

ULTIMATE  
THE **GUIDE**

*to*

# Installing Magento Extensions

# Welcome

This best practice guide contains universal instructions for a smooth, trouble free installation of any Magento extension - whether by Fooman or another developer, and whether a free or commercial extension. The guide is based on 4 years of experience supporting 100,000+ Magento extension downloads.

So what do we mean by universal instructions? Certain extensions may not strictly require every installation step to be performed for a successful installation. But including all 6 best practice steps in your standard installation routine will ensure a robust process that can be consistently applied to every installation, every time. Any 'additional' steps won't affect the installation.

For some installation steps, there are different ways of achieving the same goal. In this guide we use the easiest option using Magento's backend (where possible).

This guide will evolve to keep up with best practices for Magento and feedback we receive. Click [here](#) to access the latest version of this guide.

Best of luck with your installation!



**Kristof @ Fooman**  
July 2013

# Summary of the Steps

Step 1 - Turn off Magento's Compilation Mode .....	7
Step 2 - Turn on Magento's Cache .....	13
Step 3 - Install the Extension .....	15
Step 4 - Refresh Magento's Cache .....	21
Step 5 - Log Out of the Backend and Log Back In .....	23
Step 6 - Rerun Magento's Compilation Process .....	27
Troubleshooting .....	30

# How to Use this Guide

## Users with Basic Magento Knowledge

Follow the step by step instructions with screenshots to install the extension.



### Optional Additional Steps to include a 'Safety Net'

Follow these additional optional steps if you want to include a safety net with your installation.



### Hints and Tips

Tip boxes give you extra advice along the way.

# How to Use this Guide

## Users with Advanced Magento Knowledge

This guide is written with a basic webserver setup in mind (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.

It makes sense to automate the installation and deployment process using a command line equivalent to the Magento backend instructions we provide. Use this guide to make sure all steps are covered when setting this up.



### Command Line Equivalents

This symbol indicates that Magento provides a built-in command line equivalent to the backend instructions.

# How to Use this Guide



## Using a Test Site

The best advice we can give you is to install any new extension on a test site before adding it to your live site.

A test site, based on a copy of your current live store, can be used to test new settings or extension installations and make sure you are 100% happy with the outcome before allowing the changes to go live.

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.

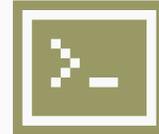
# Step 1

Turn off Magento's  
Compilation Mode

# Step 1 - Turn off Magento's Compilation Mode

Check to see if Magento's compilation mode has been enabled. If it has, it's vital to 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
```

The screenshot shows the Magento Admin Panel interface. At the top, there is a navigation bar with the following tabs: Dashboard, Sales, Catalog, Customers, Promotions, Newsletter, CMS, Reports, and System (which is currently selected). To the right of the navigation bar, there is a search box labeled "Global Record Search" and a user status indicator: "Logged in as kristof | Friday, February 24, 2012 | [Log Out](#)".

Below the navigation bar, the main content area is titled "Compilation". On the right side of this area, there are two buttons: "Disable" and "Run Compilation Process".

On the left side, there is a table titled "Compilation State" with the following data:

Compilation State	
Compiler Status	Enabled
Compilation State	Compiled
Collected Files Count	4893
Compiled Classes Count	1

On the right side of the main content area, there is a vertical menu with the following items: My Account, Notifications, Tools, Web Services, Design, Import/Export, Manage Currency Rates, and Transactional Emails. A sub-menu is open under "Tools", showing "Backups" and "Compilation". A mouse cursor is pointing at the "Compilation" option.

# Step 1 - Optional



## Back up your Magento Files

Creating a backup will allow you to revert back to a previous version of your Magento files if something unexpected happens during the extension 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.



## Refresh the Cache and Check your Site

To ensure your site is running up to date code, refresh your Magento cache (see [Step 4](#)). Next, verify that your current site works as expected (Tom Robertshaw provides an [extensive checklist](#) if you need ideas).

If you identify any issues with your existing site, it's much easier to resolve these before introducing new complexities by installing a new extension.

# Step 1 - Optional



## Take your Store Offline

The reason for taking your store offline is that we will be taking a database backup in the next optional 'safety net' step. 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. To do this, create a simple html file called "maintenance.html" (your maintenance notice) in your main Magento folder.

Next, copy and paste the following lines to the top of your .htaccess file:

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
<FilesMatch (maintenance\.html)$>
    Order allow,deny
    Allow from all
</FilesMatch>
ErrorDocument 403 /maintenance.html
```

Adjust the IP address (shown above as 127.0.0.1) to your own. You can check the IP address that you are currently using with this [Google search](#).

Further down in the same .htaccess file, change

```
#####  
## By default allow all access  
    Order allow,deny  
    Allow from all
```

to

```
#####  
## By default allow all access  
    #Order allow,deny  
    #Allow from all
```

This will redirect everyone except yourself (using your specified IP address) to the maintenance.html file.

Note: These instructions assume that you have a standard webserver setup. If your webserver serves pages directly to the visitor and doesn't sit behind another server (eg. Varnish), this could change the IP address a request appears to be coming from and the above code won't work. It also assumes that your store complies with Magento's system requirement "Ability to override options in .htaccess files".

From Magento version 1.4 onwards, you can place your store into maintenance mode by creating an empty file called "maintenance.flag" in your Magento root folder. Visitors to the site will be greeted by a "service temporarily unavailable" notice. Unfortunately, the maintenance flag also makes the site (including the backend) unavailable to you, so we don't recommend this option.

# Step 1 - Optional



## Create a Database Backup

The easiest option for creating a database backup is to use Magento's inbuilt tool. Go to **System > Tools > Backups** and select "Create Backup". If this is your first backup, it's a good idea to verify that your backup is working correctly on a separate database.

The screenshot shows the Magento Admin Panel interface. At the top, there's a navigation bar with tabs for Dashboard, Sales, Catalog, Customers, Promotions, Newsletter, CMS, Reports, and System. The System tab is active. Below the navigation bar, there's a search bar and a 'Global Record Search' field. The main content area shows the 'Backups' section. On the right side of this section, there's a 'Create Backup' button. Below the button, there are 'Reset Filter' and 'Search' buttons. The 'System' menu is open, showing options like My Account, Notifications, Tools, Web Services, Design, Backups, and Compilation. The 'Backups' option is highlighted with a mouse cursor.

You can also use other tools such as mysqldump (via the command line) or phpmyadmin to take a backup of your database. If you use phpmyadmin, be sure to tick the "Disable Foreign Key Constraints" box to be able to properly restore the backup.

Depending on your database size and server speed, this may take a while. Once the backup is finished, you will find the zipped backup under var/backups:



# Step 2

Turn on Magento's Cache

# Step 2 - Turn on Magento's 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.

**Cache Storage Management**

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

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

**Additional Cache Management**

Flush Magento Cache | Flush Cache Storage

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

# Step 3

## Install the Extension

# Step 3 - Install the Extension

## Always read the installation instructions that came with the extension!

Although you may have read many installation instructions before, specific instructions or caveats might be given for individual extensions which are essential for a successful installation or update.

## Download the extension

Download the extension, preferably as a zip file. If an extension is only available on Magento Connect, you can download it directly via this [site](#) (use with a Magento Connect 2.0 key ). Once you have the files on your computer unzip the extension files into a separate folder.

### Download Magento Extension

Enter the extension key from [magento connect](#) to download the extension archive file in latest version, without PEAR.

A "ready to paste" version is generated, just unzip the archive file content in your Magento installation for install it ;)

**New !** Now compatible with Magento Connect 2.0 keys.

[download FooMan\\_Speedster-2.0.9.tgz \(66.3 KB\)](#)

Magento Connect 2 provide already a ready to paste version in this tarball.



## Alternative Installation Options (Advanced)

- Download via command line with [Magento's pear command](#)
- [Modman](#)
- [Composer](#)

# Step 3 - Install the Extension



## Installing Extensions via Magento Connect

It's also possible to install extensions using Magento's inbuilt Magento Connect Manager, or using Magento Connect via the command line.

While the concept of Magento Connect is appealing and often seems to be the easiest option for installing a new extension, we generally advise against using it.

There are a couple of reasons for this:

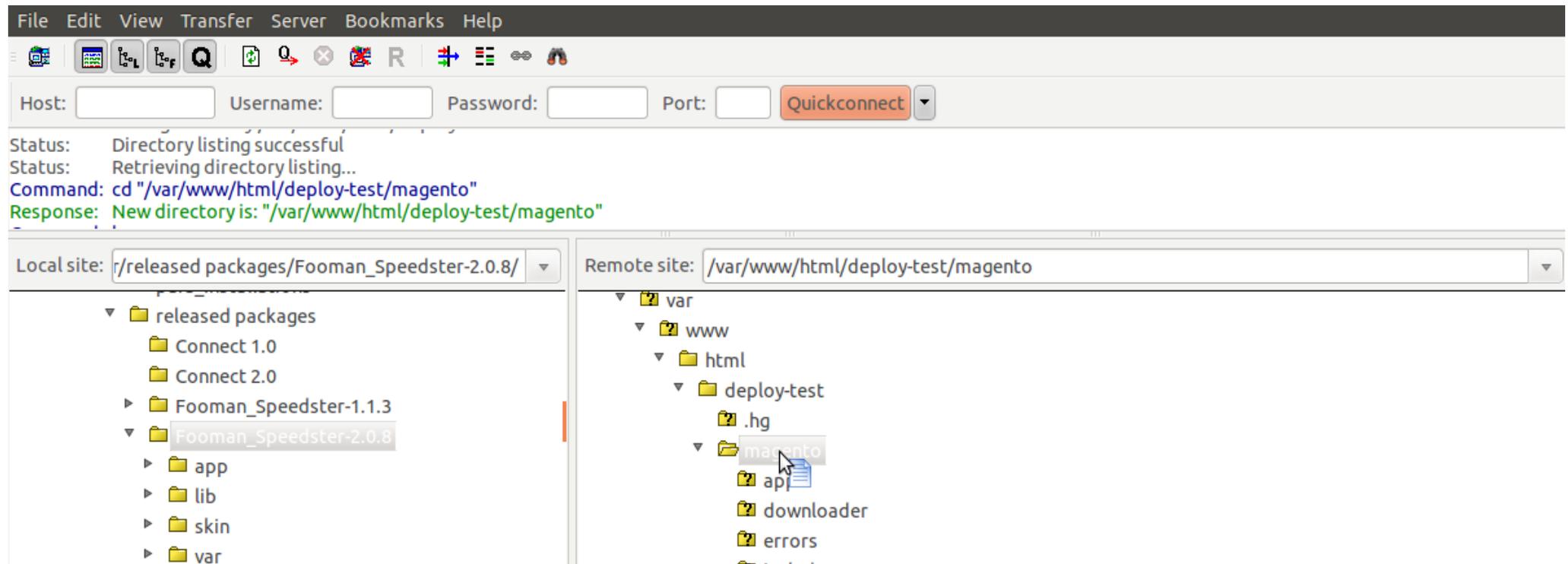
- Installing an extension with the same key at different times could install a newer version of the extension (which you did not test in your test environment)
- Getting server file permissions right can be complex, and too many people resort to giving unsafe read/write/execute permissions to everyone (777) to get Magento Connect Downloader to work
- The extension is downloaded and executed in one step, without giving you a chance to inspect the code first. Unfortunately not all extension developers are [trustworthy](#)
- Occasionally, installing an extension with Magento Connect Manager can trigger an upgrade of the Magento core itself... which you might not be prepared for!

# Step 3 - Install the Extension

## Upload Extension

Next, upload 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.



# Step 3 - Install the Extension



## Locating your Magento Root Folder

The location of the Magento root folder depends on where you originally installed Magento, and is also driven by your webhosting company. Common folder locations to check are:

```
/
/public_html
/www
/var/www/
/home/YOURUSERNAME/www
/home/YOURUSERNAME/public_html
/srv/
...
```

You know you've found the right folder when you see the following subfolders:

```
app
downloader
errors
includes
```

If you can't find the Magento root folder, get in touch with your webhosting company and ask them to help you locate it. In most cases, it is the same as the document root.

# Step 3 - Install the Extension

## 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.



### Tips for Permission Settings

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

# 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.

The screenshot displays the Magento Admin Panel interface. At the top, the 'System' menu is expanded, with 'Cache Management' selected. The main content area is titled 'Cache Storage Management' and contains a table of cache types. The table has columns for 'Cache Type' and 'Description'. Below the table, there are buttons for 'Flush Magento Cache' and 'Flush Cache Storage'. The 'Cache Management' menu is also visible, showing options like 'Index Management' and 'Manage Stores'.

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

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 &  
Log Back In

# Step 5 - Log Out of the Backend & 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.



The image shows the Magento Admin Panel login interface. On the left side, there is a vertical logo for Magento. The main content area is titled "Log in to Admin Panel". Below the title, there are two input fields: "User Name:" and "Password:". Below the "User Name:" field, there is a link that says "Forgot your password?". To the right of the "Password:" field, there is an orange "Login" button. At the bottom of the form, there is a footer that reads "Magento is a trademark of Magento Inc. Copyright © 2013 Magento Inc."

# Step 5 - Optional



## Check your Site

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



## Bring your Store Back Online

If you took your store offline before installing the extension, bring it back online. To do this, delete the following lines from the top of your `.htaccess` file (where `127.0.0.1` is your own IP address):

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
<FilesMatch (maintenance\.html)$>
    Order allow,deny
    Allow from all
</FilesMatch>
ErrorDocument 403 /maintenance.html
```

Further down in the same .htaccess file, change

```
#####  
## By default allow all access  
  #Order allow,deny  
  #Allow from all
```

to

```
#####  
## By default allow all access  
  Order allow,deny  
  Allow from all
```

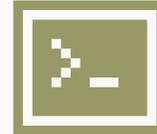
Your store will now be back online.

# Step 6

Rerun Magento's  
Compilation Process

# Step 6 - Rerun Magento's Compilation Process

If you are 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**).



## Command Line Equivalent

```
php -f shell/compiler.php -- compile
```

The screenshot shows the Magento Admin Panel interface. At the top, there is a navigation bar with the Magento logo and "Admin Panel" text, a search bar labeled "Global Record Search", and user information: "Logged in as kristof | Sunday, July 21, 2013 | [Try Magento Go for Free](#) | [Log Out](#)". Below this is a secondary navigation bar with tabs for Dashboard, Sales, Catalog, Mobile, Customers, Promotions, Newsletter, CMS, Reports, and System (which is highlighted in orange). A "Get help for this page" link is also present. A message banner below the navigation bar reads: "Latest Message: Get More eCommerce Power with Magento Enterprise [Read details](#)" and "You have 4 notice unread message(s). [Go to messages inbox](#)". The main content area is titled "Compilation" and features a "Run Compilation Process" button in an orange box. Below the button is a table showing the current compilation state.

Compilation State	
Compiler Status	Disabled
Compilation State	Not Compiled

# Step 6 - Optional



## Extension Conflicts

You can only replace each of the core classes of Magento's architecture once with an extension. Therefore, when installing more than one extension, you create a potential conflict. The potential for conflict is higher when two or more extensions replace similar Magento behaviour functions - only one extension can be the 'winner' and the other extension won't function as advertised. For further reading, check out this [Magebase article](#).



## Check for Conflicts with Existing Extensions in your Store

Check for potential extension conflicts by downloading [this file](#) from the firegento project. Read more about the Firegento project [here](#) (in German). Save the file as firegento.php and upload it via FTP or SFTP to your Magento server. Opening the file in your browser as <http://www.example.com/firegento.php> will advise you of any potential conflicts between extensions you have installed in your store.

If a conflict is identified, uninstall one of the competing extensions and contact the developers of both conflicting extensions to determine whether a workaround is available (note however that this is not always possible).

# Troubleshooting

# Troubleshooting

## I can't find the extension after installing it

This problem occurs when Magento can't read the file `app/etc/modules/NAME.xml` (where NAME should relate to the module you just installed or the developer who created the module).

If the newly installed extension is not listed under

**System > Configuration > Advanced > Disable Modules Output**, try the following:

- Ensure the uploaded files' permissions can be read by Magento
- Flush Magento's cache and/or manually delete the contents of `var/cache`. In rare cases if `var/cache` is not writable by Magento, it will cache files in your system's `tmp/` folder

## New configuration options do not appear after refreshing the cache

If new configuration options do not appear in the backend of your store after refreshing the cache several times, try the following:

- Ensure the uploaded files' permissions can be read by Magento
- Flush Magento's cache and/or manually delete the contents of `var/cache`. In rare cases if `var/cache` is not writable by Magento, it will cache files in your system's `tmp/` folder

# Troubleshooting

## There was an error processing..." page

The error page contains a reference number at the bottom. The reference number corresponds to a file placed on your server in <MAGENTO\_ROOT\_FOLDER>/var/report/. Opening this file will contain more information about the reason for the error.

If the error report mentions files from includes/src folder, it's likely that Magento's compilation mode was not disabled ([Step 1](#) of this guide). If this is the case, you won't have access to the Magento backend. To recover from this, disable compilation mode by editing includes/config.php. Change the content to:

```
define ( 'COMPILER_INCLUDE_PATH' , dirname ( __FILE__ ) . DIRECTORY_SEPARATOR . 'src' );
```

to

```
#define ( 'COMPILER_INCLUDE_PATH' , dirname ( __FILE__ ) . DIRECTORY_SEPARATOR . 'src' );
```

If this was the cause for the error, your Magento backend will now be functional again. You now need to rerun the compilation process (see [Step 6](#) of this guide).

If you need further assistance, contact the extension developer with the contents of this error log.

# Troubleshooting

## A white/blank page at any stage after the installation

A white/blank page indicates that a fatal error occurred during the page load. More information on the error can be found in your server's error logs. Please note that the server error log is different to Magento's logs – your system administrator or webhosting company can help you in locating this log.

If you need further assistance, contact the extension developer with the contents of this error log.

## I need to disable an extension

You can completely disable an extension by removing its xml file in the app/etc/modules folder. The naming convention for this file is <COMPANY\_NAME>\_<EXTENSION\_NAME>.xml

Change the filename ending to anything but xml. So for example to disable Fooman Speedster you could rename the file app/etc/modules/Fooman\_Speedster.xml to app/etc/modules/Fooman\_Speedster.off

Renaming this file instead of deleting it allows you to reenabte the extension again.

After renaming the file you need to refresh your Magento cache by deleting the contents of the var/cache folder.

Please note that disabling an extension under **System > Configuration > Advanced > Disable Modules Output** only disables block output, the rest of the extension's functionality remains enabled.

**Can this guide be improved? Do you have additional troubleshooting suggestions?**

Please let me know at [kristof@fooman.co.nz](mailto:kristof@fooman.co.nz)!