How to Package Magento Connect Extensions

Table of Contents

Preface

About This Document

This document provides information on how to package extensions using Magento Connect Manager and covers details for both pre-1.5 and 1.5 and later versions of Magento Community Edition, 1.9.0.0-RC2 and later versions of Magento Enterprise Edition.

Updates, Comments, and Suggestions

Please forward any comments, suggestions, or questions about the information contained in this document to connect@magento.com.

Disclaimer

Magento reserves the right to change at any time and without prior notification, business models or processes relevant to packaging, selling, marketing, developing extensions and applications for Magento Connect.

Magento reserves the right to remove any extensions (free or paid) that do not conform to the guidelines or exploit capabilities to promote other products.

Section 1: Introduction

What is Magento Connect?

Magento Connect is a marketplace that allows Magento community members and partners to share their open source or commercial contributions for Magento with the community. Typical contributions include:

- Development modules
- Language packs
- Design interfaces
- Themes

Community Extensions vs. Commercial Extensions

Community Extensions: Open source extensions created by Magento Community members or partners that are released under an open source license.

Commercial Extensions: Contributions available for purchase by Magento community members or partners. Commercial extensions may be sold, distributed, and licensed directly through the seller.

Version Compatibility

With two separate platforms and a spectrum of platform versions, it is important to understand the compatibility specifications for Magento extensions. Below you will find a table with Magento product versions and their respective Magento Connect Channel.

Magento Connect Channel 1.0	Magento Connect Channel 2.0
• Community Edition 1.4.x and older **NOTE:** Extensions for PEAR (1.0 Packages) cannot be larger than 2MB.	• Community Edition 1.4.2.0 beta only • Community Edition 1.5 + • Enterprise Edition 1.9.0.0-RC2 +

Section 2: How to Package & Publish Your Extension

You just built a Magento extension, now what? Publishing your extension is a two-step process:

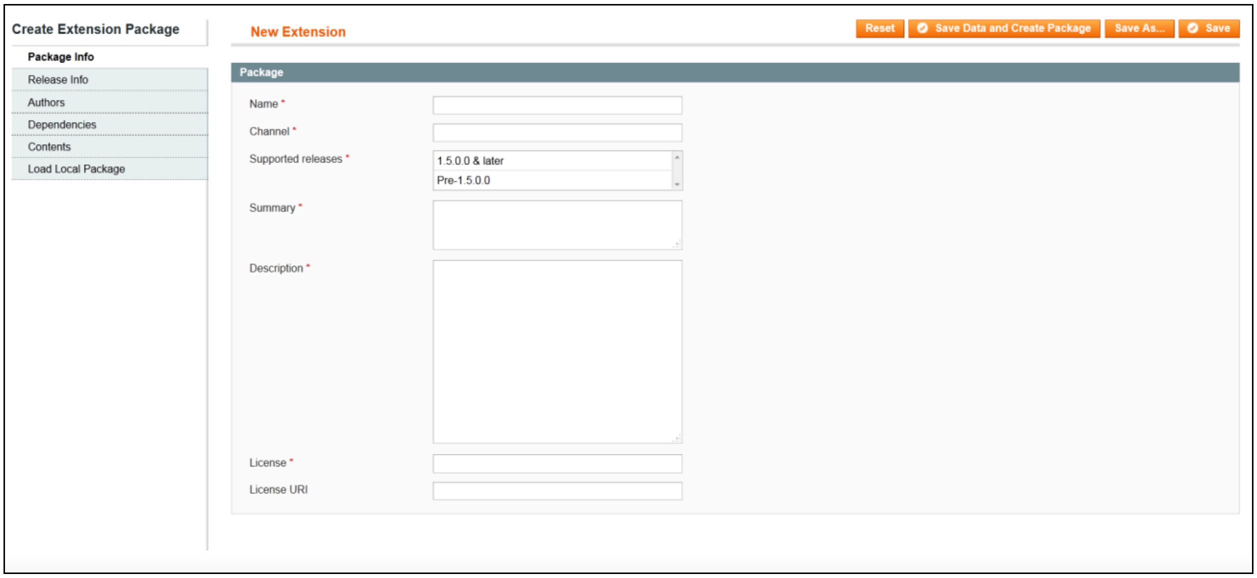
1. Package your extension
2. Create extension profile

Step One: Package Your Extension

The first step in publishing your extension is to package and upload it to the Magento Connect marketplace. You can do this from your own Magento instance – directly from the Admin panel.

To package an extension:

1. Log in to Magento Admin panel
2. From the Admin panel, select **System > Magento Connect > Package Extensions**. This takes you to the **Create Extension Package** page similar to the image below:



3. The **Create Extension Package** consists of six sections:
    A. Package Info
    B. Release Info
    C. Authors

D. Dependencies

E. Contents

F. Load Local Package

Follow the steps below to fill in the correct information for each section:

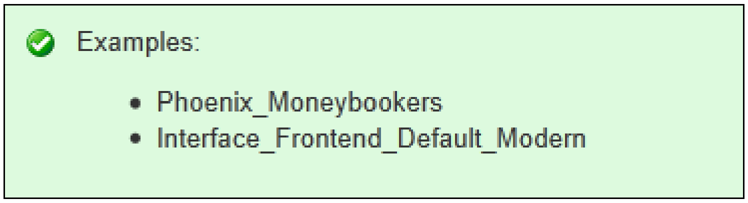
Section A: Package Info

Fill in the following fields:

Name

Name field is required. It is good practice to keep the name short using simple words whenever possible.

Formatting requirements: The name of an extension can be a combination of letters (a-z or A-Z), numbers (0-9), and the underscore character (_). Other characters are not allowed.



✓ Examples:

- Phoenix_Moneybookers
- Interface_Frontend_Default_Modern

Example of Extensions Names

NOTE: The name used in this field will be your extension key. It is case-sensitive.

Channel

Channel field is required and determined by the version of Magento you are using to package your extension. Make sure to type it exactly as it is shown below. These fields are case sensitive.

If you are using **Magento 1.5 or later to package your extension** type**:** community

If you are using **Magento 1.4 or prior to package your extension** type: connect.magentocommerce.com/community

NOTE: If the wrong channel is used during the packaging process, you will receive an error and will have to resubmit it.

Supported Releases

This field is intended for backward compatibility; see the section for [Version Compatibility]

- Select the option Pre 1.5 if this extension supports Magento versions 1.4.x and lower. For these releases, the package will be created in PEAR format and all associated files will be saved in the magento/var/pear/ directory.
- Select the option 1.5.0.0 & later if this extension supports Magento versions 1.5.x and later. For these releases, the package will be saved in the magento/var/connect directory.
- Select both options if this extension supports both older and newer versions.

NOTE: If you are using EE 1.9 or earlier versions this field is absent but this won't affect extension packaging process. You can skip this step in such situation.

Upload and Packaging Options

If you are packaging an extension using a **Magento Pre 1.5 (i.e. 1.4.x and lower**) version and the extension only supports Pre 1.5 versions of Magento use the following settings:

- Channel: connect.magentocommerce.com/community
- Supported Releases: Pre 1.5
- Package file will appear in <magento root>/var/pear/. Package will appear in PEAR format.

NOTE: If there is no <magento root>/var/pear folder then please create it before packaging your extension.

If you are packaging an extension using a **Magento 1.5 and later** version and the extension only supports 1.5 and later versions of Magento use the following settings:

- Channel: Community
- Supported releases: 1.5 & later
- Package file will appear in <magento root>/var/connect/.

If you are packaging an extension using a **Magento 1.5 and later** version, but the extension supports both Magento Pre 1.5 and Magento 1.5 and later versions use the following settings:

- Channel: connect.magentocommerce.com/community
- Supported Releases: Pre 1.5 and 1.5 & later (select both)
- Package will appear in <magento root>/var/pear.

NOTE: Another package will be created in <magento root>/var/connect. You will need this file if your extension is paid and you prefer to give customers the .tgz file to upload into their Magento Connect Manager.

If you are supporting two extensions with the same name when one of them is compatible with Magento Pre 1.5 and another with Magento 1.5+ do the following:

- Package version of your extension for Magento Pre 1.5 in Channel 1.0 format and upload it to Magento Connect.
- Package new version of your extension for Magento versions 1.5+ in Channel 2.0 format and upload to Magento Connect

NOTE: Version numbers for your Pre 1.5 and 1.5+ versions should be different. You are not allowed to upload several versions with the same version number to Magento Connect.

Tip: It's a good idea to also mention information about the supported releases in the extensions description, so merchants can quickly know whether or not they can use it.

Summary and Description

Provide a brief summary and description about the extension. This is a required field. Magento uses this information in the extension review process. The summary will be displayed in the Magento Connect Manager Admin.

License

License field is required. Select the license that your extension falls under. The full list of possible licenses as stated on the Magento community website is:

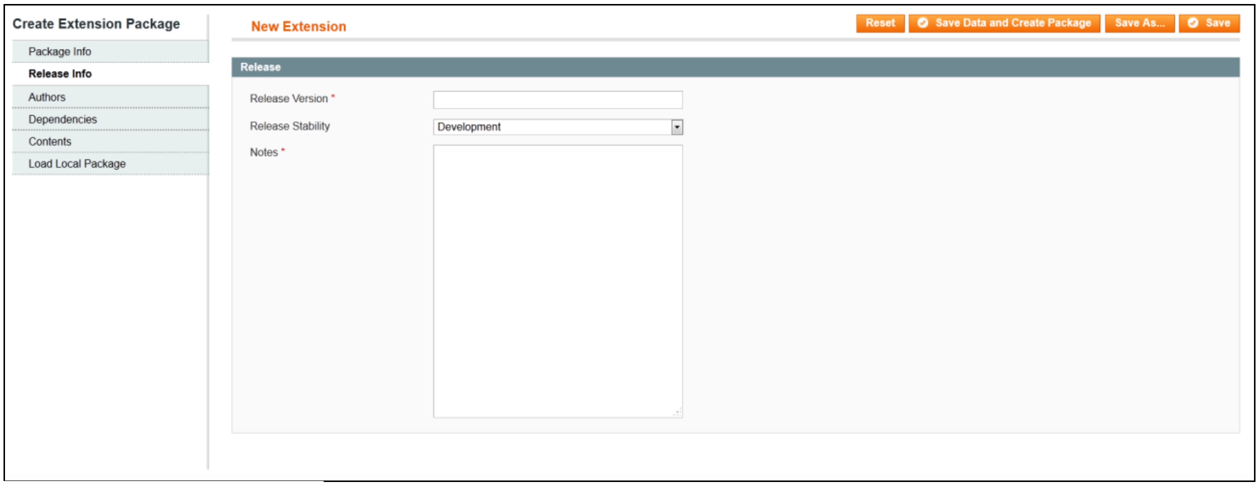
- Open Software License (OSL)
- Mozilla Public License (MPL)
- Massachusetts Institute of Technology License (MITL)
- GNU Lesser General Public License (LGPL)
- GNU General Public License (GPL)
- Berkeley Software Distribution License (BSDL)
- Apache Software License (ASL)
- Academic Free License (AFL)
- You can specify another (commercial) license if you distribute the extension package as paid.

License URL

Use this field to link to the license text. It is not required, but recommended, for example http://opensource.org/licenses/osl-3.0.php

Section B: Release Info

In order to understand this process, please see the following [Wikipedia link on Software Versioning](#).



Release Version *

Release Stability: Development

Notes *

Release Info Screen Shot

Release Version

The release version can be any arbitrary number value that you want to use as the base version. This is often set to 1.0.0 or 10.0.0. It is critical, however, to increment the release version number every time you update and repackage this extension. To protect against incorrect release version inputs, Magento Connect will not allow you to upload the same version twice.

Release Stability

Release stability can be set to:

- Alpha
- Beta
- Stable

By default, Magento Connect users are only allowed to install **Stable** release extensions. To download development, alpha, or beta releases, the user must change this setting in the backend of Magento Connect. If you upload a package with release stability alpha or beta, add this information into the extension description.

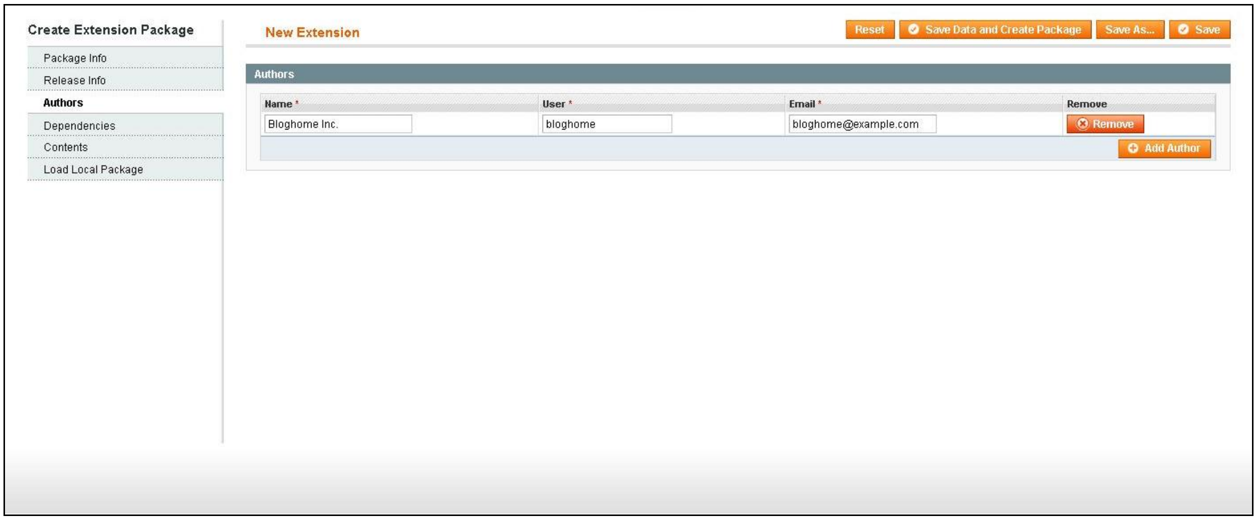
NOTE: Do not attempt to upload an extension with release stability **Development**. This option is used for testing purposes only. No **Development** release extensions will be approved.

Notes

Add any notes you may have for this release.

Section C: Authors

In the Authors section, type in the real name, user name, and email address of all members who contributed to this extension. The first user must match the user name of your magentocommerce.com account used to display your extensions on Magento Connect.



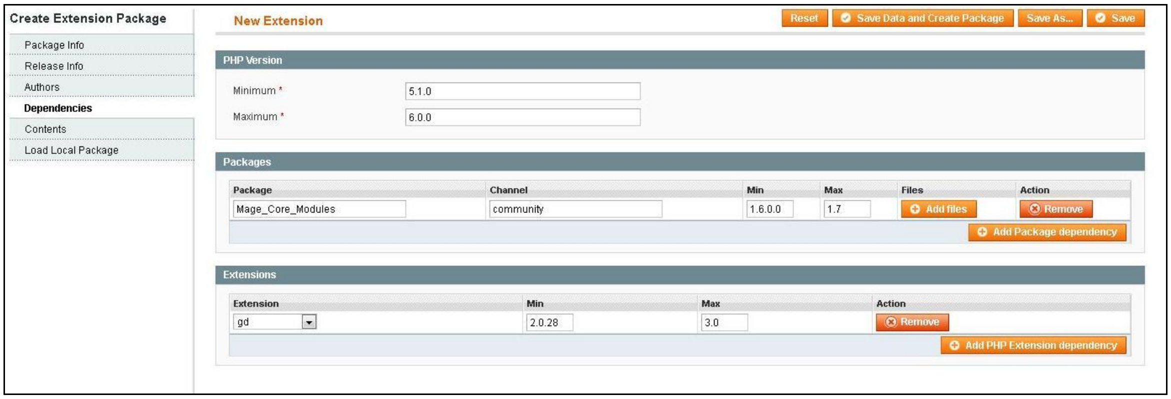
Authors

Name *	User *	Email *	Remove
Bloghome Inc.	bloghome	bloghome@example.com	Remove

Section D: Dependencies

The dependencies tab is used to specify the dependence of our code in regard to the PHP version. If it is necessary, the dependence on other extension packages can be filled out as well. During the installation of an extension, the system will check it in regard to all versions specified for it, and if dependent extensions are not installed, Magento Connect Manager (MCM) offers to install them. Also, you can specify the dependence on the PHP extensions, for example, if your module uses GD library functions of the latest version. In this case, if a necessary PHP extension is not installed on the server, MCM will show an alert the administrator.

The following screen shot provides an example of dependency on the Mage_Core_Modules package. This means that our extension will only work with Magento versions within the set range. The PHP version_compare function http://php.net/version-compare is used to verify the compatibility of versions during the extension installation.



PHP Version

Minimum * 5.1.0

Maximum * 6.0.0

Packages

Package	Channel	Min	Max	Files	Action
Mage_Core_Modules	community	1.6.0.0	1.7	Add files	Remove

Extensions

Extension	Min	Max	Action
gd	2.0.28	3.0	Remove

Dependencies Screen Shot

Section E: Contents

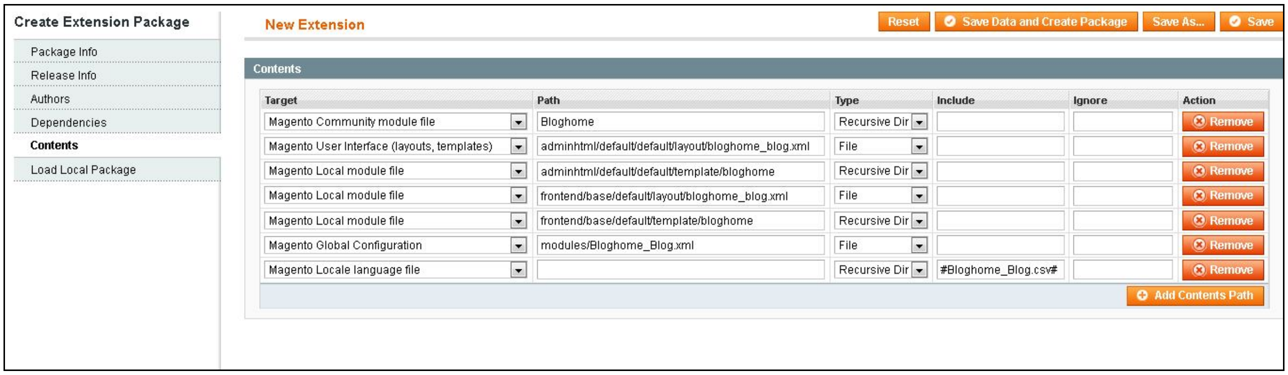
This is one of the most important sections in the extension packaging process. It is also the easiest to mess up. Be careful here.

For proper compilation of extension package contents, use the following tips for the Target field. The following options correspond to the path on the disk:

- Magento Local module file - ./app/code/local
- Magento Community module file - ./app/code/community
- Magento Core team module file - ./app/code/core
- Magento User Interface (layouts, templates) - ./app/design
- Magento Global Configuration - ./app/etc
- Magento PHP Library file - ./lib
- Magento Locale language file - ./app/locale
- Magento Media library - ./media
- Magento Theme Skin (Images, CSS, JS) - ./skin
- Magento Other web accessible file - ./
- Magento PHPUnit test - ./tests
- Magento other - ./

Where "./" represents the Magento root directory.

In the **Path** field, enter the path relative to the path in the **Target** field. In the **Include** and **Ignore** fields, enter a regular expression which will check the path with the file name only for the rows where the **Type** field is set to Recursive Dir. A file to be included in the extension package must match the expression specified in the Include field. If the file matches the expression in the Ignore field, it will not be included in the extension package correspondingly.



Target	Path	Type	Include	Ignore	Action
Magento Community module file	Bloghome	Recursive Dir			Remove
Magento User Interface (layouts, templates)	adminhtml/default/default/layout/bloghome_blog.xml	File			Remove
Magento Local module file	adminhtml/default/default/template/bloghome	Recursive Dir			Remove
Magento Local module file	frontend/base/default/layout/bloghome_blog.xml	File			Remove
Magento Local module file	frontend/base/default/template/bloghome	Recursive Dir			Remove
Magento Global Configuration	modules/Bloghome_Blog.xml	File			Remove
Magento Locale language file		Recursive Dir	#Bloghome_Blog.csv#		Remove

Contents Screen Shot

Section F: Load Local Package

Once you've filled in all of the fields, click **Save Data and Create the Package.**

This will save it in a package data file and create a package file.

What is the package data file?

The package data file is an XML document that stores the rules for building the package. This file is saved on the disk in the magento/var/connect/ folder if packaged with Magento 1.5 and later or in the magento/var/pear directory if packaged using Magento 1.4 or prior. A different location can be specified if you click the **Save As** button.

What is the extension package file?

The Extension Package file contains all source code needed. In addition, the archive includes another file that is created by a packager - package.xml. This file contains information about the extension, including the description of the structure of files and folders included in the package, and md5 sum for each file.

Now that the extension package file is ready, it can be uploaded. To do this, you'll need to create an extension profile in Magento Connect.

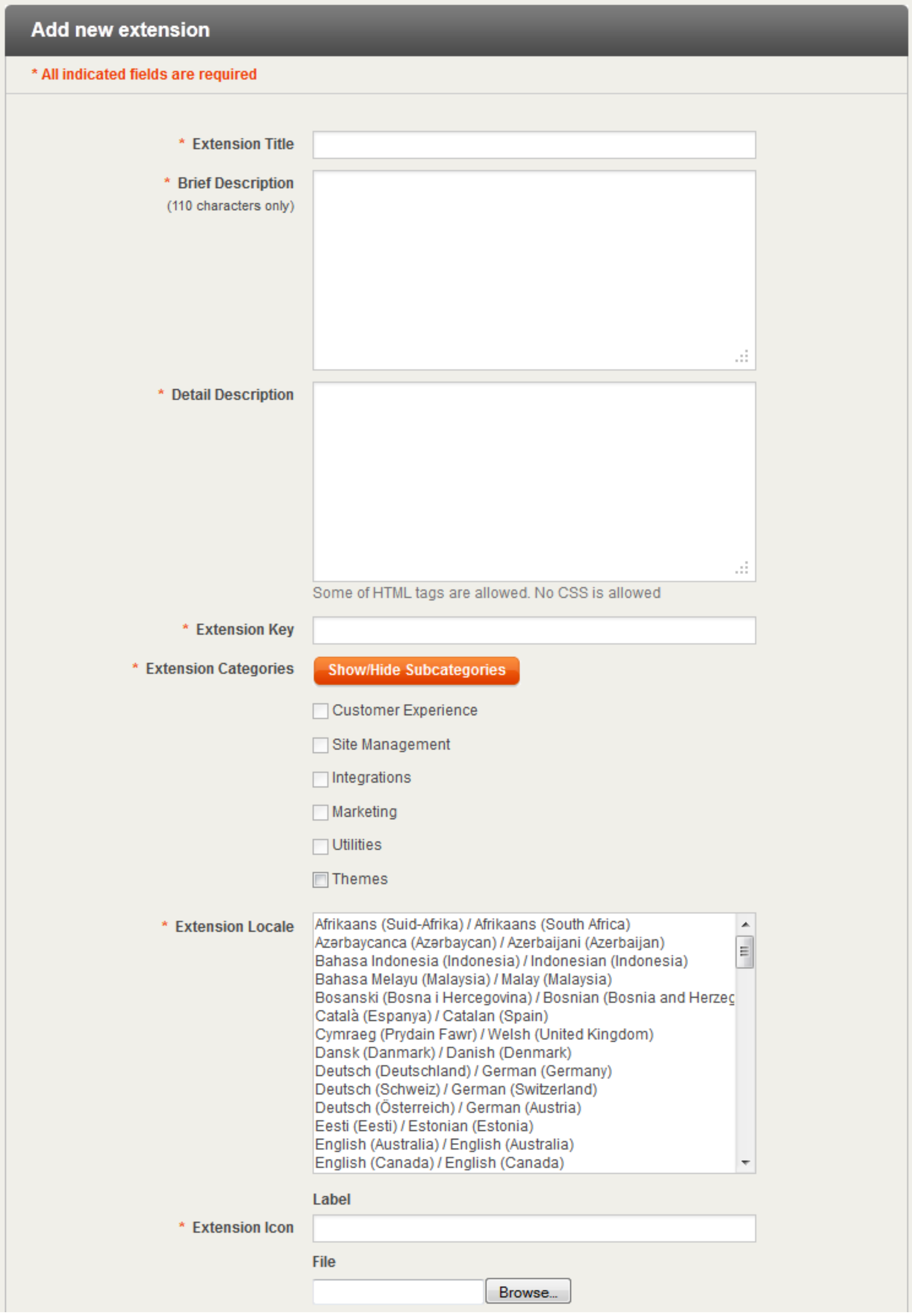
NOTE: First you should set write permissions to the appropriate folders

Step Two: Create an Extension Profile

Creating an extension profile is the critical final step in publishing a Magento extension. The extension profile page describes your extension, options to download if it is a free extension, or a link to purchase if it's a paid extension. A good extension profile should also include other important information that users may need to be aware of to configure the extension, as well as contact information for support questions. Developers must also follow the design guidelines or risk the extension not being approved. You can download the design guidelines here.

To create an extension profile page:

1. Log in to **magentocommerce.com.**
2. From the top right of the page, click on **My Account** to access your account dashboard.
3. From the left side panel, select **Developers** to expand its options.
4. From the expanded **Developers** panel, click **Add New Extension**.



Add new extension

* All indicated fields are required

* Extension Title

* Brief Description
(110 characters only)

* Detail Description

Some of HTML tags are allowed. No CSS is allowed

* Extension Key

* Extension Categories

□ Customer Experience

□ Site Management

□ Integrations

□ Marketing

□ Utilities

□ Themes

* Extension Locale

Afrikaans (Suid-Afrika) / Afrikaans (South Africa)
Azərbaycanca (Azərbaycan) / Azerbaijani (Azerbaijan)
Bahasa Indonesia (Indonesia) / Indonesian (Indonesia)
Bahasa Melayu (Malaysia) / Malay (Malaysia)
Bosanski (Bosna i Hercegovina) / Bosnian (Bosnia and Herzeg
Català (Espanya) / Catalan (Spain)
Cymraeg (Prydain Fawr) / Welsh (United Kingdom)
Dansk (Danmark) / Danish (Denmark)
Deutsch (Deutschland) / German (Germany)
Deutsch (Schweiz) / German (Switzerland)
Deutsch (Österreich) / German (Austria)
Eesti (Eesti) / Estonian (Estonia)
English (Australia) / English (Australia)
English (Canada) / English (Canada)

Label

* Extension Icon

File

Add New Extension

5. From the new extension profile page, fill in the following:

   a. **Extension Title:** Short descriptive title that explains your extension. The title should be limited to 45 characters to be properly displayed on Magento Connect.

   NOTE: The extension name will create the URL for your extension profile. This URL cannot be changed. You would need to create a new extension profile to correct any name errors.

   b. **Brief Description:** Briefly describe the extension's functionality. This field is limited to 110 characters, plain text only. This information is displayed in Magento Connect search results and in search engine results.

   c. **Detail Description:** Accurately describe functionality as well as any other information that a merchant might need to be aware of. This is the best place to market your extension. Include information like:

      - Stability
      - Supported releases
      - Support information
      - Where to locate and configure the extension in the admin panel.

   Please refer to the design guidelines for a list of supported HTML tags.

   Extension Key: The extension key must be the exact same name you used during the packaging process. It is case-sensitive. The name of an extension can be a combination of letters (a-z or A-Z), numbers (0-9), and the underscore character (_). Other characters are not allowed. You cannot change the file name of your package to match the key. The name used during the packaging process is stored within the package.

   Once you save the profile, this field cannot be changed. You will need to create a brand new profile if the key does not match the package name exactly.

   If your extension is paid, please include an extension key as well. This key will be used if you choose to make your extension free in the future.

   NOTE: You can now switch your extension from paid to free and vice-versa. However, do not change the extension key once the extension is published on Magento Connect as changing the key can cause issues for merchants that have already downloaded the extension. If you are interested in changing the status of an extension please contact our team for recommendations on best practices.

d. **Extension Categories:** Select up to three subcategories for your Magento Connect Extension.

   Subcategories classify your extension so that merchants can more easily find it in Magento Connect. Your extension will be displayed under these subcategories and their respective main categories on Magento Connect.

   Note: Select Show/Hide subcategories to expand subcategory menu.

e. **Extension Locale:** Select the languages supported in the extension package.

f. **Extension Icon:** This icon will be your extension's default image and should adhere to our design guidelines. More screenshots maybe be added once you have saved your extension profile.

   Note: Screenshots are required. Be sure to reference to the design guidelines for acceptable screenshot use.

g. **Magento Edition Information:** If this will be a free extension, checkmark the box next to **Is Free**. Then, fill in the following fields:

   Note: A version must first be selected in order to complete the rest of the fields.

h. **License Type:** In this field choose the type of license you will be using for your extension. License type is a required field. This should match the License Type entered during the packaging process.

   You can choose from one of the available options below or choose Custom License:

   - Open Software License (OSL)
   - Mozilla Public License (MPL)
   - MIT License (MITL)
   - GNU Lesser General Public License (LGPL)
   - GNU General Public License (GPL)
   - BSD License (BSDL)
   - Apache Software License (ASL)
   - Academic Free License (AFL)

i. **License Name:** Choosing a License Type will automatically fill in this field. If you choose Custom License, type in the name of the license.

j. **License URL:** Choosing a License Type will automatically fill in this field. If you choose Custom License, type in the license URL.

k. **Versions:** Check all the versions which this extension supports.

   Note: It is critical that the versions selected here reflect the **Supported releases** you selected when packaging this extension. If there is a conflict between the versions selected here and the supported releases for the package, your extension will not work in Magento Connect.

6. If this is a paid extension, fill in these additional fields:

   a. **Landing Page:** This link must be a direct link to a web page off of your own website which may provide additional details on the extension and has options to purchase, download and your contact information. Your company homepage or other products not related to this specific extension cannot serve as the landing page.

   b. **Price:** Specify the price of your extension.

   c. **Currency:** Available options are US Dollar or Euro.

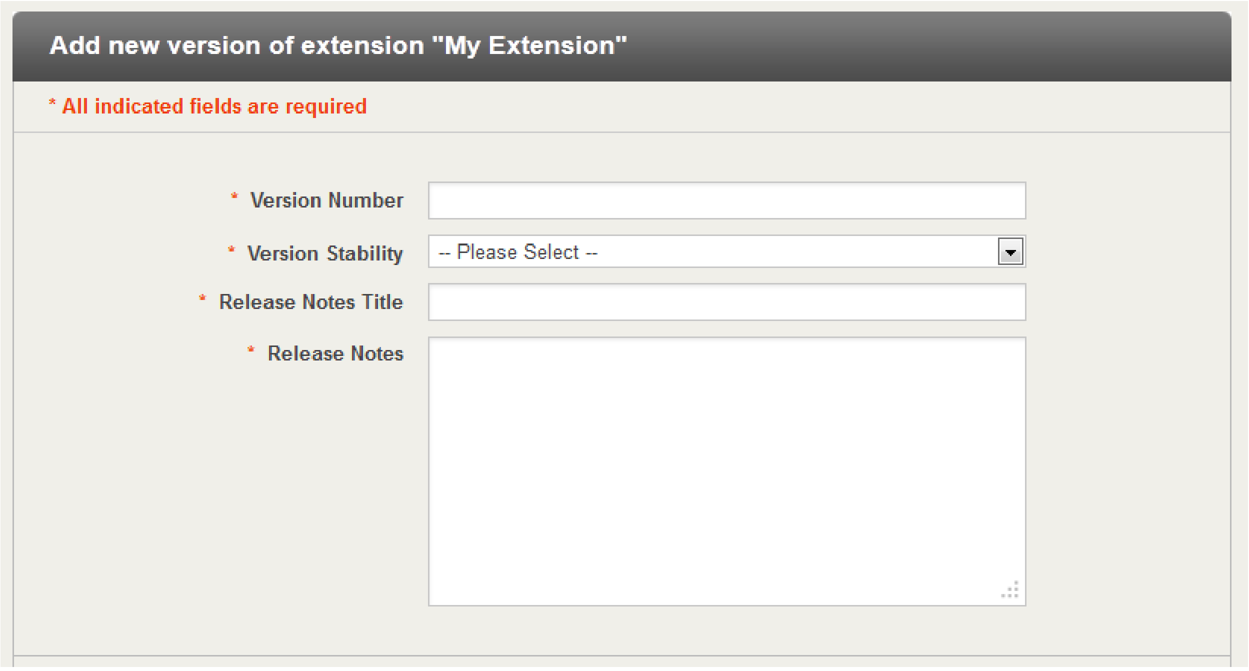
7. Click **Save** to create this extension profile. If saved properly, you will be re-directed to your extensions list under “My Extensions”.

8. **Screenshots:** Once you have created your extension profile with the required information above, you can add screenshots to better explain your extension by clicking the “Edit” link from your “My Extensions” list and navigating to the Screenshots Tab. Here you will find your Extension Icon as the first image. This image is set as your “base image.” You must add at least one screenshot to your extension profile. Be sure to refer to the design guidelines for acceptable screenshot use. Click **Save** when you have finished uploading your screenshots.

Section 3: Uploading Your Free Extension Package

Once you've created your extension profile page with the information required in the section above, you will need to upload your extension package from the Versions Tab:

NOTE: Remember – Your extension must be packaged under the username of the developer uploading the extension. Packaging in any other name will result in an error.



Add new version of extension "My Extension"

* All indicated fields are required

* Version Number

* Version Stability -- Please Select --

* Release Notes Title

* Release Notes

Version Information

Fill in all required fields with the same information used during the extension packaging process. Do not change the name of the package file when you upload it.

- If your extension supports Pre 1.5 versions of Magento upload the package from: <magento root>/var/pear/.
- If your extensions supports1.5 and later versions of Magento upload the package from: <magento root>/var/connect/.
- If your extensions supports both Pre 1.5 versions and 1.5 and later versions of Magento upload the package from: <magento root>/var/pear/. Since both Pre 1.5 and Post 1.5 versions are selected in the compatibility section of the release notes Magento Connect will create and upload the package to the second channel for you.

NOTE: Please be aware, if there is an error in your extension package that prevents a successful upload, you must repackage the extension (using a newer and higher version number) in order to attempt again.

Common Upload Errors

The most common upload errors consist of these mistakes that can easily be avoided during the extension creation process:

1. Remember that the name is case sensitive. Your package must be identical to the name listed on the extension profile page.
2. Channel is case sensitive. Always use lower case **community**
3. The lead author listed when packaging the extension must be the user who is uploading the extension or the upload will result in an error.
4. Package using **“User Name”** not “Screen Name” or the upload will result in an error.

   See the Magento Connect FAQ for more information.

Section 4: Submitting Your Extension

In order to submit your extension for review, you must click **Submit for Approval** from within the Edit Extension Tab of your profile or the My Extensions List. Your extension will be in a **Not Submitted** status and will not be reviewed until you submit your extension.

Extension Status

Once your extension is submitted successfully, you can view your extension status by logging back into the My Account section on magentocommerce.com. The status of your extension will be displayed on the Manage Extensions list.

Extension Statuses

Submitted – This means your extension has been successfully submitted for review.

Not Approved – If you have submitted your extension and it is listed as Not Approved this means that there was a problem with your extension. You will also receive an email to the email address on file explaining why your extension was not approved. The majority of extensions which are listed as Not Approved happen due to the extension not being packaged correctly or meeting the general design guidelines. You can download the guidelines here.

Live – Once we approve your extension it is listed as Live and is immediately available on Magento Connect.

Offline – You can take your live extension offline at any time from within the Edit Extension Tab of your profile.