

RMA

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

- Installation
- Configuration
 - Initial Configuration
 - Managing Request Types
 - Managing Reasons
 - Managing Request Statuses
- Usage
 - RMA Submission
 - RMA Processing
- Notes
- Uninstallation
- Troubleshooting

Having hard time finding an answer to your question?

Check out our Knowledge Base.

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

Return Merchandise Authorization (RMA) is a transaction whereby the recipient of a product arranges to return goods to the supplier to have the product repaired or replaced; or in order to receive a refund or credit for another product from the same retailer or corporation within the product's warranty period.

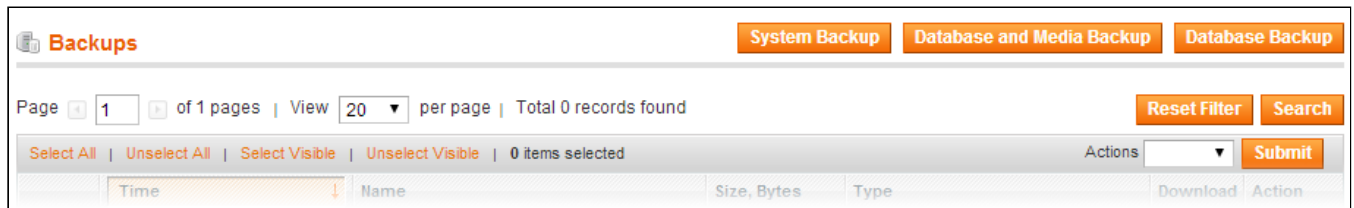
Installation

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



The screenshot shows the 'Backups' section of the Magento admin interface. At the top, there are three orange buttons: 'System Backup', 'Database and Media Backup', and 'Database Backup'. Below these buttons, there is a pagination bar showing 'Page 1 of 1 pages', 'View 20 per page', and 'Total 0 records found'. There are also 'Reset Filter' and 'Search' buttons. Below the pagination bar, there is a table with columns for 'Time', 'Name', 'Size, Bytes', 'Type', 'Download', and 'Action'. The table currently shows 0 items selected.

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

Initial Configuration

[General](#) | [Contacts](#) | [Email Templates](#)

The General Settings of the RMA extension can be configured in **System -> Configuration -> AheadWorks Extensions -> RMA (or Sales -> RMA -> Settings)** backend page.

General tab

- **Allow to request RMA after order completion, days** - this option restricts the period within which an RMA for a given order can be raised
- **Allow guests to request RMA** - enables/ disables Guest RMA form. If this option is disabled, RMA functionality will only be available to registered users.
- **Enable "Print Label" option** - this option allows your customers to easily print out the RMA label with the item details
- **Enable "Reason" option** - enables Reason frontend selector, which
- **Allow "Other" option for reasons** - [works in conjunction with the previous option] defines whether a customer can specify their own reason for an RMA
- **Allow per-order item RMA** - this option enables / disables the possibility to request RMA for a single item out of the whole order. If this option is disabled, the customers will only be able to raise RMAs for the entire order.
- **"Confirm Shipping" is required** - makes it necessary for a customer to Confirm Shipment. After a customer clicks "yes", the RMA status is automatically updated to "Package sent"
- **Confirm Shipping popup text** - the notification text for the above option
- **Forbidden filename extensions** - the extension allows customers to attach files to their requests. This option sets the restriction on the file types acceptable for uploading.
- **Max attachment size, kb** - this option restrict the maximum filesize for the RMA request attachments.

General	
Allow to request RMA after order completion, days *	30 [STORE VIEW]
Allow guests to request RMA	Yes [STORE VIEW]
Enable "Print label" option	Yes [STORE VIEW]
Enable "Reasons" option	Yes [STORE VIEW]
Allow "Other" option for reasons	Yes [STORE VIEW]
Allow per-order item RMA	Yes [STORE VIEW]
"Confirm Shipping" is required	Yes [STORE VIEW]
Confirm Shipping popup text	Having clicked "Yes", you won't be able to cancel RMA or make any changes. Please ensure that: 1. You have included the RMA Label inside the package 2. You have sent the package to the RMA department [STORE VIEW]
Forbidden filename extensions for uploading	php,html [STORE VIEW] ▲ Comma-separated.
Max attachment size, kb	2048 [STORE VIEW]

RMA Policy tab:

Since version 1.5, the RMA extension allows displaying the RMA Policy static block in the Create New RMA

screen.

- **Show policy** - enables/disables the functionality
- **Policy Block** - in this selector, you should pick the policy block, which will be displayed to a customer.

RMA Policy	
Show policy	Yes [STORE VIEW]
Policy block	RMA Policy [STORE VIEW]

How does it look like?

Package Opened * Request Type *

Yes Replacement

Additional Information

RMA Policy

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet.", comes from a line in section 1.10.32.

I agree

Contacts tab:

In this tab, you can specify the Contact details.

- **Enable email notifications** - in this selector, you should select the recipients of the email notifications. The extension can send emails to both Customer and store Admin ("All" option), to customer only, to admin only, or, the notifications can be disabled completely.
- **RMA department display name** - the value specified here will be displayed as a signature in the outgoing email notifications.
- **RMA department email** - the email address of the returns department.
- **RMA department address** - this field is meant for the postal return address. It can be displayed in the RMA status notifications.

Contacts	
Enable email notifications	All [STORE VIEW]
RMA department display name	Customer Service [STORE VIEW]
RMA department email	[STORE VIEW]
RMA department address	[STORE VIEW]

Email Templates tab:

In this tab, you can define the **Email Sender** for the module, and select the email templates* for admin and customer. The extension's default templates are stored in /app/locale/en_US/template/email/awrma/ folder.

- **Base template for customer** - defines an email notification template that will be sent to the customer when store administrator leaves a comment in an RMA request;
- **Base template for admin** - defines an email notification template that will be sent to the store administrator when customer leaves a comment in an RMA request;

Email templates	
Email sender	General Contact [STORE VIEW]
Base template for customer	AW RMA Base template for customer (Default T [STORE VIEW]
Base template for admin	AW RMA Base template for admin (Default Terr [STORE VIEW]

**NOTE: the email templates selected here do not contain actual notifications text. The content is pulled from the Status settings.*

Managing Request Types

Managing Reasons

Request Type is an optional RMA attribute, designed to improve the RMA fulfillment process.

There are 2 request types: "Refund" and "Replacement" by default. You can manage them on the **Sales -> RMA -> Manage Types** page. If no RMA request type is assigned, the "Request type" field is not displayed in the "Request new RMA" form.

Reason is an optional RMA attribute.

Reasons are disabled by default, you can enable this functionality in the module's general settings. Then, you should create the appropriate Reasons in **Sales-> RMA -> Manage Reasons** grid

Managing Request Statuses

You can manage statuses on the **Sales > RMA > Manage Statuses** page.

There are 4 statuses that are present by default ("Pending Approval", "Approved", "Package Sent", "Resolved (canceled)"). These statuses can't be deleted. You can only change their templates, name, or sort order, but they will still perform the original function.

The following fields are available while editing status:

Main Settings tab

- **Name** - status name
- **Resolve RMA on obtaining status** - sign of the request completion when this status is obtained. Resolved RMAs cannot be further updated.
- **Sort Order** - this option allows you to arrange the Statuses in a list according to your preferences

Store Templates tab

- **Notification sent to customer (leave blank not to send)** - text of notification which will be sent to customer
- **Notification sent to administrator (leave blank not to send)** - text of notification which will be sent to administrator
- **Notification sent to messages history (leave blank not to send)** - text which will be added to the Comments chatbox (all HTML formatting will be escaped)

Template Variables

The **{{var request}}** variable contains current RMA request and can be used in templates with the following methods:

- **{{var request.getAdminUrl()}}** - returns backend URL to manage RMA request
- **{{var request.getCustomerName()}}** - returns customer name (e.g. John Doe)
- **{{var request.getCustomerEmail()}}** - returns customer email
- **{{var request.getFormattedCreatedAt()}}** - returns the date of RMA request creation

- `{{var request.getCustomerStreetAddressFromPrintLabel()}}`
- customer address from label printed
- `{{var request.getNotifyRmaAddress()}}` - returns RMA department address
- `{{var request.getOrderid()}}` - returns order increment ID
- `{{var request.getPackageOpenedLabel()}}` - returns package opened label (Yes/No)
- `{{var request.getRequestTypeName()}}` - returns request type name
- `{{var request.getStatusName()}}` - returns current status name
- `{{var request.getTextId()}}` - returns formatted RMA request ID (e.g. #0000000007)
- `{{var request.getUrl()}}` - returns frontend URL to manage RMA request

You can also add a store-specific set of templates (via **Add Store Template** button in the page bottom). This will add an extra section with the Store View selector, Name field (the value will override the Display Name specified in the **General Settings**) and 3 notification message fields

Usage

RMA Submission

The RMA extension allows raising an RMA request through 3 channels:

1. **Request from a registered user** can be posted from the special **RMA tab** their account dashboard. The RMA can also be requested from the **MyOrders -> View Order** screen.
2. **Request from a guest user** (if this functionality is allowed in **module's settings**) - guests can post RMAs from a separate form. When a guest enters the form, he must specify the **email address** used for ordering and the **order ID**. After these details are supplied, the further procedure is the same as for the registered customers
3. **Admin-initiated requests** - store admins can also open RMAs on customers' behalf. It can be done either via **Add New** button in **Sales-> RMA-> All RMA** grid, or via **Create Request From This Order** button in **Sales-> Orders-> View Order-> RMA Requests** screen

RMA Processing

The RMA extension only allows to track and manage the requests. The extension does not have any provisions for auto-fulfillment. I.e., actual sending replacement items, issuing refunds, etc must be performed by the store admin manually.

Once a new RMA gets posted, it obtains "**Pending Approval**" status, and gets listed under **Sales-> RMA-> Pending Requests**. The request can be further processed through the **Edit RMA** screen.

The main tabs are Request Details and Request Options.

- **Request Details** section contains the links to the **Order**, **Customer Account**, and the **External View URL** for the request.

Pay attention to the **ID** editable field - all the RMA requests are given simple increment IDs by default. The ID of a given RMA can be manually changed to any desired value via this field.

- **Request Options** section allows to update the RMA state. *Approving, Resolving, Rejecting, etc* is done by setting the correspondend status in "**Set Status To**" selector.

Other options include miscellaneous activities: changing request reason, adding request Comments (visible to customer) and Notes (visible to admin users only), adjusting Customer Address, etc

Request Details	
ID	#0000000025
Order ID	#100000027
Customer Name	John Doe
External URL	3E343F9152E6463DBD55E
Request Options	
Package Opened	Yes
Set status to	Pending Approval
Request type	Replacement
Post tracking code	

Notes

- Admin doesn't receive notifications for events which he initiated;
- Admin's notes, additional info for printing label, and comments should not contain HTML tags
- Order increment ID`s in comments will be automatically converted to links for order
- A new RMA request considers the qty of the refunded items in the selected order and the qty of other RMA requests for the selected order. The items from the resolved RMA request can be used for a new RMA request.
- If a **custom Footer block** is in use, the extension may fail to add the link to the Guest submission form. This link can be inserted manually, the URL is www.your_domain.com/index.php/awrma/guest_rma/index/

Uninstallation

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

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

to

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

Now your Magento is unaware of the existence of RMA extension

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_rma_entity_comments`, `aw_rma_entity_status`, `aw_rma_entity_types`, `aw_r
```

Note, if you are using the table with prefixes, you must specify them in all table names.

If you are not sure how to do that or expect any troubles with it, please contact your server administrator regarding the matter.

If you remove the tables as it is described above, you will need to configure the extension again after it is reinstalled. All the extension-specific data will be lost .

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
это

Станьте первыми кому понравится

Ни одной