## Unit 1- Planning for Magento Installation

# 1. Ecommerce- Overview

### 1.1 What is E-commerce?

E-commerce (Electronic Commerce) is a type of business that involves the commercial transaction or purchasing or selling of goods and services through electronic channels known as internet. It was first introduced in the year 1960 through EDI (Electronic Data Interchange) on VAN (Value-added network). Using E-commerce, you can sell physical products or services (where the payment is made online).

*Following are the categories of E-commerce:*

**B2B (Business to Business):** This transaction is between businesses. It is between the wholesaler and the retailer or between the manufacturer and the wholesaler.

**B2C (Business to Consumer):** This transaction is between businesses and consumers. In this type of transaction, merchants sell products to consumers through shopping cart software.

**C2C (Consumer to Consumer):** In this type of transaction, one consumer or customer interacts with other consumers through internet.

**C2B (Consumer to Business):** This transaction is between the consumer or the customer and businesses or organizations where the consumer makes a product that the organization uses it to complete the business.

### 1.2 What is Magento?

Magento is an open source E-commerce software, created by *Varien Inc*., which is useful for online business. It has a flexible modular architecture. It is scalable and it has many control options that helps the user to build both user-friendly and search engine friendly websites.

Magento uses E-commerce platform, which offers companies the ultimate E-commerce solutions and extensive support network. Magento allows user to update E-commerce website automatically. It is simple, quick and versatile to use.

Varien Inc. developed Magento, and first released on *March 31, 2008.*

### 1.3 Why to Use Magento?

Magento is an open source E-commerce software.

Magento is scalable and offers small organizations to build business.

Magento enables searching and sorting of products in several ways.

Magento easily integrates with many of the third-party sites, which are needed to run effective E-commerce website.

## 1.4 Features

Magento provides different payment methods such as credit cards, PayPal, cheques, money order, Google checkouts, etc.

Magento enables shipping of products in one order to multiple addresses.

Magento helps to manage the orders easily by using the admin panel.

Magento provides order of product status and history of product. It also supports e-mail and RSS feeds.

Magento supports multiple languages, different currencies and tax rates.

Magento filters the products and displays in grid or list format.

Magento makes it easy to browse the products. It has features such as image zoom-in and checking of stock availability.

Magento has built-in SEO (Search Engine Optimization).

## 1.5 Advantages

Magento is user friendly E-commerce software.

Magento is compatible with smartphones, tablets and other mobile devices.

Magento provides multiple payment options, so every visitor can make payment based on their preferred payment gateway.

Magento has many extensions which support the development of an online store.

## 1.6 Disadvantages

Magento uses larger disk space and memory.

Magento takes longer time to build the customized functionality.

Magento is very slow zompared to other E-commerce sites.

Magento needs proper hosting environment. If the hosting environment is improper, the user can face many problems.

# 2. Magento – Installation

Step-by-step procedure for Magento installation. Before installing Magento, you require the following system requirements.

## 2.1 System Requirements for Magento

Database: MySQL 5.1+

Web Server:

- Apache 2.x
- Nginx 1.7.x

Operating System: Cross-platform

Browser Support: IE (Internet Explorer 7), Firefox, Google chrome

SSL (Secure Socket Layer): A valid security certificate is required for HTTPS

PHP Compatibility: PHP 5.4+

**Refer below link for detailed PHP extensions requirement**

http://devdocs.magento.com/guides/v2.3/install-gde/system-requirements-tech.html

## 2.2 Download Magento Zip file from https://magento.com/tech-resources/download


**Download Composer**

Refer Magento Install ubuntu 16.04.docx for **detailed description**.

# 3. Magento – Architecture

Magento has a wonderful architecture behind its system. It's a wonderful combination of Zend Framework and MVC Architecture. That is the reason why Magento is a professional open-source eCommerce solution offering unprecedented flexibility and control.

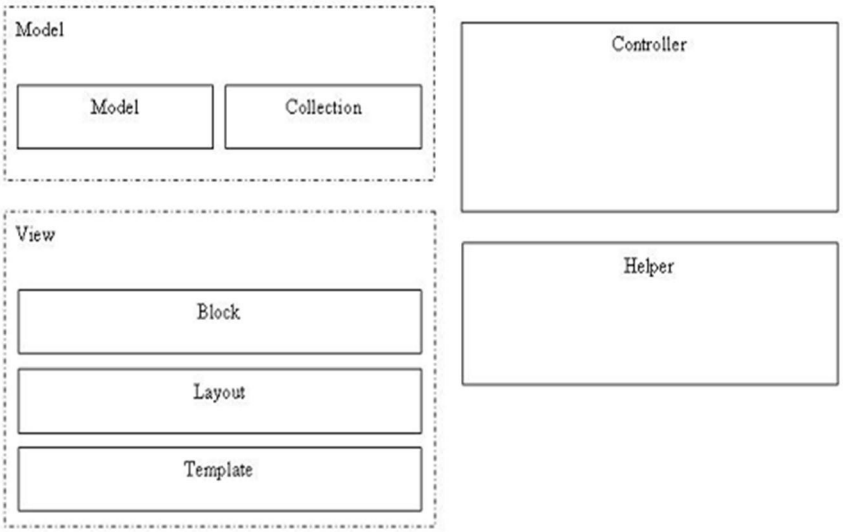
Using this framework, Magento was built with 3 central tenets in mind.

– **Flexibility**: We believe each solution should be as unique as the business behind it. Magento's code allows for seamless customizations.

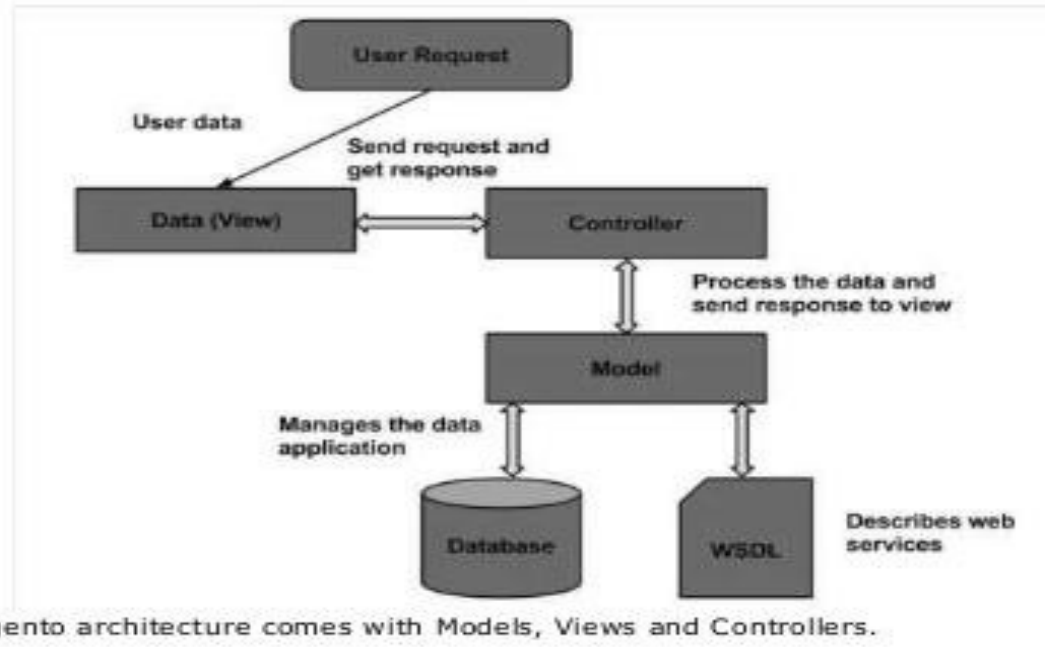
– **Upgradeable:** By separating the core code from community and local customizations, Magento can be easily customized without losing the ability to upgrade.

– **Speed and Security:** The coding standards used by the developers follow best practices to maximize the efficiency of the software and provide a secure online storefront.

Magento Architecture follows MVC frameworkk and Views itself is broken into 3 parts, the model into 2 and you have controllers and helpers where helpers are module specific. If you are thinking that why helpers are module specific then you should know that Magento helpers and in fact all the models and controllers extends Magento core controllers, models and helpers so the common features are there in the super class whereas you can also add module specific features which you need not share to make big helper classes.




## Block Diagram




The Magento architecture comes with Models, Views and Controllers.

**Architectural Diagram**

**User Request:** The user sends a request to a server in the form of request message where web browsers, search engines, etc. act like clients.

**View:** View represents the data in particular format. It is the user interface which is responsible for displaying the response for user request. It specifies an idea behind the presentation of the model's data to the user. Views are used to reflect "how your data should look like".

**Controller:** The controller is responsible for responding to user input and perform interactions on the data model objects. It uses models to process the data and send responses back to the view.

**Model:** The model is responsible for managing the data of the application. It contains logic of the data and represents basic data object in the framework. It responds to request from the view and to the instructions from the controller to update itself.

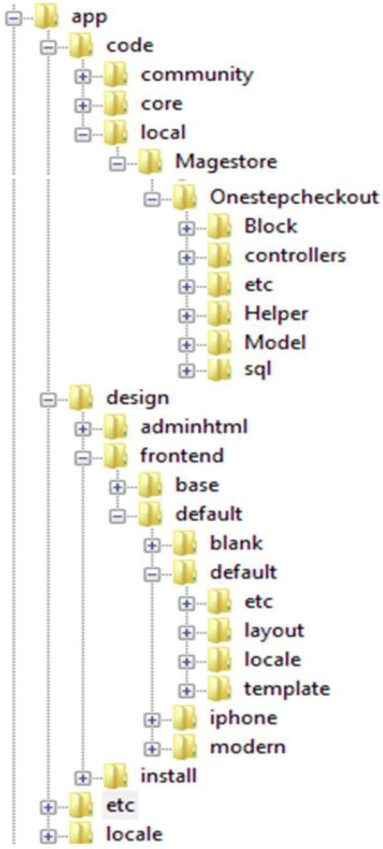
**Database:** Database contains the information which is requested from the user. When the user requests data, view sends requests to the controller, the controllerrequests from the model and the model fetches the required information from the database and responds to the user.

**WSDL**: WSDL stands for Web Services Description Language. It is used for describing web services and how to access them.

# 4. Magento Folder Structure

Magento is constructed based on object-oriented programming under MVC. The Magento code system is stored in the form of dispersion in order to increase the expansion ability for the system. A Module Structure in Magento. A module is an extension which extends the features and functionality of Magento. You are probably familiar with the idea of modules from other software, but if not, some concrete examples of modules would be additional payment gateway integrations, or a featured items promotional tool.

Main folders in Magento module are:




Code folders of a module which are usually written in the folder app/code/local include:

– **<Namespace>**

**<Modulename>**

Block: folder containing blocks

Helper: folder containing the help files

controllers: folder containing module's controllers

etc: folder containing module's configuration files

Model: folder containing module's models

sql: containing module's data installation files

– *Namespace:* is a unique name that identifies your company or organization. The intent is that each member of the world-wide Magento community will use their own Package names when creating modules in order to avoid colliding with another user's code.

– *Block:* the driver behind the arrangement of Magento templates.

– *controllers*: containing control layers of application flow. They receive input which is users' requests via HTTP header and then forward the request to the layer directly in charge of processing requirements.

– *etc*: containing configuration files of a module (XML file). Based on the configuration file, we can set up or overwrite the old settings just by placing the correct XML tags.

– *Helper*: containing utility methods that will allow you to carry out common tasks on objects and variables.

– *Model*: Layers providing data and services related to data and business logic. These classes work directly with the database and provide data for other components.

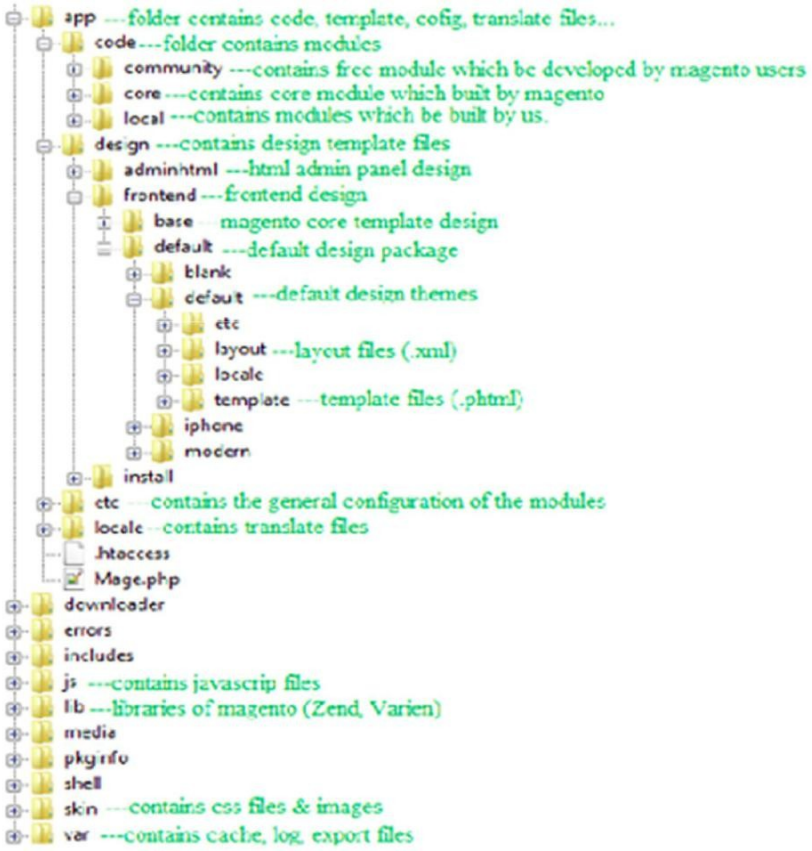
– *sql*: containing files of installation and database update of the module database.

Besides code folders, we have folders containing files about interface, configuration, language, etc.

– *layout*: (*/app/design/frontend/packagename/themesname/layout)*. Layouts are basic XML files that define the block structure for different pages, as well as controlling the META information and page encoding. Layout files are separated on a per-module basis, with every module bringing with it its own layout file.

– *template*: (*/app/design/frontend/packagename/themesname/template).* Templates are PHTML files that contain (X)HTML markups and any necessary PHP tags to create the logic for the visual presentation of information and features.

– *Skins*:*skin/frontend/packagename/themesname)* default Skins are block-specific JavaScript and CSS and image files that compliment your (X)HTML. What are blocks you ask? Good question, and don't worry, we are almost done defining the components of Magento.



# 5. Magento Design Concepts and Terminology

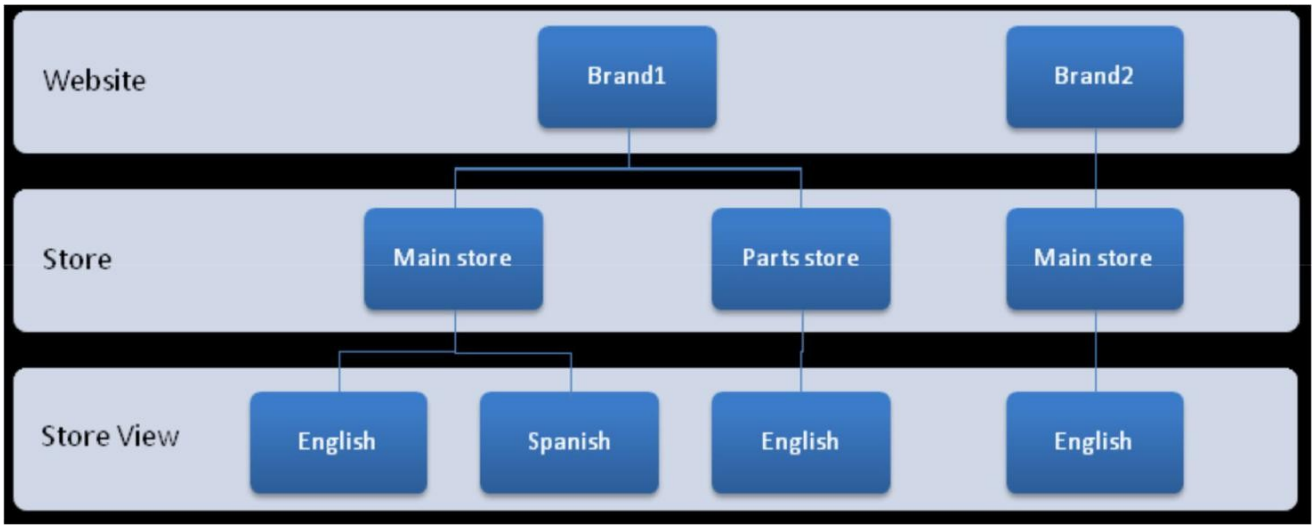
Magento supports the creation and management of multiple stores in a single Magento installation, Magento has a hierarchy of concepts that define the relationship between the individual stores in a Magento installation.

In Magento, a **website** is a collection of **stores**, which themselves are collections of **store views**. These layers, although perhaps initially confusing, provide you with powerful flexibility when setting up online businesses in Magento.

A **website** is made up of one or more stores, which share the same customer information, order information and shopping cart.

**Stores** are collections of store views and can be setup in a variety of ways. Their main function is to provide a logical container that allows you to group related store views together in a website.

**Store Views** are the actual store instances in Magento. Most stores will have a single store view associated with them. But a store can also have multiple store views, which are typically used for different languages. Therefore, if you wanted to have a store displayed in English and Spanish, for example, you could create the store once and create two different store views for that store.




**Hierarchy of websites, stores and store views in Magento.**

## SCENARIO 1—ONE STORE

A company named “Bongo’s Instruments” wants to create an online presence. Bongo has a single catalog and does not need to support multiple languages. This is the simplest scenario in which Bongo’s Instruments is the *website*, *store*, and *store view*.


Figure 2. Single website, store and store view.


Figure 3. Single website with multiple stores.

## SCENARIO 2—MULTIPLE INDEPENDENT STORES

A company named "**My Laptops**" wants an online presence with two separate stores that both sell laptops but at different prices and with different product selections in some categories.

They also want to offer English and Spanish language options per store. Within each store they need to synchronize customer and order information, but they do not need to share this information between the two stores.

In this scenario,My Laptops would have two websites (which stops customer and order info from being shared with stores in the other website), each with one store and two store views (one for English and one for Spanish).


Figure 4. Multiple websites with multiple stores and store views.

# 6. Overview of Magento2 Components

Magento 2 shows an impressive advancement in the e-commerce field by bringing some key features and making more efficient resource use of the current generation of Servers. Here are a few of the improvements we mentioned:

**Increased Speed –**The full page caching provides an immense boost in loading speed on sequential page visits while the optimized code makes Magento 2 25% faster right after installation compared to version 1.9

**Increased Security –** The strengthened hashing algorithms (SHA-256) help in password management while the X-Frame-Options HTTP request header provides protection against clickjacking attacks. There are also new settings regarding mode switching (developer/production) that restrict permissions on application files.

**Improved Scaling Potential** – Database separation allows for dedicated databases servicing specific store features thus increasing the potential load the platform can take. On another note, the overall Scalability of the Catalog has been improved by more than 1800%

**Simpler and Faster Checkout Process** – During our tests, we managed to checkout from Magento 2 via a credit card in around 40 seconds from start to finish plus the loading times which is extremely fast. This was possible due to a few tweaks in the form fields and the removal of the credit card type menu which is obsolete nowadays. Making the checkout process seamless, is a great improvement that will surely increase the usage of the application.

## 6.1 Components of Magento

**Introduction**

Catelog Management

Site Management

Very easy checkout

Payment gate

Search engine optimization

Order managemet

A look of Magento CMS Structure

Thoughts on Development

### 6.1.1 Catelog Management

Manage Product and prices

Tax rate for location customer and group

Import and export

Upload product from CSV file

### 6.1.2 Site Management

Multiple site managemet (global and local) from same admin panel

Multiple currency setup option

Google SEO management

### 6.1.3 Very easy checkout

One page checkout for order as a result increase in sales

Secure guest address

### 6.1.4 Payment gate

Multiple payment gateways like paypal, amzon and google

Also cheques /money order /purchase order via magento connect

### 6.1.5 Search engine optimization

Autogenerate SEO

Site maps

### 6.1.6 Order Management

Graphs of today's order

Timeline order

Email notification for order

Information to store manager