

Lesson 1 : Using Wordpress as a CMS

INTRODUCTION:

Functionality wise most Content Management System (CMS) can be broadly divided into three unique sections. They are as follows:

1) An Administration Section

A CMS administration section is normally accessed through a 'Challenge and Response' system, (i.e. a Login / Registration system).

A CMS administration system is perfectly capable of recognizing its administrator by validating a 'Login ID and Password' combination and only then permitting access to its internal resources. The internal resources of a CMS are usually a number of tools that permit a user to capture, format and store user information within the CMS DataBase Management System (DBMS).

Once the user information is stored within the DBMS it can be accessed at will using a Graphical User Interface (GUI), updated (i.e. edited), deleted or displayed, on demand using the CMS environment.

2) The Database Management System

All CMSs require some sort of data store beneath the Admin section's graphical interfaces to store and retrieve user and administration data on demand.

The data store can be any one of the many excellent DBMS available for the Internet such as:

- MySQL
- Postgres
- Oracle
- MSSQL

WordPress uses MySQL, which is a free to use, robust, production quality DBMS as its data store.

3) CMS built-in programs that retrieve user data on demand.

All CMS have built-in program that can retrieve user data stored within the CMS data store, on demand. These programs are built to:

- Extract such data from the CMS data store
- Format this data as an HTML page
- Bind a cascading style sheet to the HTML page
- Deliver this to a requesting Browser as a well formatted XHTML and CSS compliant page (Hopefully)

A simple visual representation of a CMS system as shown in diagram 1:

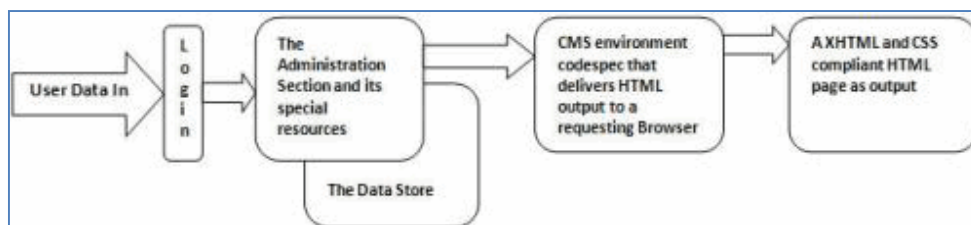


Diagram 1

Although WordPress (henceforth called WP) is widely used as a Blogging environment, at its heart it's a pretty sound, stable, Content Management System (CMS).

To call WP a CMS we should be able to identify all the three sections described earlier, within it. Let's take a quick look at WP to see if we can do this and thus justify calling WP a CMS.

Accessing the WP Admin Dashboard:

The WP Admin section is assessable only via the WP Login system. Please refer to diagram 2.



Diagram 2

Only when you've successfully logged in as the WP administrator, the **WP Dashboard** is displayed in your Browser window.

On click of specific links within the WP Dashboard, the WP - WYSIWYG editor is accessed via which user content is captured and saved to the WP database. Please refer to diagram 3.

WP user content is normally of **two** types:

BLOG Posts

STATIC Pages

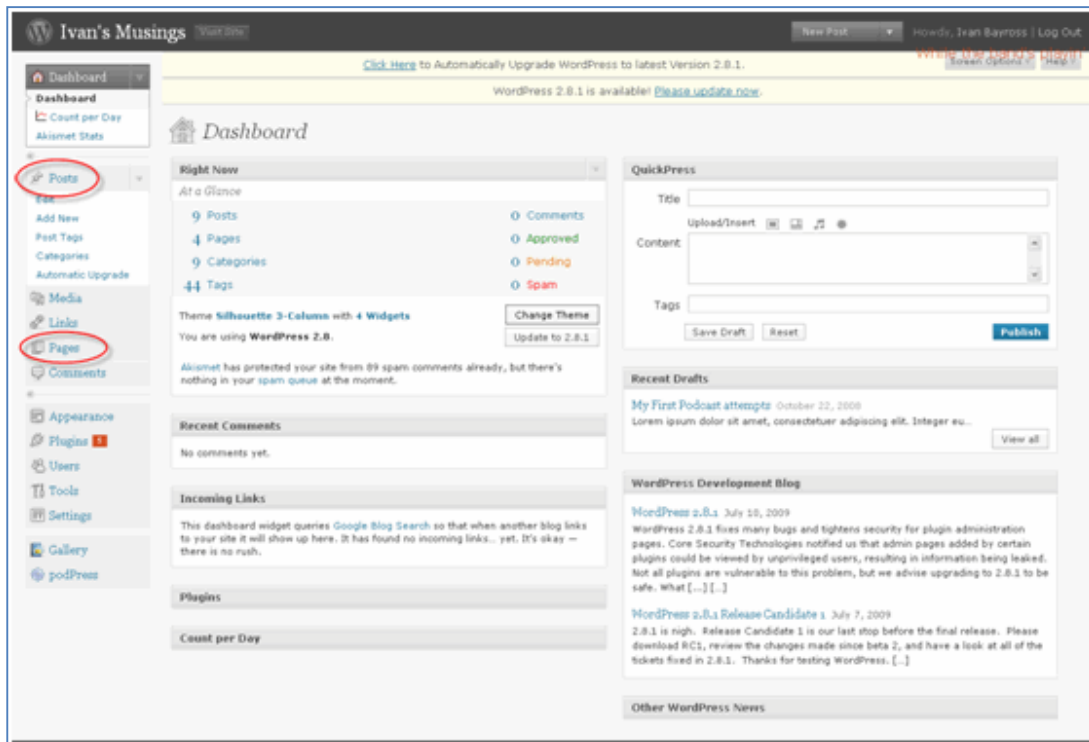


Diagram 3- The WP Dashboard

NOTE: the Posts / Pages links on the right hand side of the WP Dashboard U.I. encircled in red.

Capturing User Information in WP:

When either of these links is clicked, they open out to display sub links. Please refer to diagrams 4 and 5.

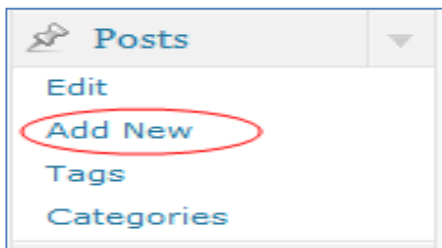


Diagram 4 On click of the Posts link

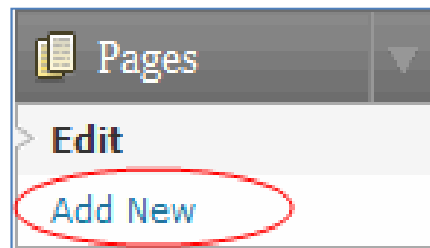


Diagram 5 On click of the Posts link

When the link **Add New** of either Posts or Pages is clicked the WP - WYSIWYG editor is displayed in the Browser window. Please refer to diagram 6. Using this editor, user content for either a BLOG Post or a STATIC page is entered, formatted as desired and saved to the WP database.

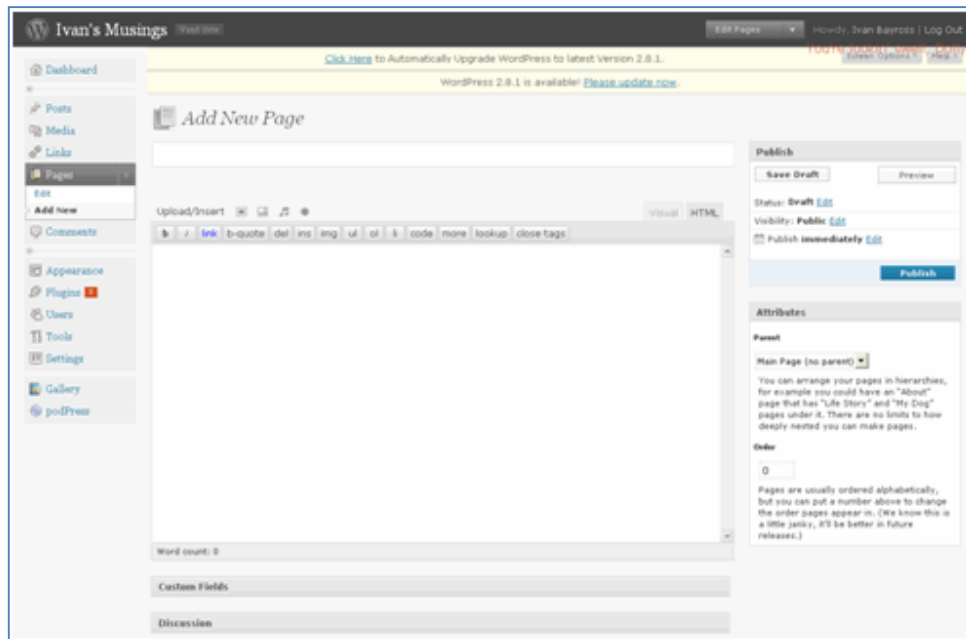


Diagram 6

Recognizing WP as A CMS:

User information is now saved within the WP database system and is available for all further processing until such information is deleted by the Administrator.

The **first** step of capturing user information within a database has been achieved.

Two other steps are required to qualify as a CMS. They are, on demand, such user information must be:

- 2.Extracted from the database from where it is stored
- 3.Displayed to a site visitor within (a XHTML and CSS compliant) web page

WP does both **2** and **3** exceedingly well.

Please refer to diagram 6 for an example of the output of steps 2 and 3 by WP.

WP Delivering User Information on Demand:

If you visit my Blog, Ivan's Musings, and click the HOME link (i.e. **on visitor demand**), WP extracts the information I've stored, associated with my HOME page, from its data store and serves it to your Browser as an HTML page (HOME content enclosed in the **red outlined box** in diagram 7).

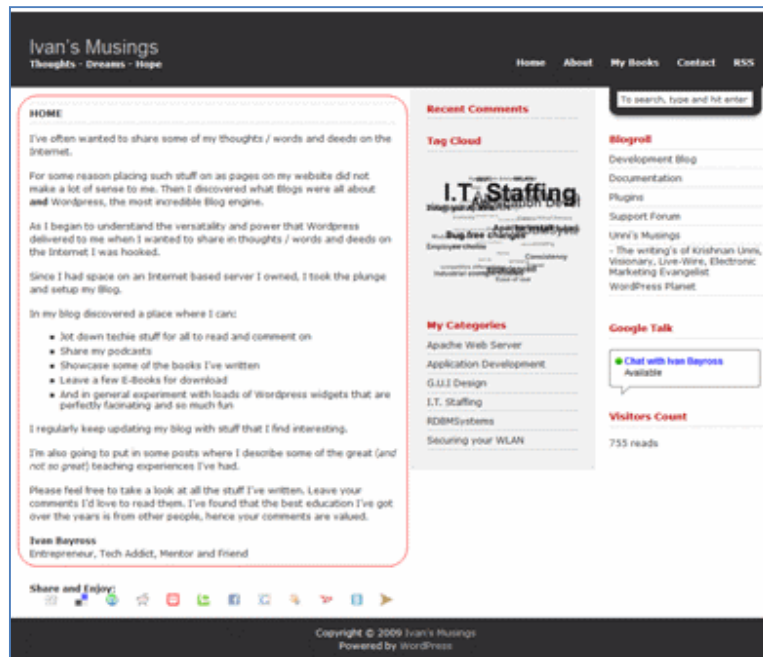


Diagram 7

Hence WP does behave as a **Content Management System** and can be used to build websites. As a bonus, WP is pretty light on system resources when compared to most other CMS systems available today.

Installing, setting up and configuring WP are so simple that most non-techies can do it themselves. For help, there are lots of tutorials (Video, Audio and Text based) on the web indicating how this is done.

The tons of excellent WP plug-ins and widgets that extend the functionality of the WP framework, add huge value when using WP as the CMS of choice for building interactive websites.

Creating E-Commerce websites complete with shopping carts linked to payment gateways (eg. PayPal / Google pay and others) really are a snap using WP, when compared to hand coding them from scratch or using other CMS available which have a much steeper learning curve.

WP being a well matured product, there are several experienced WP developers around who are ready and able to build user specific plug-ins and/or widgets to add some special functionality to a website at very modest prices.

Finally, the key to using WP as a CMS to build, good looking, database driven, interactive websites is to think very slightly out of the box.