[Magento](http://www.google.com/url?sa=t&source=web&ct=res&cd=1&url=http%3A%2F%2Fwww.magentocommerce.com%2F&ei=rAjcSe_TDN2mmQfn-u2gDg&usg=AFQjCNE-Pi0CSCkT_zzkJ8EEXQhcI0Ueag&sig2=lz5tJKtRFpJUPILqt4TAXg) is the next generation of eCommerce! It is incredibly well architectured, fully flexible, scalable, and fun to work with. If you’ve ever thought about creating an online shop, Magento is your choice! Today we are going to install it locally with XAMPP Lite and review the essentials.

**Step 1 – What’s Magento?**

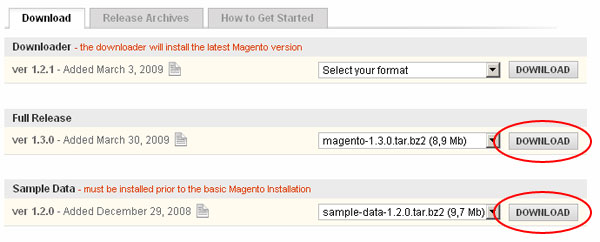


*“Magento is a feature-rich, professional open-source eCommerce solution that offers merchants complete flexibility and control over the look, content, and functionality of their online store. Magento’s intuitive administration interface contains powerful marketing, search engine optimization and catalog-management tools to give merchants the power to create sites that are tailored to their unique business needs. Designed to be completely scalable and backed by an extensive support network, Magento offers companies the ultimate eCommerce solution.”*

That’s Magento in their own words. Very cool about Magento is the concept behind it. Due to its high OOP style, you can  
easily integrate it with your own solutions.

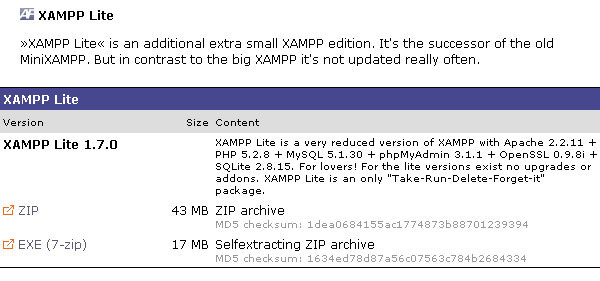
**Step 2 – Downloading Magento**

First of all we are going to download Magento. Browse to [magentoecommerce.com/download](http://www.magentoecommerce.com/download) and select your preferred format of Magento. You can choose between a smaller “Downloader” version which will get the files during the installation process and a “Full Release” which already contains all files. Take the sample data as well since they are helpful for getting used to Magento



**Step 3 – Downloading XAMPP Lite**

XAMPP Lite is a Take-Run-Delete-Forget-it package which doesn’t require an installation. XAMPP Lite contains Apache, MySQL 5.1, PHP 5.2, phpMyAdmin 3.1 and much more. It is an extra small edition of its big brother  
XAMPP. It brings (almost) everything we need for Magento. You can neither take the zip file or the selfextrackting zip archive: <http://www.apachefriends.org/en/xampp-windows.html#646>



After you are done downloading, extract XAMPP Lite to your HD like so X:/xampplite (where X is the letter of your partition).

**Step 4 – Setup XAMPP Lite**

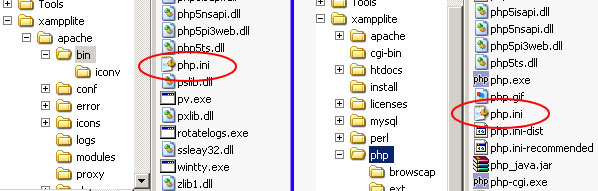
Magento makes high demands on the servers hardware and software. That’s my we need to adjust XAMPP Lite a bit. Some extensions must be loaded in additionally. Within the **two** files D:\xampplite\apache\bin\php.ini and D:\xampplite\php\php.ini we need to uncomment 4 lines of code

In **both** files search for

* ;extension=php\_curl.dll
* ;extension=php\_mcrypt.dll
* ;extension=php\_mcrypt\_filter.dll
* ;extension=php\_mhash.dll

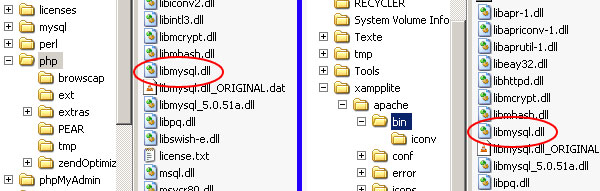
…and remove the semicolon to uncomment them like so:

* extension=php\_curl.dll
* extension=php\_mcrypt.dll
* extension=php\_mcrypt\_filter.dll
* extension=php\_mhash.dll

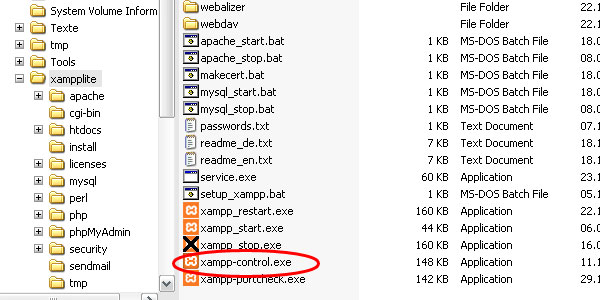


**Step 5 – Replace libmysql.dll**

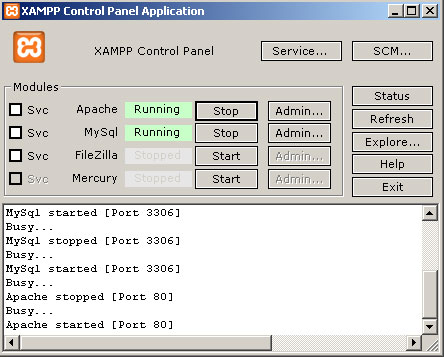
In order to get Magento working we need to replace the libmysql by a newer version. Simply download <http://windows.php.net/downloads/snaps/php-5.2-win32-VC6-x86-latest.zip> and copy the included libmysql.dll to the folders D:\xampplite\apache\bin\ **and** D:\xampplite\php\.



**Step 6 – Start XAMPP Lite**

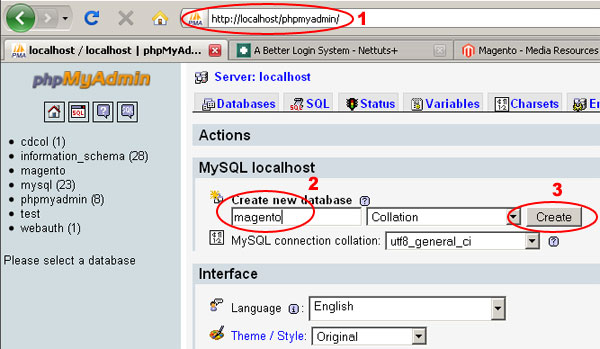


Click on D:\xampplite\xampp-control.exe to start the control panel. Within that panel you can launch the Apache server and the MySql server.



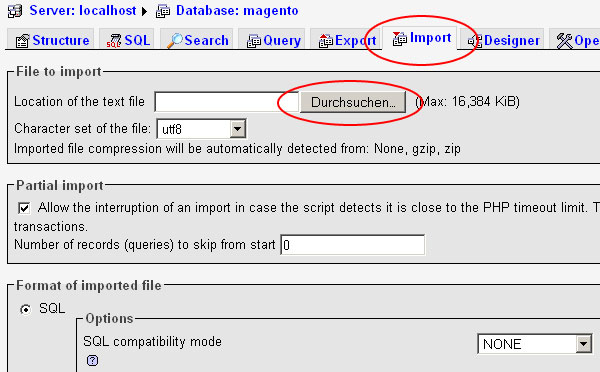
**Step 7 – Creating new DB**

Browse to 127.0.0.1/phpmyadmin to open phpMyAdmin. Choose the name of the database (‘magento’ is a good choice ;-) and click ‘create’.

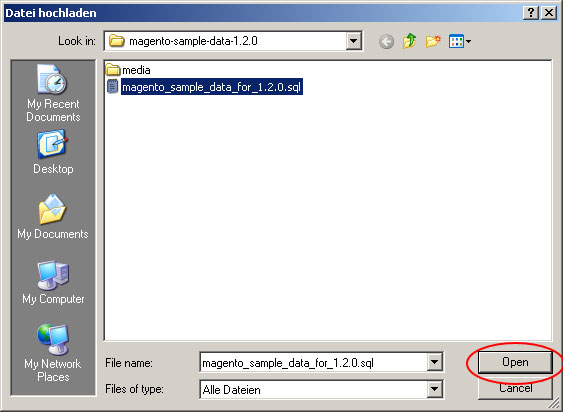


**Step 8 – Installing sample data**

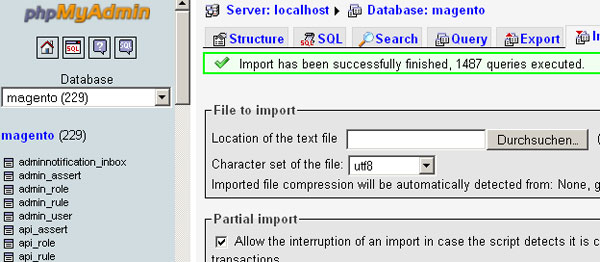
Magento’s sample data have to be installed **before** installing Magento itself. Otherwise there will be an error.



Got to the tap ‘import’ and choose magento\_sample\_data\_for\_1.2.0.sql which you downloaded in Step 2.

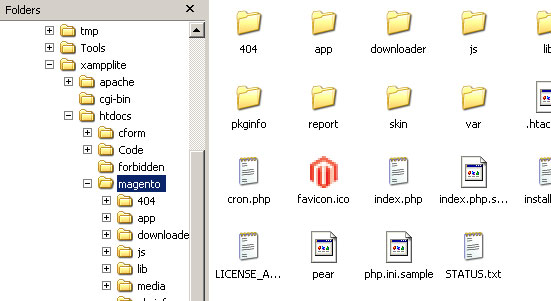


Then click ok to execute the import. This will take a while (at least 20 seconds). So please be patient and do not interrupt that process. The result should look like this:

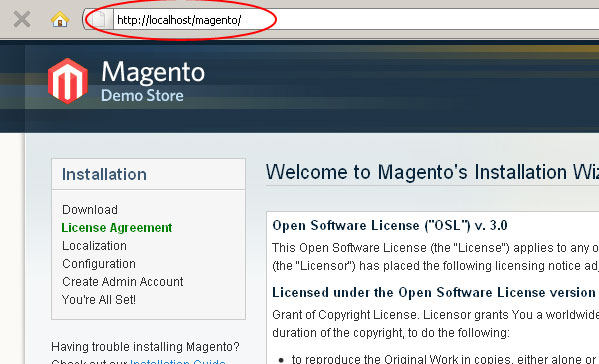


**Step 9 – Installing Magento**

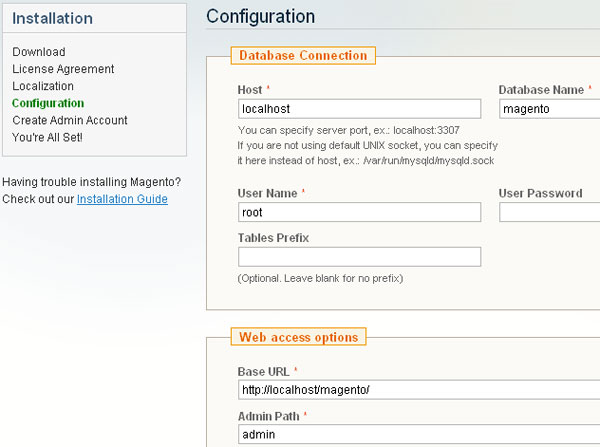
First extract magento-1.3.0.tar.bz2 that you downloaded in Step 2 to your htdocs folder: D:\xampplite\htdocs\magento\. Make sure all files are in that folder, including the index.php. Your directory structure should look like this:



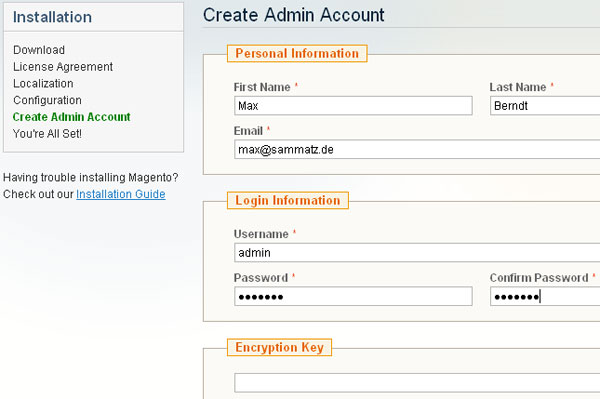
Then open localhost/magento in your browser. Magento will now start the installation wizard. First you will need to accept the license.



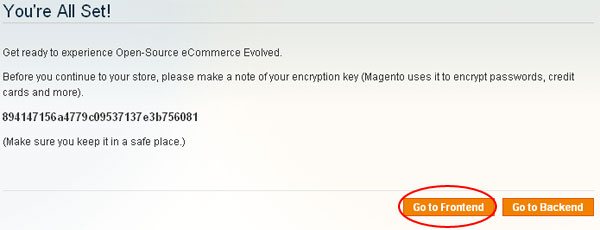
Choose your Localization and your currency and go on. Under ‘Configuration’ you can leave every thing as it is. In a productive Environment you would need to enter the admission data you got from your hosting company.



In the next step you need to create your admin account. Enter your personal data and your login information. Your can leave the field for ‘Encryption Key’ blank.



After you confirmed your Login Information with ‘next’ you finally finished the installation.



**Step 10 – Make yourself familiar with Magento’s frontend**

When you click on ‘Go to Frontend’, it will redirect you to localhost/magento/. Now you can discover the store with its sample data, add products to cart or try Magento’s famous one-page-checkout. Just play a bit with your fresh Magento Store.

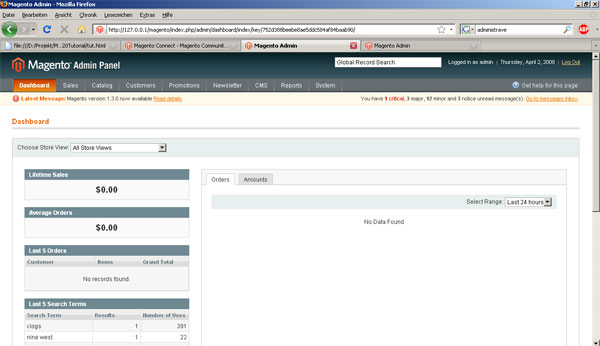


**Step 11 – The Admin Backend**

When going to Magento’s administrative user interface, the backend, make sure to use 127.0.0.1 instead of localhost. Some browser do not save cookie information when there isn’t a dot in the domain. Magento needs cookie information to identify you as approved to the backend.



Visit http://127.0.0.1/magento/admin/ to login. Once you entered your Account Information you’ll be redirected to the Dashboard.



**Step 12 – Intro to Themes**

You can install a theme through a platform called “Magento Connect”. A theme in Magento is responsible for the visual output.

Magento has a very powerful theme concept. For instance, you can use a default theme which comes with Magento and a new one of your own side by side. That new individual theme only contains files which differ from the default theme.

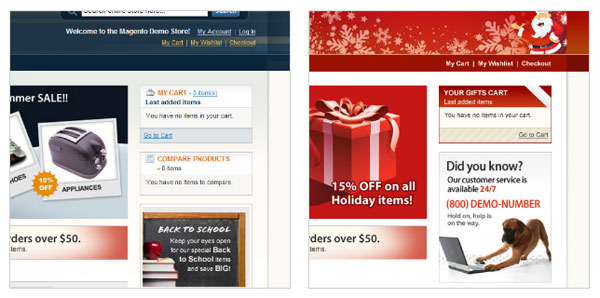


If you simply want to move the mini cart from the left sidebar to the right sidebar you just copy the according file from “default” to your new theme directory and adjust some code. Magento first loads all files from your private theme and then falls back to the next theme lower in hierarchy (in this case “default”) and requires all missing files from that theme.

Doing so, Magento conserves the ability to be upgraded. With every Upgrade pushed out by the Magento Team, files in your default theme are getting overwritten. Your adjusted files are protected this way.

**Step 13 – Where to begin with themes?**

Magento themes are split into two directories. When working with themes, these two directories will remain your base starting point.  
The files are separated into “web accessible” (such as image and Javascripts) and those that can be hidden from it for security reasons.

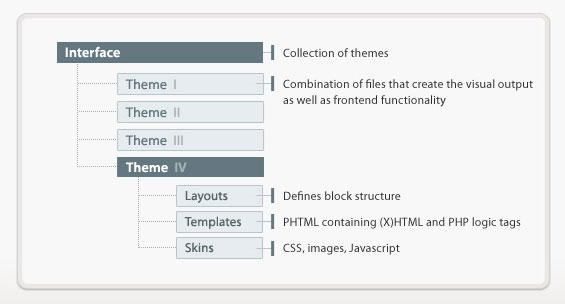


* Directory 1: app/design/frontend/default/default/ — This directory contains the layout, translation (locale) and template materials.
* Directory 2: skin/frontend/default/default/ — This directory contains the images, CSS and block-specific Javascripts.

**Step 14 – Interface and theme**

An interface is a collection of themes. Let’s have a look at the two directories as they represent the structure of interfaces and themes:

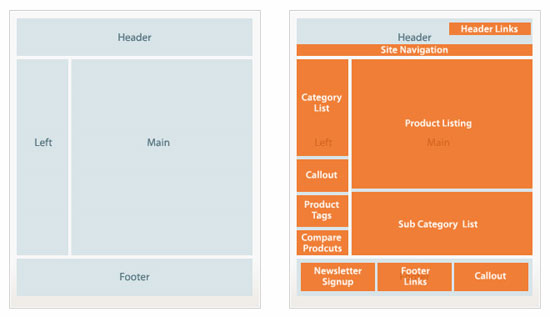
* Directory 1: app/design/frontend/default/default/
* Directory 2: skin/frontend/default/default/



In both cases, default indicates the interface name, and default indicates the theme name. So if you were working on a theme called “my\_theme” in an interface called “my\_interface”, you would be working in the ‘app/design/frontend/my\_interface/my\_theme/’ directory.

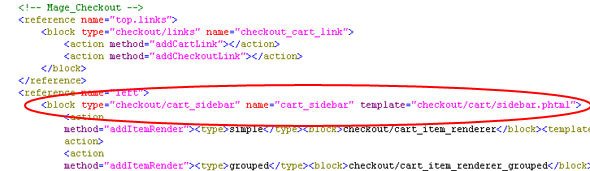
**Step 15 – What are Blocks?**

In Magento we work with structural blocks and content blocks. Structural blocks represent the basic structure of a page. Usually we have structural blocks like header, left sidebar, middle content, right sidebar and footer. We can assign **content blocks** like category list, navigation, search bar, callout etc. to a certain **structural block**.



**Step 16 – Intro to Layout**

Content blocks are assigned to structural blocks with the help of **layout**. Layout is build with XML files and can be found under app/design/frontend/default/default/layout. This way you can reuse earlier created templates, such as the mini cart module on the sidebar, on different pages by simply calling them within a layout xml file.



**Step 17 – Roundup**

You learned how to setup Magento and the basics of Magento’s design terminologies. This will help you diving into that wonderful Open Source platform. To dive deeper into Magento, I recommend the following resources – which cover the fundamentals.

**Main Resources**

* **http://www.magentocommerce.com**
* **http://www.magentocommerce.com/design\_guide**
* **http://inchoo.net/wp-content/uploads/2008/06/designers\_guide\_to\_magento.pdf**
* **http://www.magentocommerce.com/media/screencasts**
* **http://www.magentocommerce.com/media/tour**

**Books**

* [php|architect’s Guide to E-Commerce Programming with Magento](http://www.amazon.com/architects-Guide-E-Commerce-Programming-Magento/dp/0973862173/ref=pd_bbs_sr_1?ie=UTF8&s=books&qid=1239100138&sr=8-1)
* [Magento: Beginner’s Guide](http://www.amazon.com/Magento-Beginners-Guide-William-Rice/dp/1847195946/ref=pd_bbs_sr_2?ie=UTF8&s=books&qid=1239100138&sr=8-2)