# Magento for Designers: Multi-Store Functionality

Magento is a stunningly powerful e-commerce platform. In this miniseries, we’ll learn how to get started with the platform, get to know the terminologies, set up a store and all related aspects of it and finally learn how to customize it to make it our very own.

In this seventh part, we’ll be learning how to set up the fabled Magento multi store functionality in simple, easy to follow steps. Excited? Let’s get started!

## The Full Series

* [Part 1: Installation and Setup](http://net.tutsplus.com/tutorials/php/magento-for-designers-part-i/)
* [Part 2: Products, Taxes, Categories, Payment Gateways, etc.](http://net.tutsplus.com/tutorials/php/magento-for-designers-part-2/)
* [Part 3: Theming](http://net.tutsplus.com/tutorials/php/magento-for-designers-part-3/)
* [Part 4: Building the Theme](http://net.tutsplus.com/tutorials/php/magento-for-designers-part-4/)
* [Part 5: Building the Product Information Page](http://net.tutsplus.com/tutorials/php/magento-for-designers-part-5/)
* [Part 6: Building the Home Page, with a Content Slider](http://net.tutsplus.com/tutorials/php/magento-for-designers-part-6/)
* [Part 7: The Fabled Multi-Store Functionality](http://net.tutsplus.com/tutorials/php/magento-for-designers-multi-store-functionality/)

## A Quick Recap

In the last few parts, we took a high level look at how Magento themes are put together, the components that drive it and how everything works together to create a working theme. We also looked at a number of API methods that Magento exposes so that we can painlessly acquire information about the product.

## What Are We Learning Today?

Today, we’ll be taking a focused, step by step look at what Magento multi store is, how it can help you, and finally how to set it all up. I know it sounds interesting and you’re itching to get started, so let’s go!

## What’s Magento Multi Store?

Magento ships with a number of very powerful features under the hood that go mostly unnoticed by the folks that install and use it on a daily basis. The feature we’re looking at today, the multi store functionality, happens to be one of those; and arguably, it’s one of the most advanced ones. There isn’t a general vernacular for this feature so I’ll be just calling it multi store from now on.

Using this multi store functionality, we can use a single Magento installation and use it to power any number [within technical limits, of course] of stores or websites. This is definitely a godsend for stores that would like to sell products on different domains but would like to have a unified administrative centre. If you need even more granular control, you can set up different, separate stores under the same domain to differentiate between items even better.

## The Terminologies Involved

[Magento’s documentation](http://www.magentocommerce.com/knowledge-base/entry/overview-how-multiple-websites-stores-work/) regarding this topic has a pretty decent explanation of the terminologies so I’ll make a concise introduction here:

**Website:** Container for stores and their related views

**Store:** Exactly what you’d expect but in the abstract sense. You can control your catalog through your store but nothing will be rendered on screen

**Store View:** These enable the actual rendering of your store. If you want to have a multi-lingual store, you’d be modifying the store view, not the store itself.

All these terminologies will start making more sense once you see them in action. For now, we’ll move on.

## Today’s Goal

We’re just going to implement the system I talked about as an example above. We’re going to add a store front to a domain without installing Magento on it. Instead it’ll be using an already installed version of Magento to do all the heavy lifting. And that’s about it! I’m choosing to focus specifically on this scenario since I’ve been getting numerous messages through various channels asking how to set it up. So let’s get started!

## Requirements

There are only 2 requirements/restrictions here.

* **Computing Power**

Choose your servers carefully. Running a single store on an installation often chokes most shared hosts and brings it down to its knees. Imagine running multiple store on one. If you decide to run everything off a single installation, please make sure your server will be able to handle it. If the total number of products is small, say less than 100, it shouldn’t be a concern but once you get past 500 or so, it will really begin to show.

* **Version of Magento You’re Running**

This method will require the 1.4.x.x branch or higher. Sure, you can get the functionality to work with previous versions but it is a convoluted, error prone, lengthy affair where you had to modify the core thus introducing even more work when you’ll need to upgrade in the future. Each new version is faster and brings more features to the table which means I’ll be focussing exclusively on the newer branch. If you’re still running an old branch, back everything up and upgrade. Please.

## Step 1: Initial Preparations

First up, if you’re on your local server, you probably entered nothing for the URL during the installation and thus Magento has the default value stored. We’ll need to change to point it to a concrete location. We’ll do it quickly right now.

http://net.tutsplus.com/wp-content/themes/tuts/images/global/box.gif

Go to System -> Configuration and under the Web category, change the base URL value to reflect your local set up. I have mine working under a predictably named Magento directory thus my URL. Please remember to add the trailing slash; otherwise, Magento will spaz out.

With that out of the way, we can get to the next step: creating a root category for the new store. You may ask why. My reasoning is that with the additional stores, the number of prospective categories will increase as well. With organizational structure in mind, it makes sense to keep each store’s categories in a separate root category. We’ll be creating one now.

http://net.tutsplus.com/wp-content/themes/tuts/images/global/box.gif

Just give a name to the new category and make sure it’s set to active.

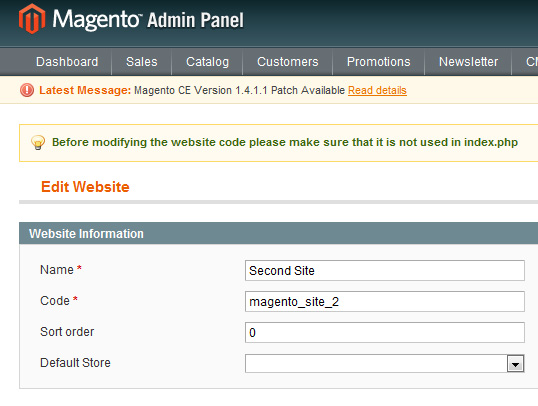
http://net.tutsplus.com/wp-content/themes/tuts/images/global/box.gif

Finally, set the is anchor value to yes as well. This is important.

## Step 2: Setting up the New Site

First, we’ll set up the new site in Magento’s back end.

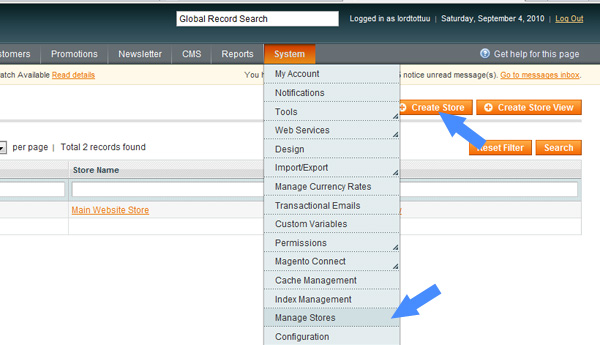
http://net.tutsplus.com/wp-content/themes/tuts/images/global/box.gif

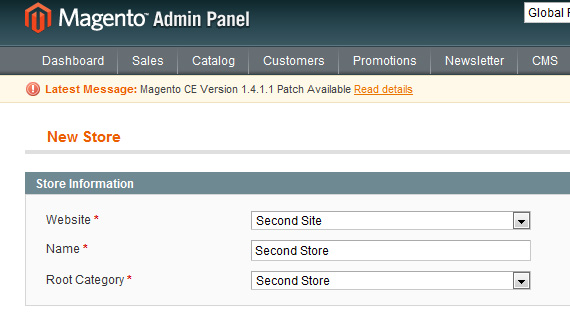


In the resulting screen, key in a name for the site as well as a code for it. Neither is really important, other than that both need to be unique.

## Step 3: Setting up the New Store

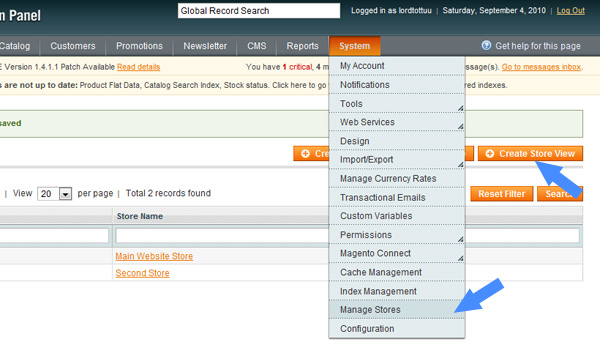
Now we’ll move on creating a new store.

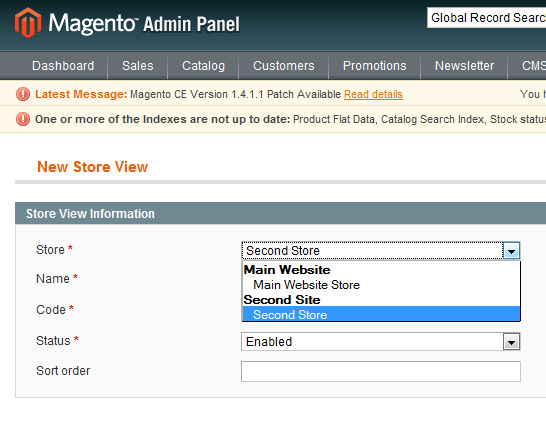


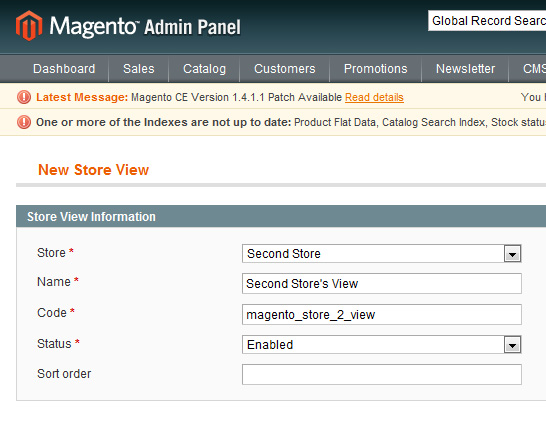


This should be self-explanatory. We set the website and category to the ones we created a few steps back. The name of the store is just for human readability, so feel free to name it as you like. I know having the same name for the store and category is a little confusing here. Feel free to name it as you like in your installations. There’s no concrete naming scheme here for you to follow.

## Step 4: Setting up the New Store View







Again, these steps should be fairly self-explanatory. We select the appropriate website and store for the view along with keying in a name and code for it. Additionally, remember to set its state to enabled. It may seem obvious, but we tend to forget it, so I thought a quick “heads up” was in order.

With these, most of the back end work in Magento comes to an end. We’ll need to revisit this later so for now we’ll focus on preparing the new domain.

## Step 5: Prepping the New Domain

Ok, to be honest, this is the easiest step there is. Just FTP into the server with the working Magento installation and copy the index.php file as well as the htaccess file over to the new domain.

http://net.tutsplus.com/wp-content/themes/tuts/images/global/box.gif

Open up index.php and look for the following code at around line 45,

[view plaincopy to clipboardprint?](http://net.tutsplus.com/tutorials/php/magento-for-designers-multi-store-functionality/)

1. $mageFilename = '$mageFilename = 'app/Mage.php';';

Change it to the following.

[view plaincopy to clipboardprint?](http://net.tutsplus.com/tutorials/php/magento-for-designers-multi-store-functionality/)

1. $mageFilename = '../magento/app/Mage.php';

Remember to point it to the Mage.php file of the working installation. Both my sites run under sub folders in my server so I just ask it to go a directory up, go into the magento folder, into the app folder and then access the required file. Depending on your server set up, this may vary, for example, your main installation may be in your account root while the other site may be under a parked domain. In that case, the following code will do.

[view plaincopy to clipboardprint?](http://net.tutsplus.com/tutorials/php/magento-for-designers-multi-store-functionality/)

1. $mageFilename = '../app/Mage.php';

As I said, just make sure you point Magento to the right location

## Step 6: Making it all Work – the HTACCESS file

Now open up the copied htaccess file and append the following to it.

[view plaincopy to clipboardprint?](http://net.tutsplus.com/tutorials/php/magento-for-designers-multi-store-functionality/)

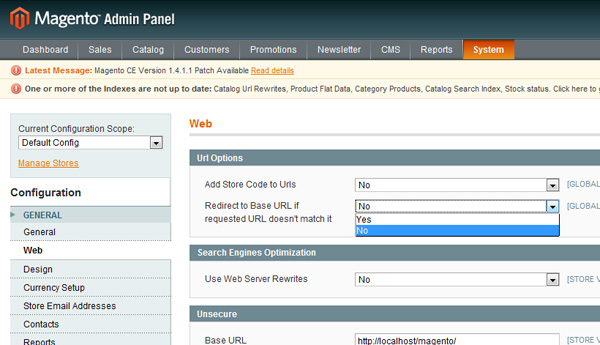
1. SetEnvIf Host .\*base.\* MAGE\_RUN\_CODE="base";
2. SetEnvIf Host .\*magento\_site\_2.\* MAGE\_RUN\_TYPE="magento\_site\_2";

Simple as that! Note that we’ve used the website code we keyed in earlier. If you don’t remember it, just go back in and copy it over. This is very important so make sure typos are avoided.

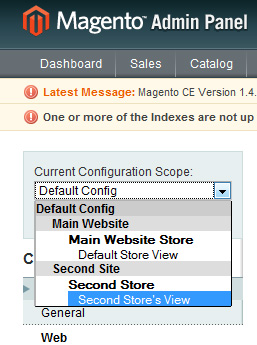
## Step 7: Final Steps

We’re pretty much done now. Just the final few steps before we get everything working.

Go to System->Configuration and choose the web tab.

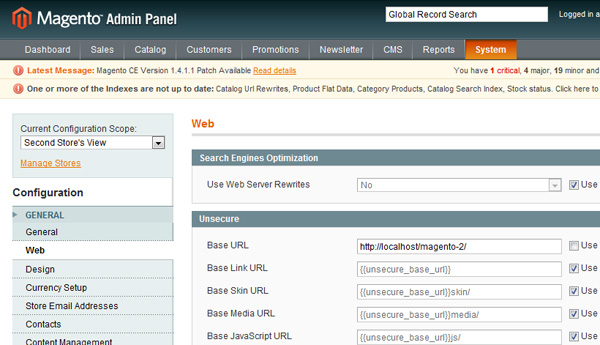


As shown in the picture above, please change the redirect to base URL to “no.”



If you’ve noticed that the select element on the left has extra options now, you get a cookie! Now that we have different stores, we can now change their settings and store policies on a per store basis.

Access our second store’s view and choose the web tab.



Uncheck the use website checkbox and change the base URL to the URL of your second domain. For this article, I did everything on my local server so I have a sub-folder imaginatively named magento-2 as a container for my second store front.

Click on save. And that’s about it. Go to your second domain and start browsing through your catalog!

## What We’ll be Doing in the Next Part

All that’s left now is creating a custom module; then, and I think by then this series will have runs its course. I’m thinking of finishing off with a Magento tweaks and optimization how-to, if enough people show interest for such an article.

If you do feel that something else needs to be covered before this series comes to a close, don’t hesitate to leave a comment below to let me know!

## The Last Word

And we are done! Today, we learnt what Magento’s multi store functionality is and how to set it up as quickly as possible. Hopefully this has been useful to you and you found it interesting. Since this is a rather new topic for a lot of readers I’ll be closely watching the comments section so chime in there if you’re having any doubts.

Questions? Nice things to say? Criticisms? Hit the comments section and leave me a comment. Happy