared

Cancellation events:
- New customer subscription
- Customer logged in
- Customer last activity
- Customer came back by link sent
- Customer group changed
- Wishlist shared
- Product was added to wishlist
- New abandoned cart appeared (selected)
- Customer placed a new order (selected)
- Customers unsubscription

Once selected event(s) happen they cancel email sending for the object

List of events

Event Title	Event Description	Is Cron required?
Order obtained 'X' status	This event captures updates of an order status. Custom Order Statuses are supported as well as the standard Magento ones	Yes
New customer signup	Customer created an account at the store	No
New customer subscription	Customer subscribed to the Newsletter	No
Customer logged in	Customer logged in to their account	Yes
Customer last activity	Works for registered users only. This event will capture any activity of a customer (logging in/ out, ordering, just browsing the store,	Yes

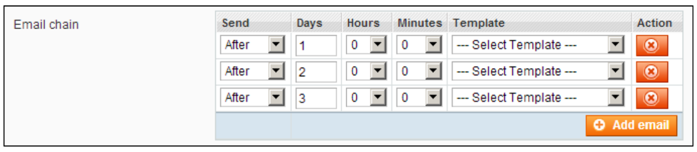
Event Title	Event Description	Is Cron required?
	etc) *NOTE: if used as a trigger, this event IS NOT cancelled by "Customer Logged In". Make sure that you use "Customer Last Activity" as a Cancellation event in such case*	
Customer came back by link sent	This event captures a customer's returning to the store via the link included in some of the previous follow-up emails. By default it is designed to work for abandoned cart reminders, but can also work for any email with the "Restore Session" link included. The links are validated by comparing their unique security code with the security code in the corresponding email from the mail queue. That's why we do not recommend clearing mail queue from messages with a few than 2-3 months age.	No
Customer group changed	Customer was moved to a different Customer Group	No
Wishlist shared	Customer shared their Wishlist	No
Product was added to wishlist	Customer added a new item to their Wishlist. *NOTE: if this event is used as a trigger, you should also select it as a Cancellation Event in the same rule. Otherwise, the customer will get an email for every item they add to wishlist.*	No
New abandoned cart appeared	A new Abandoned Cart appeared at a store. NOTE: Magento considers the cart to be "Abandoned" after 1 hour of customer inactivity	Yes
Customer submitted a review	A registered and logged in customer added a review	Yes
Customer birthday*	Works for registered customers only. Allows setting negative delay in the email queue (i.e., send emails **X days before** birthday)	Yes
Customer placed a new order**	This event is intended as a cancellation for order- and cart-related rules (*Order Obtained 'X' Status, New abandoned cart appeared*). E.g., if you would like to cancel an abandoned cart reminder if a customer orders something, the correct Cancellation Event is "Customer Placed A New Order"	No
Customers unsubscription**	This event cancels the subsequent notifications in case a customer clicks the unsubscribe link	No

**For Trigger Event only; cannot be used as a Cancellation Event*

***For Cancellation Events only; cannot be used as a trigger*

NOTE: The extension requires Cron for sending out emails of any type. Besides, some of the supported events are also processed via Cron (i.e., the email will be scheduled next time the cron runs after the event occurred). Other events are processed instantly (i.e., the email is scheduled immediately when an event occurs). See "Is Cron Required?" column in the table above.

Email chain - in this section, you can add email instances that would be sent to a customer when the rule gets triggered. Note, this section is empty by default, you must add at least 1 email when creating a rule.



Email chain

Send	Days	Hours	Minutes	Template	Action
After	1	0	0	--- Select Template ---	
After	2	0	0	--- Select Template ---	
After	3	0	0	--- Select Template ---	

Conditions

***'SKU'* condition**

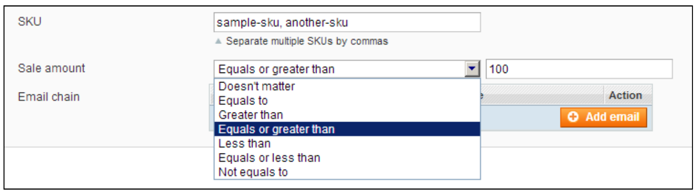
In case you need the rule to get triggered only if some particular product is in customer's order / cart / wishlist, you can specify the comma-separated SKU(s) of the target product(s) in the correspondent field.

***'Sale Amount'* condition**

The *Sale amount* field is applicable to Order or Cart-related events only. It allows you to send emails for only carts/orders which Grand Total field meets the condition specified. If no condition is specified or no value is entered, the condition is treated as empty. Sale amount value should be given in a base currency.

***'Excluded categories'* condition**

If your rule deals with products (items in an order, a product is being added to customer's wishlist, etc.), you can exclude products from particular categories. In case the defined event happens, the object which caused the event is examined for products and, if any of the products belong to at least one of the specified categories - the email is cancelled.

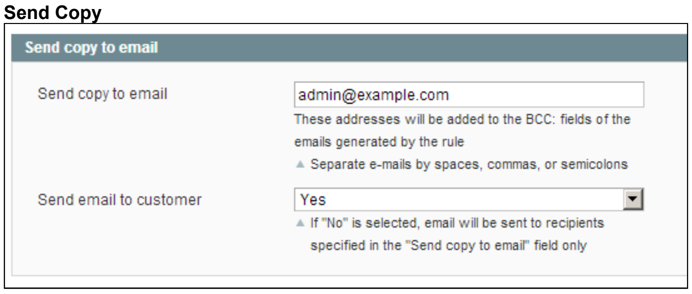


SKU	sample-sku, another-sku Separate multiple SKUs by commas
Sale amount	Equals or greater than — 100
Email chain	

Additional parameters

Send Copy | Sender Details | Stores & Product Types | Newsletter Subscribers | Cross-Sells | Google Analytics

Send Copy



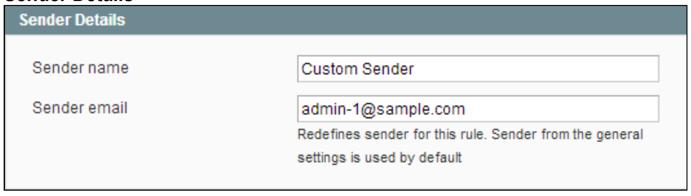
Send copy to email

Send copy to email	admin@example.com These addresses will be added to the BCC: fields of the emails generated by the rule Separate e-mails by spaces, commas, or semicolons
Send email to customer	Yes If "No" is selected, email will be sent to recipients specified in the "Send copy to email" field only

You can add one or several BCC addresses separated by spaces, commas, or semicolons. These email addresses will receive all messages triggered by this rule. Emails will sent only for addresses specified in BCC field when sending emails to a customer is disabled. Also, first email address will be used in "To" email field when FUE not integrated with Custom SMTP and sending emails to a customer is disabled.

Note: *customer email in recipient field is always displayed in email preview even if sending emails to a customer is disabled.*

Sender Details

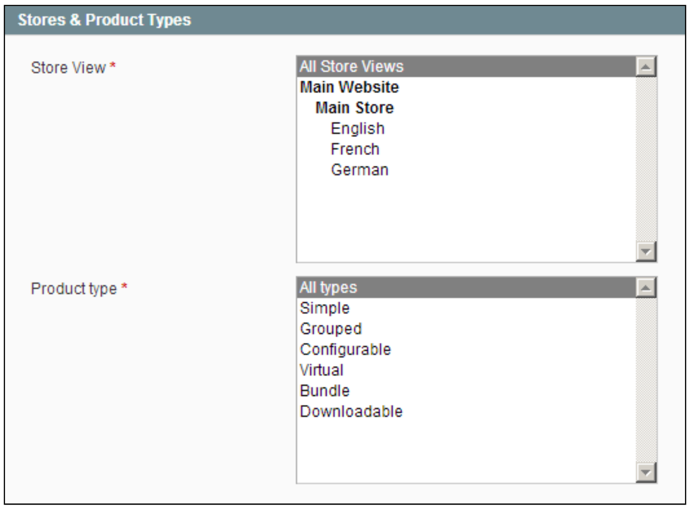


Sender Details

Sender name	Custom Sender
Sender email	admin-1@sample.com Redefines sender for this rule. Sender from the general settings is used by default

Redefines the Sender name and email fields for the rule. Sender from the general settings is used by default.

Stores & Product Types

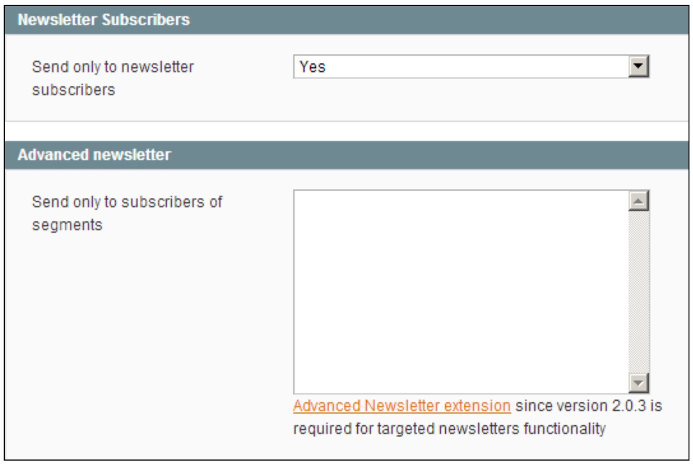


Stores & Product Types

Store View *	All Store Views Main Website Main Store English French German
Product type *	All types Simple Grouped Configurable Virtual Bundle Downloadable

In this tab, you can restrict the rule to work on certain **Store Views** only.
In the **Product Types** selector, you can set a similar restriction based on Product type. Note, this selector will only appear for product-related events (*New Abandoned Cart Appeared, Order Obtained X Status, Wishlist Shared/ Product Added To Wishlist*)

Newsletter Subscribers



Newsletter Subscribers

Send only to newsletter subscribers	Yes

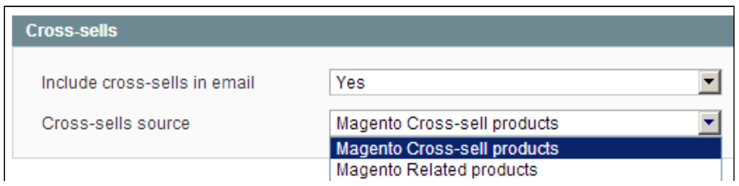
Advanced newsletter

Send only to subscribers of segments	Advanced Newsletter extension since version 2.0.3 is required for targeted newsletters functionality

You can restrict rule execution to the Newsletter Subscribers only.

If you have aheadWorks Advanced Newsletter installed, you will also be able to set restrictions basing on the **Segment** subscription

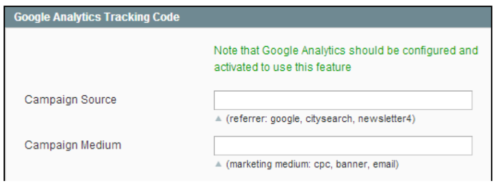
Cross-sells



Cross-sells

Include cross-sells in email	Yes
Cross-sells source	Magento Cross-sell products

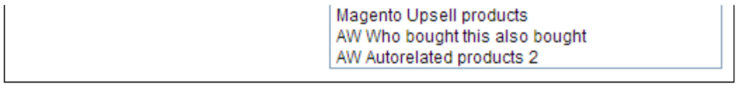
Google Analytics

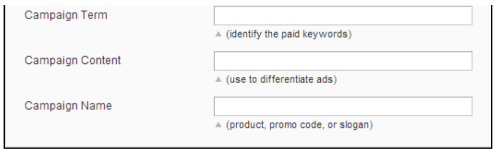


Google Analytics Tracking Code

Note that Google Analytics should be configured and activated to use this feature

Campaign Source	(referrer: google, citysearch, newsletter4)
Campaign Medium	(marketing medium: cpc, banner, email)


Magento Upsell products
AW Who bought this also bought
AW Autorelated products 2


Campaign Term
(identify the paid keywords)
Campaign Content
(use to differentiate ads)
Campaign Name
(product, promo code, or slogan)

The Follow Up Email extension allows you to send emails filled in with items related to purchased products (This feature supports "Order obtained *some* status" and "New abandoned cart appeared" events). The related items can be inserted from the Magento Cross-sell products; Magento Related products; Magento Upsell products; AW Who Bought This Also Bought, or AW Automatic Related Products 2 block.

Note that the AW Who bought this also bought and AW Autorelated products 2 blocks can be added if only the aheadWorks extensions are set up. Products in e-mail will be added from ARP2 and WBTAB blocks which are set to be shown on shopping cart page. ARP2 integration works only since 2.3.1 version.

The following line should be inserted into the template to make the block with related items appear:

```
{{var related}}
```

Note: the variable doesn't work in test emails.

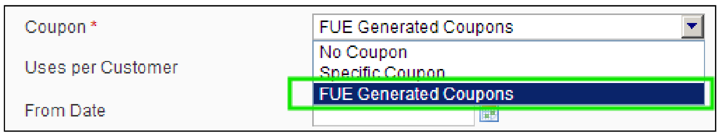
You can change the style of the block with related products in the **/app/design/frontend/<your_package>/<your_theme>/template/followupemail/related.phtml** file.

Google Analytics currently supports 'New abandoned cart appeared' event only.
Note, Google Analytics should be configured and activated in order to use this feature.

Coupons

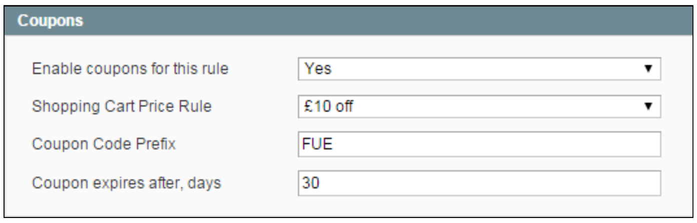
This functionality is available since Magento 1.4.1.0.

FUE allows generating coupons automatically for some rule. At first, you need to create the Shopping Cart Price Rule (**Promotions** > **Shopping Cart Price Rule**) and select **FUE Generated Coupons** value (**Auto** for Magento EE) in the **Coupon** field.



Coupon *	FUE Generated Coupons
Uses per Customer	No Coupon / Specific Coupon / FUE Generated Coupons
From Date	

After this you may select shopping cart price rules which has **Coupon** as **FUE Generated Coupons** in **Shopping Cart Price Rules** field at **Coupons** tab when you editing or creating FUE rule. When the rule gets triggered, new coupon is generated and will be available in email template.


Coupons

Enable coupons for this rule	Yes
Shopping Cart Price Rule	£10 off
Coupon Code Prefix	FUE
Coupon expires after, days	30

Coupon lifetime (i.e., the value specified in **Coupon expires after, days** field) is calculated starting from the moment the rule is triggered, not from the moment the email actually goes out.

If you have any delay specified in the **Email Chain**, make sure to adjust the **Coupon expires after, days** accordingly

To show the coupon code in the emails, make sure that the template contains following code:

```
{{depend var has_coupon}}
<p>Your coupon code is: {{var coupon.code}}, expires at {{var coupon.expiration_date}}</p>
{{/depend}}
```

Also you can add several coupons in the same email - see how

You can include several coupons in a single email, you should just use the construction below in an email template.

The extension will generate and include the correspondent number of coupons. Note, all the coupons will still be generated for the same rule, i.e., *the conditions, discount amount and other parameters will be identical*.

```
{{depend var has_coupon}}
<p>Your coupon code is: {{var coupons.1.code}}, expires at {{var coupons.1.expiration_date}}</p>
<p>Your coupon code is: {{var coupons.2.code}}, expires at {{var coupons.2.expiration_date}}</p>
...
<p>Your coupon code is: {{var coupons.10.code}}, expires at {{var coupons.10.expiration_date}}</p>
{{/depend}}
```

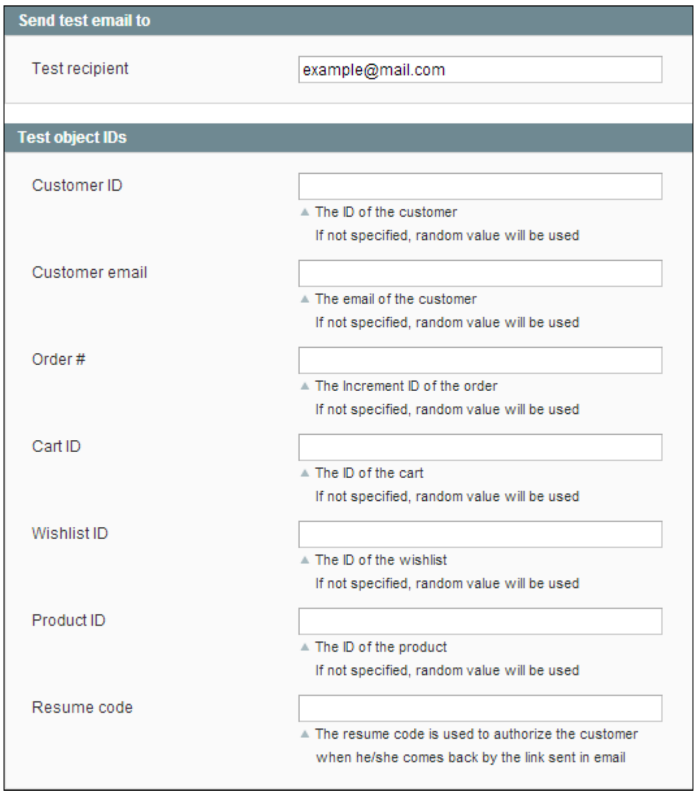
Expired coupons are automatically removed from database by cron jobs. To remove any coupon generated by FUE extension manually, go to the **Follow Up Email** > **Coupons** page.

Note: *the **Stop further rules processing** option doesn't work with FUE Coupons.*

Sending test email

After defining the rule fields you may probably want to see a test email message. You can do so from 'Send Test Email' tab. There you can specify test email recipient and the IDs of the objects that you would like to use as date source for the variables.

Since version 3.5.2 if no values are specified, the random ones will be used. Generated emails will be added to the email queue, and you can preview them from the **Follow Up Email -> Mail Log** backend page.



Send test email to

Test recipient	example@mail.com

Test object IDs

Customer ID	The ID of the customer If not specified, random value will be used
Customer email	The email of the customer If not specified, random value will be used
Order #	The Increment ID of the order If not specified, random value will be used
Cart ID	The ID of the cart If not specified, random value will be used
Wishlist ID	The ID of the wishlist If not specified, random value will be used
Product ID	The ID of the product If not specified, random value will be used
Resume code	The resume code is used to authorize the customer when he/she comes back by the link sent in email

Mail log

The Follow Up Email extension has its own mailing queue, accessible via **Follow Up Email-> Mail Log.**

In the mail log grid, you can view any of the currently scheduled, already sent, failed and cancelled emails. The grid shows the *status*, *created at/ scheduled at/ sent at* timestamps, rule details (*sequence number, rule name, trigger event*) and recipient details (*name/ email address*).

You can also apply certain actions to any of the queue items - ***Preview****, **Cancel**, **Delete** and **Send Now** (the latter sends the selected email(s) instantly, disregarding their schedule or status)*

**Preview action is only available for an individual email. Other actions can be applied to multiple emails at once.*

Integration with other aheadWorks extensions

The Follow Up Email extension features integration with a number of other aheadWorks extensions. If any of the listed modules is installed at your store together with the Follow Up Email, new options will become available.

- Market Segmentation Suite (MSS) - this extension allows you to define filtering rules. While being processed these rules can take into account a set of different conditions regarding customer's properties, their shopping cart, wishlist, previous purchases, etc.You can create a MSS rule and assign it to the Follow Up Email rule. Thus an additional condition is applied to the process of emails scheduling: if an MSS filter is assigned to a Follow Up Email rule, the emails will be scheduled only if MSS rule is validated successfully.
- Advanced Newsletter (since the 2.0.3 version of Advanced Newsletter) extension, which allows using its segments as an extra conditions for the Follow Up Email rules. I.e., a rule can be configured to send email only to the subscribers of specific segments
- Custom SMTP extension, which allows using a custom mailing service for the outgoing emails
- Who Bought This Also Bought and Automatic Related Products 2 (since 2.3.1) extensions allows inserting extra promo blocks in the follow-up. See cross-sells section for details

Create Follow up Email templates

The Follow Up Email extension works with the templates stored in **Newsletter-> Newsletter Templates**. The extension comes with a number of pre-defined templates, and can also use any of the existent Newletter templates.

Template Name*, *Template Subject and ***Template Content*** fields will be taken from the template and processed with the filter. The ***Sender Name*** and ***Sender Email*** fields **are ignored by the extension**, it uses the values specified in the module's General Configuration (or the custom values specified in the rule config).

Variables

The Follow Up Email extension allows using variables in the email templates. In addition to Magento's standard Newsletter variables, the extension adds a number of pre-defined Special Variables, and allows retrieving values from Magento objects.

Special template variables

There are a number of special variables available within the template. These variables can be used "as-is", they require no additional parameters. See reference table below.

List of pre-defined Special Variables

Variable name	Meaning	Example of usage
customer_name	Customer full name, if he/she is registered, or the name taken from the billing/shipping address, or a part of email address before @, or 'Friend'	`Dear {{var customer_name}}!`
customer.firstname	Customer first name, works correct for both - registered and not registered customers	Dear {{var customer.firstname}}!
customer_email	Customer email	`If you'd like to unsubscribe the email {{var customer_email}}, please click here: <a href="...">Unsubscribe</a>`
sequence_number	The number of the email in the chain sequence. When the rule is triggered, the email chain is sorted by its *Send After Days* field, and the *sequence_number* contains the sequence number of the email in the chain.	`This is {{var sequence_number}} email we've sent to you.`
time_delay	The synonym of the *Send After Days* in the chain.	
time_delay_text	The detailed, 'human-like' text representation of the *time_delay* variable. The logic is as shown in the example: ***time_delay*** / **Result**: *time_delay >=1* → in *time_delay* day(s) *1/24 <= time_delay < 1* → in *time_delay*24* hour(s) 1/24/60 <= time_delay < 1/24 → in *time_delay*24*60* minute(s) time_delay <= 1/24/60 → right after All suffixes are configurable by AW_Followupemail.csv locale file located in **app/locale/***selected_language*.	***time_delay*** / **Result**: 12 → in 12 day(s) 1 → in 1 day(s) 0.5 → in 12 hour(s) 0.0416 → in 1 hour(s) 0.0208 → in 30 minute(s) 0,000694 → in 1 minute(s) 0,0005 → right after
time_delay_absolute	The same as time_delay, but for *'Customer birthday'* rule. If email has to be sent before a customer birthday happens - the value of the *Send After Days* field should be negative. The*time_delay_absolute* variable contains the absolute value of the field.	`Let us congratulate you on your birthday that will happen after {{var time_delay_absolute}}days`
url_unsubscribe	The link to unsubscribe	See details
url_unsubscribe_from_all	The link to unsubscribe from all follow up emails	See details
url_resume	The link to resume customer session	See details

Accessing Objects

The extension works with Magento objects (Product, Order, Cart, Customer), and therefore, the module is capable of reading the details stored in an object. The parameter values can be called in a template, generally, a variable should look like **var** *object*.**get***Parameter*() , where *object* and *Parameter* are to be replaced with actual values. For example, var product.getName() variable will display the Product Name in an outgoing email.

Click here to see more examples of object-related template variables

PRODUCT	Example
var product.getProductUrl()	*http://example.com/catalog/product/view/id/51/s/computer_fixed/*
var product.getSku()	*computer_fixed*
var product.getName()	*Gaming Computer*
var product.getPrice()	*4999.95*
var product.getFinalPrice()	*4749.95*
CART	
var cart.getCustomerFirstname()	*John*
var cart.getCustomerLastname()	*Smith*
var cart.getCustomerName()	*John Smith*
var cart.getCustomerEmail()	*john_smith@example.com*
ORDER	

PRODUCT	Example
var order.getBillingAddress().format('html')	*John Smith* *49 North Main Street, United States* *Manchester, NY 14504* *T: (585) 237-4635* *F: (585) 237-4635*
var order.getBillingAddress().getName()	*John Smith*
var order.getCreatedAtFormated('long')	*Jan 8, 2009*
var order.getCustomerName()	*John Smith*
var order.getEmailCustomerNote()	
var order.getShippingAddress().format('html')	*John Smith* *49 North Main Street, United States* *Manchester, NY 14504* *T: (585) 237-4635* *F: (585) 237-4635*
var order.getShippingDescription()	*Flat Rate - Fixed*
var order.status	*Pending*
var order.getStoreGroupName()	*Main Store*
var order.increment_id	*100006581*
OTHER	
var customerName	*John Smith*

For Magento Enterprise Users:

In the Enterprise Edition of Magento, you can use the reward points object in your email templates. Examples:

- *{{var reward.getCurrencyAmount()}} - returns currency amount*
- *{{var reward.getFormatedCurrencyAmount()}} - returns formated currency amount in the currency of the website*
- *{{var reward.getPointsBalance()}} - returns customer's points balance*

Links

In addition to the standard "fixed" links, the Follow Up Email extension allows inserting dynamic links into the emails (e.g., a link to the product a customer has purchased). See below section for details.

Frontend Links | Resume Session Link | Unsubscribe Link

Frontend links

Using standard Magento methods, you can include dynamic links to virtually any Magento frontend area. The approach is based on using the *store* directive (which returns *www.your_domain.com/* if used), supplied with URL postfix and optional parameters.

For example, you can insert a **direct link to a product page** (the *store* directive gives the base part, and the *var* gives the product ID we'd like to treat):

```
<a href="{{store url="catalog/product/view" id="$row_item.product.id"}}">{{var row_item.name}}</a>
```

Do not modify the line of the code above and DO NOT INSERT YOUR STORE URL

In a similar manner, your can insert the **link to the product reviews page**:

```
<a href="{{store url="review/product/list" id="$row_item.product.id"}}">Product reviews</a>
```

You can also create a **link to re-order page** for your registered customers:

```
<p> Please follow the link to reorder <a href="{{store url="sales/order/reorder" order_id="$order.id"}}">link</a></p>
```

This link will work for any order-related rule, and can be used e.g. for order failure notifications (*"Order Obtained 'Cancelled' Status"*), etc

Using such an approach you can reproduce almost all the links used by Magento in its front-end part.

NOTE: the availability of a given URL parameter depends on the type of the rule your are using the template for. For instance, ***product.id*** can only be used for product-related events (*New Abandoned Cart Appeared, Order Obtained X Status, Wishlist Shared/ Product Added To Wishlist*)

"Resume Session" link

The Restore Session URL is intended for the abandoned cart reminders, and by default, it leads customer directly to the ***Cart*** page.

The extension allows inserting special **Resume Session** link into outgoing templates (the link is inserted via a special variable, *url_resume*). When a customer follows that link, their store session will be restored: any items they might have in their Shopping Cart, will be automatically added again. The customers can also be automatically logged in, if this function is enabled in the extension's general settings.

You may insert the link for restoring abandoned carts this way:

```
<a href="{{var url_resume}}">Restore your cart</a>
```

You may also set the optional ***goto*** URL parameter to redirect the customer to the page you want. The URL in the following example will bring the customer to ***Checkout*** page:

```
<a href="{{var url_resume|goto:checkout}}">Restore your cart and go to checkout</a>
```

The value of *goto* parameter will be URL-decoded so if you need to use special chars such as '/' inside of it - make sure you've encoded it correctly.

The extension tracks the customers who return to your store via the Resume Session link. You can view the tracking statistics in **Follow Up Email-> Link Tracking*** backend page.

Note, this page can produce 0 results for completed abandoned cart report, in case Magento log cleaning is enabled in ***Backend -> System -> Configuration -> Advanced -> System -> Log.***

Unsubscribe link

When customers click the unsubscribe link, they get unsubscribed from further messages from that rule. That means that:

- *All the currently scheduled emails from this particular rule to that particular customer will get cancelled when the Unsubscribe link is clicked*
- *The next time the customer triggers that rule, no emails will be created for him*

```
<a href="{{var url_unsubscribe}}">Unsubscribe from emails of such type</a>
```

This link does not unsubscribe customer from Newsletter. If *some other rule* gets triggered, customer will be receiving emails as well.

Unsubscribe from all link

When customers click the unsubscribe from all link, they get unsubscribed from all follow up emails.

```
<a href="{{var url_unsubscribe_from_all}}">Unsubscribe from all emails</a>
```

The unsubscribe from all action URL may also be supplemented with optional *goto* parameter as described in the example above.

Advanced options

The extension's own email templates processor allows using special coding in the templates. See below for the details on available options.

Multi-level directive including

Since version 3.0 the Follow Up Email template processor can evaluate directives nested one inside another in as many levels as you wish. The included directives are processed from the deepest level and from the left to the right. For example, you can use such a construction:

```
{{set a 123}}{{set c {{var {{set b {{var a}}}}}}}}
a = {{var a}}
b = {{var b}}
c = {{var c}}
// returns
a = 123
b = 123
c = 123
```

The 'single back quote' (" ` ") character is used to mark the 'unprocessed' text. It is especially useful in ***set*** and ***eval*** directives described below. To insert a single back quote sign anywhere in the template, use the 'back slash' character (e.g. " \` ").

Custom variables

To give you an opportunity to operate your custom variables inside the template, the new ***set***, ***unset***, and ***eval*** directives were added.

The ***set*** directive creates a variable with the name and value specified:

```
{{set variable_name variable_value}}
```

```
{{set variable_name "long variable value"}}
{{set variable_name 'one more long variable value'}}
{{set variable_name `unprocessed variable value`}}
```

You may embrace long variable value by single or double quotes.

The single back quotes (" ` ") should be used when you don't want the processor to 'dive' in the variable value; in such a case the value will be assigned 'as is'. See example of the ***eval*** directive to learn how to use it.

The ***unset*** directive deletes the variable specified:

```
{{unset variable_name}}
```

For example:

```
{{set a 'One'}}
{{var a}} // returns 'One'
{{unset a}}
{{var a}} // returns ''
```

The ***eval*** directive returns the evaluated value of the variable specified:

```
{{set a `{{set b 1}}b={{var b}}`}}
{{eval a}} //returns 'b=1'
```

Note that in the example above the text enclosed in single back quote characters inside ***set*** directive is assigned to ***a*** variable 'as is', without processing.

Control structures

You can use ***if .. else ..endif*** and ***if .. elseif .. endif*** control structures at your templates. Result of following sample is "Price greater than 100":

```
{{set price 150}}
{{if price>100}}
Price is greater than 100
{{else}}
Price is not greater than 100
{{endif}}
```

This is an example of using if with 'elseif' operator:

```
{{set price 150}}
{{if price<100}}
Price is smaller than 100
{{elseif price>200}}
Price is greater than 200
{{else}}
{{var price}}
{{endif}}
```

This sample prints "150" because variable price is not greater than 200 and not smaller than 100.

Calculating of expressions

Available operators at present time is: substraction (-), addition (+), multiplication (*), division (/), modulo (%), round (round).

Samples:

{{expr 10 + 20}} out 30 as result

{{expr 10.5678 round 2}} out 10.57 as result

Order, Cart or Wishlist item displaying

The ***foreach*** directive lets you 'walk' through any object collection that can be iterated (typically it's intended to walk through the items collection of an Order, a Cart, or a Wishlist objects, depending on the current rule and event context).

The directive syntax is as follows:

```
{{foreach var="object_name" [alias="row_alias_name"] template="template_source:template_name"}}
```

The *object_name* is an items collection object that will be iterated (typically an Items collection of an Order, a Cart, or a Wishlist, depending on the rule context), and the *template* is an email template taken from the Magento email system. **templateSource** defines the source of template: **'email'** means 'Transactional email', and **'nsltr'** means 'Newsletter templates'. 'template_name' is the name of the template. Optional *rowAliasName* parameter is the name of the object presenting the item inside the row template. If *rowAliasName* is omitted, the object inside the directive is referenced to as ***row_item***.

Along with ***row_item*** variable inside each row, there is a counter named as ***row_alias_name*** with ***_row_number*** suffix (i.e. ***row_item_row_number*** by default), representing the number of the row that is currently being iterated. After ***foreach*** directive ends, ***row_item_row_number*** remains among filter variables containing the total number of the rows iterated.

For example, if you want to display an Order items in the table, then your the template can look like this:

```
<table>
 <thead><!-- Define table header here -->
  <tr>
   <th>No</th>
   <th>Product name</th>
   <th>Thumbnail</th>
   <th>Description</th>
   <th>Price</th>
   <th>Qty ordered</th>
   <th>Row total</th>
  </tr>
 </thead>
 <tbody>

{{foreach var="$order.getAllVisibleItems()" alias="order_item" template="nsltr:row_order_item"}}

 </tbody>
 <tfoot><!-- Define table footer here -->
  <tr>
   <th colspan="5">Rows total: {{var order_item_row_number}}</th>
   <th colspan="3">Grand total: {{var order.grand_total|formatPrice}}</th>
  </tr>
 <tfoot>
</table>
```

The template for the table row will look like the following (please note that inside the row template we need to define only row tags, no main table opening/closing tags are applicable here):

```
<tr>
 <td>{{var order_item_row_number}}</td>
 <td><a href="{{store url="catalog/product/view" id="$order_item.productId"}}">{{var order_item.name}}</a></td>
 <td align="center"><img src="{{thumbnail size="75" source="order_item.product"}}"/></td>
 <td>{{var order_item.product.description}}&nbsp;</td>
 <td align="right"><nobr>{{var order_item.price|formatPrice}}</nobr></td>
 <td align="right"><nobr>{{var order_item.qtyOrdered|formatDecimal}}</nobr></td>
 <td align="right"><nobr>{{var order_item.row_total|formatPrice}}</nobr></td>
</tr>
```

Note that inside the {{var }} directive both camelCase (as qtyOrdered used) and glue_by_underlines (as row_total used) variable naming work.

Since an Order, a Cart, or a Wishlist object Items collection often does not contain full information about the appropriate product, a special Product object is re-created and added to each item of the Items collection, thus allowing you to reach all the wealth of the product attributes.

Note that the AW Abandoned cart row, AW Order row, AW Wishlist row options can't be selected in the Template field of the General tab (when a rule is created).

Formatting values

The extension allows changing the representation of certain variables' values. See below for details on available formatting options.

Price values

The new ***formatPrice*** variable modifier formats the value given with current store Currency setting according to its locale. The usage of the ***formatPrice*** is quite simple:

```
{{var price|formatPrice}}
```

Decimal values

The new ***formatDecimal*** variable modifier formats the value given. The usage of the ***formatDecimal*** is as follows:

```
{{var number|formatDecimal}}
```

```
or {{var number|formatDecimal:decimals}}
or {{var number|formatDecimal:decimals:dec_point:thousands_sep}}
```

The modifier accepts either zero, one, or three parameters (not two):

- If no parameters are given, *number* will be formatted without decimals, but with a comma (",") between every group of thousands.
- If one parameter is given, *number* will be formatted with *decimals* decimals with a dot (".") in front, and a comma (",") between every group of thousands.
- If three parameters are given, *number* will be formatted with *decimals* decimals, *dec_point* instead of a dot (".") before the decimals and *thousands_sep* instead of a comma (",") between every group of thousands. Note that only the first character of *thousands_sep* is used

Date & Time values
The new ***formatDateTime*** variable modifier treats the value given as a timestamp and formats it according to the format specified after the first ':' sign:

```
{{var variable|formatDateTime:format}}
```

This modifier allows using the following formats: *full*, *long*, *medium*, *short*.

The example below displays the date of order got updated:

```
{{var order.updatedAt|formatDateTime:long}}
```

Also you can use parameters in format like PHP *date()* function.

Uninstallation

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

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

```
to
```

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

   Now your Magento is unaware of the existence of FUE

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

```
DROP TABLE `aw_followup_config`, `aw_followup_link_track`, `aw_followup_queue`, `aw_followup_rule`, `aw_followup_unsubscribe`
```

   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 and create all rules again after it 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 extension installation I receive a 404 error in System->Configuration->Follow Up Email.

Logout from backend and login back.

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.

I'm trying to test "Abandoned Cart" rule from frontend (by adding items to cart and leaving the store), but I don't receive any emails.

When testing the module like this, an email will go out in 1 hour: Magento considers a Cart to be abandoned after 1 hour of user inactivity.

Customer Last Activity rule doesn't work/no emails created.

Navigate to System > Configuration > Advanced > System > Log and set field **Enable Log** to 'Yes'.

Нравится Станьте первыми кому понравится это Ни одной