



Order Manager for Magento 1

A few quick notes before installing your extension:

- It's a great idea to [install any new extension on a test site](#) before adding it to your live site. Once you have successfully installed the new extension on your test site and are happy with its performance, repeat the extension installation on your live site. Distributed version control systems (such as git or mercurial) can be used to sync live and test sites with minimal effort
- The instructions below are for basic setups (one server which provides Apache, Mysql and PHP). For more advanced setups, the steps are still applicable but need to be adjusted to your particular setup. Your system administrator or web host can provide you with details
- [We don't recommend installing extensions via Magento Connect](#) - please follow the instructions below

If you prefer, [we can install the extension for you](#) for NZ\$49 (24-48 hour turnaround time, Monday-Friday).

Summary

[Step 1 - Turn off Magento's Compilation Mode](#)

[Step 2 - Turn on Magento's Configuration Cache](#)

[Step 3 - Install the Extension](#)

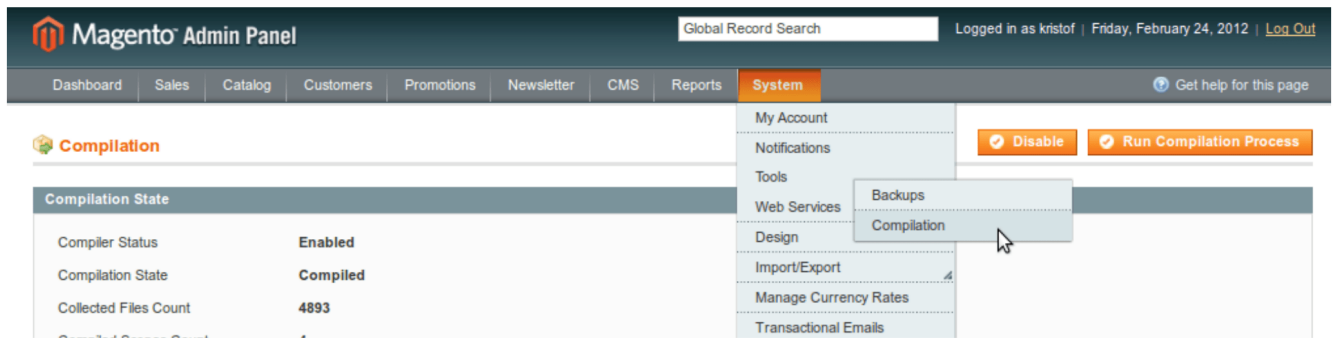
[Step 4 - Refresh Magento's Cache](#)

[Step 5 - Log Out of the Backend and Log Back In](#)

[Step 6 - Rerun Magento's Compilation Process](#)

Step 1 - Turn Off Magento's Compilation Mode

Check to see if Magento's compilation mode has been enabled. If it has, turn it off before installing an extension or making any code changes (go to **System > Tools > Compilation**). In a worst case scenario, not disabling compilation mode before installing an extension could make your site inaccessible (including the admin area). Yikes!



Command Line Equivalent: `php -f shell/compiler.php -- clear`

As part of this step we also recommend:

- **Backing up your Magento files** so you can revert back to a previous version if something unexpected happens during installation. If this is your first backup, take additional care to verify that the backup is both complete and functional by performing a trial run of your recovery process
- **Refreshing the cache** to ensure your site is running up-to-date code, refresh your Magento cache (see Step 4). Verify that your current site works as expected ([this is a good list of what to look for](#)). If you identify any issues with your existing site, it's much easier to resolve these before adding a new extension
- [Taking your store offline](#) before taking a database backup (below). If we kept the store online and needed to rollback to this backup, we would lose any orders placed after the backup was taken. The alternative to this approach would be to keep the store live, but to re-add those orders manually if you needed to revert back to the old database. To take your site offline to customers, but retain access to the backend yourself, you can redirect users (excluding yourself) to a static page saying that the site is currently unavailable but will be back online shortly
- [Creating a database backup](#)

Step 2 - Turn On Magento's Configuration Cache

Turn on Magento's configuration cache – this will prevent Magento from trying to install the new extension too early. If the extension needs to make any database changes, triggering the installation too early could miss some database changes and create errors which are difficult to fix later on.

Magento Admin Panel Global Record Search Logged in as kristof | Friday, February 24, 2012 | [Log Out](#)

Dashboard Sales Catalog Customers Promotions Newsletter CMS Reports **System** Get help for this page

Cache Storage Management

Select All | Unselect All | Select Visible | Unselect Visible | 0 items selected

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	DISABLED
☐ Blocks HTML output	Page blocks HTML.	BLOCK_HTML	DISABLED
☐ Translations	Translation files.	TRANSLATE	DISABLED
☐ Collections Data	Collection data files.	COLLECTION_DATA	DISABLED
☐ EAV types and attributes	Entity types declaration cache.	EAV	DISABLED
☐ Web Services Configuration	Web Services definition files (api.xml).	CONFIG_API	DISABLED

Additional Cache Management

- My Account
- Notifications
- Tools
- Web Services
- Design
- Import/Export
- Manage Currency Rates
- Transactional Emails
- Custom Variables
- Permissions
- Magento Connect
- Cache Management**
- Index Management
- Manage Stores

Flush Magento Cache Flush Cache Storage

Actions Submit

Step 3 - Install the Extension

3.1 Download the extension, either from [your account on our website](#) or via the direct link sent to you in the order confirmation email. Once you have the files on your computer unzip the extension files into a separate folder.

3.2 Upload the extension by uploading the unzipped files to your Magento server via FTP or SFTP (an excellent option is to use the free opensource program [Filezilla](#)). To do this:

1. Connect to the server
2. Navigate to your Magento root folder
3. Upload the previously unzipped files to your root folder by dragging them from your local computer to your Magento root folder

Host: Username: Password: Port: Quickconnect

Response: Current directory is: "/home/magento3"
 Status: Directory listing successful
 Status: Retrieving directory listing...
 Command: ls
 Status: Listing directory /home/magento3

Local site: /Users/admin/Desktop/Fooman Extensions/fooman_speedster-3.0.14/ Remote site: /home/magento3

Filename	Filesize	Filetype	Filename	Filesize	Filetype	Last modified	Permissions
..		File folder	..		File folder	12/1/2014 4:48:...	drwxr-xr-x
app		File folder	BACKUP		File folder	7/3/2014 10:50:...	drwxr-xr-x
lib		File folder	cgi-bin		File folder	7/3/2014 10:50:...	drwxr-xr-x
skin		File folder	fcgi-bin		File folder	7/3/2014 10:50:...	drwxr-xr-x
var		File folder	FILES		File folder	10/27/2014 7:1...	drwxr-xr-x
			homes		File folder	7/3/2014 10:50:...	drwxr-xr-x
			logs		File folder	7/3/2014 10:50:...	drwxr-xr-x
			public_html		File folder	12/19/2014 4:3...	drwxr-xr-x
			skin		File folder	12/19/2014 3:5...	drwxr-xr-x
			tmp		File folder	12/19/2014 4:2...	drwxr-xr-x
			.bash_logout	18	BASH_LOG...	7/3/2014 10:50:...	-rw-r--r--

Host: Username: Password: Port: Quickconnect

Status: Retrieving directory listing...
 Command: pwd
 Response: Current directory is: "/home/magento3"
 Status: Directory listing successful

Local site:	Remote site:
C:\Users\admin\Desktop\Fooman Extensions\fooman_speedster-3.0.14\	/home/magento3/public_html

Filename	Filesize	Filetype	Last modified
..			
app		File folder	12/19/2014
lib		File folder	12/19/2014
skin		File folder	12/19/2014
var		File folder	12/19/2014

Filename	Filesize	Filetype	Last modified
..			
app		File folder	12/19/2014 4:3...
downloader		File folder	11/26/2013
errors		File folder	11/26/2013
includes		File folder	11/26/2013
js		File folder	12/18/2014 11:...
lib		File folder	12/11/2014 8:4...
media		File folder	12/11/2014 3:3...
pkginfo		File folder	11/26/2013
shell		File folder	11/26/2013
skin		File folder	12/19/2014 4:3...
var		File folder	12/13/2014 12:...
.htaccess	5,673	HTACCESS...	12/11/2014 3:3...
.htaccess.sample	4,568	SAMPLE File	11/26/2013
api.php	2,834	PHP File	11/26/2013
cron.php	2,831	PHP File	11/26/2013
cron.sh	717	SH File	11/26/2013
favicon.ico	1,150	Icon	11/26/2013

Having issues locating your Magento Root Folder? Check [this FAQ](#).

3.3 Check the file permissions of the newly uploaded files: Files need to be readable by the webserver. Pay particular attention to the user and group of the files, and to the files' read/write permissions. It's not possible to give a generic recommendation for which settings to use because it depends entirely on your individual webserver setup. Compare file permissions of the newly uploaded files and folders against existing files on the server. Alternatively, ask your webhost or server administrator for the best settings to use.

Filename ^	Filetype	Filesize	Last modified	Permissions	Owner/Group
..					
app	Directory		20/02/10	drwxr-x—	apache apache
downloader	Directory		20/02/10	drwxr-x—	apache apache
errors	Directory		20/02/10	drwxr-x—	apache apache
includes	Directory		20/02/10	drwxr-x—	apache apache
js	Directory		20/02/10	drwxr-x—	apache apache
lib	Directory		20/02/10	drwxr-x—	apache apache
media	Directory		14/08/08	drwxr-x—	apache apache
pkginfo	Directory		20/02/10	drwxr-x—	apache apache
shell	Directory		20/02/10	drwxr-x—	apache apache
skin	Directory		20/02/10	drwxr-x—	apache apache
var	Directory		05/09/10	drwxr-x—	apache apache

Selected 1 directory.

The ideal scenario is to give the server the absolute minimum access rights needed to perform a given task. Granting universal read/write/execute (777) permissions can be unsafe and is generally not recommended.

Step 4 - Refresh Magento's Cache

Refreshing the Magento cache allows Magento to register the addition of the new extension

and to run any necessary installation routines. Alternatively, turning off the cache will have the same effect.

Cache Storage Management

Select All | Unselect All | Select Visible | Unselect Visible | 0 items selected

Cache Type	Description
☐ Configuration	System(config.xml, local.xml) and modules configuration files(config.xml).
☐ Layouts	Layout building instructions.
☐ Blocks HTML output	Page blocks HTML.
☐ Translations	Translation files.
☐ Collections Data	Collection data files.
☐ EAV types and attributes	Entity types declaration cache.
☐ Web Services Configuration	Web Services definition files (api.xml).

Additional Cache Management

My Account
Notifications
Tools
Web Services
Design
Import/Export
Manage Currency Rates
Transactional Emails
Custom Variables
Permissions
Magento Connect
Cache Management
Index Management
Manage Stores

Flush Magento Cache | Flush Cache Storage

Associated Tags | Status

Associated Tags	Status
CONFIG	ENABLED
LAYOUT_GENERAL_CACHE_TAG	DISABLED
BLOCK_HTML	DISABLED
TRANSLATE	DISABLED
COLLECTION_DATA	DISABLED
EAV	DISABLED
CONFIG_API	DISABLED

Step 5 - Log Out of the Backend and Log Back In

Log out and back into Magento to reload any access rules for the backend and prevent a potential “Access Denied” error on newly added admin areas that you haven’t been granted access to. Logging back in with a full administrator account will grant you the new permissions required.

Check your site to make sure your newly installed extension works as advertised, and the rest of your store still behaves as you would expect (you can use the same [checklist](#) we linked to earlier).

If you took your store offline before installing the extension, [bring it back online](#).

Step 6 - Rerun Magento’s Compilation Process

If you're using Magento’s compilation mode, you need to hit the “Run Compilation Process” button to update the compiled state with the newly installed extension (go to **System > Tools > Compilation**).

Compilation

Run Compilation Process

Compilation State

Compiler Status	Disabled
Compilation State	Not Compiled

Command Line Equivalent:
`php -f shell/compiler.php -- compile`

Installation Complete!

Troubleshooting

If you run into any issues, check out the [Installation Help](#) FAQ.

"Access denied" for OrderManager actions after applying SUPEE-6285

Magento's security patch SUPEE-6285 has prevented some functionality of Fooman Order Manager (v3.0 and below) from working correctly, resulting in "Access Denied" permission issues.

To fix this issue, there are two steps:

1. Make the code changes described below
2. Make sure that all users, including any limited access roles, have access to **Sales > Orders** if you want these roles to be able to perform the mass actions

OrderManager v2.1.0 and under

In `/app/code/community/Fooman/OrderManager/controllers/OrderController.php`

Add

```
protected function _isAllowed()
{
    return Mage::getSingleton('admin/session')->isAllowed('sales/order');
}
```

right after

```
class Fooman_OrderManager_OrderController extends Mage_Adminhtml_Controller_Action {
```

For OrderManager 2.1.0 to 2.1.x

In /app/code/community/Fooman/OrderManager/controllers/Sales/OrderController.php

Add

```
protected function _isAllowed()
{
    return Mage::getSingleton('admin/session')->isAllowed('sales/
order');
}
```

right after

```
class Fooman_OrderManager_Sales_OrderController extends Mage_Adminhtml_Controller_Action {
```

For OrderManager 2.2.0 to 2.3.x

In

/app/code/community/Fooman/OrderManager/controllers/Sales/OrderManagerController.php

Add

```
protected function _isAllowed()
{
    return Mage::getSingleton('admin/session')->isAllowed('sales/
order');
}
```

right after

```
class Fooman_OrderManager_Sales_OrderManagerController extends Mage_Adminhtml_Controller_Action
{
```

Issues with Magento Workflow

If you see an error message related to the Magento workflow, the first thing to do is to confirm whether the error is being caused by an underlying issue with the standard workflow. If Magento's standard workflow is not functioning correctly, then this error will also be present when using Fooman Order Manager.

Troubleshooting steps:

1. Disable Fooman Order Manager by editing
app/etc/modules/Fooman_OrderManager.xml to change true to false

2. Save and refresh the cache
3. Repeat the workflow action which is giving the error message to see if you get the same error message with Fooman Order Manager disabled
4. Re-enable Fooman Order Manager by changing `app/etc/modules/Fooman_OrderManager.xml` back to true. Save and cache refresh

If you see the same error even with Fooman Order Manager disabled, this indicates a problem with the standard Magento workflow that will need to be resolved independently of Fooman Order Manager.

If you don't see the same error with Fooman Order Manager disabled, please contact us for help.

Reporting Any Issues/Bugs

We are proud of our quality extension code - it's been widely tested and we stand by it 100%. If something does happen and you think you might be experiencing an issue or bug, please contact us via support@fooman.co.nz and we will help you out.