

**Sussex Geek Dinner  
February 28<sup>th</sup> 2007**

**An Introduction to the Expression Engine CMS  
or ‘the designers friend’**

Notes for V1.5.2

**Geoff Adams  
[www.oakenpage.com](http://www.oakenpage.com)**

## Contents

<b>1. EXPRESSION ENGINE [EE] TALK BACKGROUND.....</b>	<b>3</b>
<b>2. INTRODUCTION.....</b>	<b>4</b>
<b>3. ELLISLAB AND EE VERSIONS .....</b>	<b>5</b>
<b>4. EE FUNCTIONALITY OVERVIEW.....</b>	<b>6</b>
4.1 EE CORE VERSION.....	6
4.2 EE FULL VERSIONS.....	7
<b>5. EE – THE FUNDAMENTALS .....</b>	<b>9</b>
5.1 TEMPLATES: AN INTRODUCTION .....	10
5.1.1 <i>Templates and site urls</i> .....	10
5.1.2 <i>Editing templates</i> .....	10
5.1.3 <i>Comments</i> .....	10
5.1.4 <i>Embed templates: basics</i> .....	11
5.2 WEBLOGS .....	12
5.2.1 <i>Weblog is not a web-log</i> .....	12
5.2.2 <i>Field groups define weblog structure</i> .....	12
5.3 CATEGORIES .....	13
5.4 TEMPLATES & WEBLOGS – JOINING THE TWO .....	14
5.4.1 <i>EE Tags</i> .....	14
5.5 EE VARIABLES .....	16
5.5.1 <i>Tag specific variables</i> .....	16
5.5.2 <i>EE variables</i> .....	16
5.5.3 <i>Dynamically assigned global variables [for reusability]</i> .....	17
5.5.4 <i>User defined global variables [for ease of maintenance]</i> .....	17
5.5.5 <i>Conditional variables [for power]</i> .....	17
5.6 EXTENDING YOUR TEMPLATES.....	18
5.6.1 <i>Single entry pages and urls</i> .....	18
5.6.2 <i>Multy-entry pages and urls</i> .....	18
5.6.3 <i>Template embed variables</i> .....	19
5.6.4 <i>PHP in templates</i> .....	19
<b>6. SUMMARY.....</b>	<b>20</b>
<b>7. APPENDIX: RESOURCES .....</b>	<b>21</b>
<b>8. APPENDIX: TEMPLATE AND TAGS EXAMPLE.....</b>	<b>22</b>
<b>9. APPENDIX – MAGNOM PRODUCT NOTES .....</b>	<b>24</b>
9.1 CONTAMINANTS IN INDUSTRIAL FLUIDS .....	24
9.2 MAGNOM FILTER .....	24
9.3 WEBSITE GOALS .....	24

## 1. Expression Engine [EE] talk background

This is an introduction to the Expression Engine [EE] CMS and to its core concepts – weblogs, templates, categories, variables and EE tags.

Expression Engine is a product of EllisLab [ [www.ellislab.com](http://www.ellislab.com) ] or pMachine<sup>1</sup> as was.

These notes are based upon a talk I gave at the Sussex Geek Dinner [28<sup>th</sup> Feb 2007], in Brighton – one of the UK's centers of web geekery. I have polished these notes in the light of the presentation.

Simon Harriot<sup>2</sup> was in need of speakers for the dinners which are mainly , but not exclusively, attended by back end developers, so I thought that they would be interested in a CMS which is:

- designer friendly
- works very well when creating standards compliant coding
- provides lots and lots of back end access

My perspective is that of a intermediate level user, not long over the main EE 'hump'. At time of writing I have created one EE site for clients [ [www.magnum.com](http://www.magnum.com) ] and one for myself [ [www.oakenpage.com](http://www.oakenpage.com) ].

In particular, I wanted to draw together some of the things that puzzled me when I was first learning, and before I became an EE Guru [or **not**]. ☺

Actually, as you will see, I have not touched vast areas of EE functionality in the talk. The nice thing is that it's there if I need it, without needing to rely on third party stuff.

I've tried to provide a logical structure to understand what EE does, and how to go about using it. The real nitty gritty stuff starts in section 5.

Disclaimer: I have tried to be accurate, but do please do check. This was written using version 1.5.2 [current release]. The next version will be version 2, which promises to be a very significant update!

Having said that, Leslie Camacho from EllisLab has checked these over, so I'm pretty confident that it is right and true.

---

<sup>1</sup> The company name used to be pMachine, but the name was changed in March 2007 to EllisLab, as pMachine was also the name of the original product from this company. In this case, 'a rose by any other name is just as sweet'. BTW I have not updated any graphics which show the word pMachine!

<sup>2</sup> Our kind host & organiser - [www.sussex.geekdinner.co.uk](http://www.sussex.geekdinner.co.uk)

## 2. Introduction

How I came to use Expression Engine - I had a project, which definitely required a CMS.

My client manufactures an excellent device for filtering ferrous particles from industrial fluids [*see Appendix 9: it is well worth a read for anyone at all interested in modern industry*]. There are a number of products, covering a range of industries and cleaning a wide range of fluids in differing operating environments – for which they had many case studies. Clearly this required a complex interlinked site – creating/maintaining that manually would be a full time job – and that mandated a CMS – but which one?

### Some background

I've been developing scientific software for over 20 years. During that time I have developed a keen sense of the importance of:

- Good user interface
- Good architecture [data structures etc]
- Planning for maintainability

I developed my first website in 98 – a table driven monstrosity. That experience put me off for 5 years, until I came across standards based sites using CSS. Since then I have been developing standards compliant sites using CSS/HTML.

Whilst maintaining some sites, one can quickly see the potential of a CMS. Problem is that most did not play well with standards based coding ☹

So for this project my requirements for a CMS were:

- Must play well with a standards based coding.
- Must not to force me into any particular theme [i.e. complete separation between user facing pages and backend]
- Must allow me to easily set up the backend architecture I would need
- Not to require PHP etc [though good to have access if needed]
- Not to require use of hundreds of plugins of dubious origin
- To be robust – not to have lots of code forks or be requiring constant updates
- To be secure – I did not want to be continually updating/patching!
- To have good support
- Reasonable price

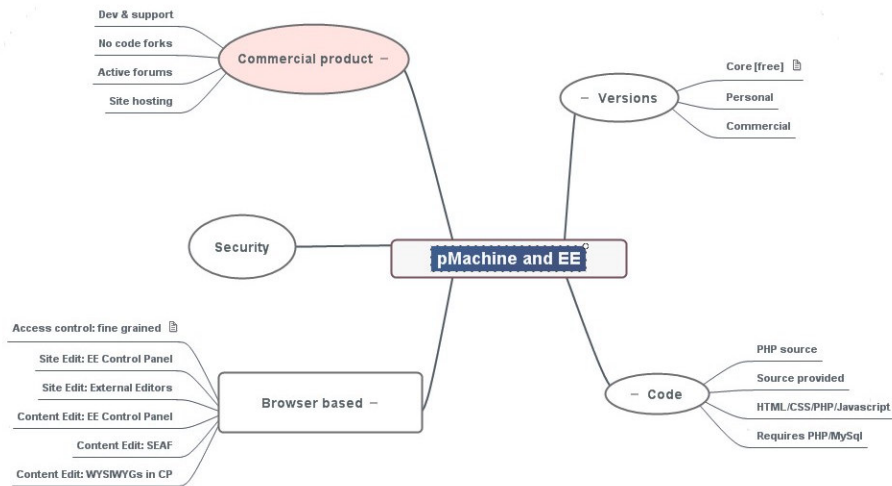
I had used WordPress before and felt that I would have to go through flaming hoops to try to get it to do what I wanted/needed. I had a set of well known<sup>3</sup> CMS's I was thinking about and for various reasons was heading toward Text Pattern – when I saw some very favourable comments on EE, and also saw the results of the EE \$15000 shootout [<http://www.pmachine.com/shootout/>]. There were no 'EE look and feel' sites there – most were standards compliant. I investigated further, went ahead and was not disappointed!

---

<sup>3</sup> One EE recent forum thread which has some informal comparisons is <http://www.ellislab.com/forums/viewthread/38244/>.

### 3. EllisLab and EE versions

About EllisLab, the company responsible for EE, the different EE versions, code and how you use it.



EE is a commercial product. There is a small full time development team and support team who all frequently appear in the very active forums [useful & pleasant – very helpful to newbies and little in the way of flame wars]. There is also good documentation.

This has the advantage that there is a single controlling vision, features don't get lost, etc.

There are three versions: [prices as of Feb 2007]

- Core version – **free** but somewhat limited<sup>4</sup> in capabilities [no forums, galleries, wikis, ecommerce]
  - Personal [\$99.95]– full program – but site must be for non-profit purposes
  - Commercial [\$249.95] – full program
- [also a \$10 hosted 30 day trial – no need to setup, that all done for you]

**You do need to get a fresh appropriate license for each site**, and there is a modest annual renewal fee [though site will still keep working without it] [\$20 – personal, & XXX for commercial]

**EE is written in PHP.** The source code is provided and you can therefore modify it if required [obviously at own risk!] – better to create an extension or plugin. EE requires PHP/MySql, and it supports HTML/CSS/PHP & Javascript coding.

**EE is browser based**, so you install EE at the website and then browse to the login page. This opens the control panel [CP] where you can do everything [if you want]! CP access is fine grained, so you can limit lower level users to what they can do, or even see!

In the CP, you can create and edit entries, define & modify page templates, setup membership levels – do backups etc etc.. At present the CP does not use AJAX – but there seem to be hints on that front.

EE has an **excellent reputation for security**:

- 1 minor exploit in its history [over three years]
- lots of anti spamming tools for blogging [captcha's, black/white lists etc]

<sup>4</sup> Still pretty powerful – I could have done maybe 90% of [www.magnum.com](http://www.magnum.com) in EE Core

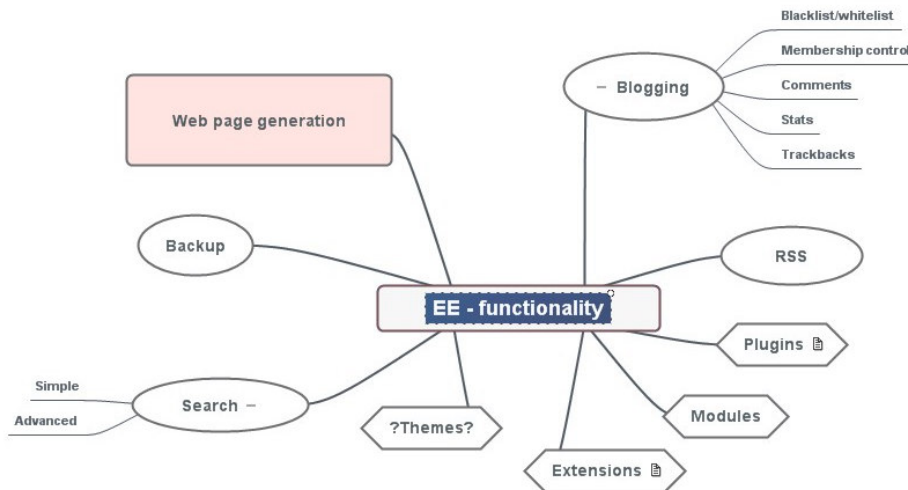
## 4. EE functionality overview

Talking about the:

- Core of EE [whats in the free version]
- Extra stuff available in personal/commercial versions

### 4.1 EE Core version

The Core [free] version offers a **lot** of good stuff:



At the heart is the web page generation stuff – templates, weblogs, categories etc. I'll come back to this in detail later on.

**Blogging**  – there are a number of modules that support blogging in various ways:

- Commenting [viewing existing, creating and previewing]
- Membership [signup etc]
- Trackbacks
- Stats [visitors, last visited, entries, guests etc etc]
- Blacklist/whitelist – part of spam prevention arsenal [includes EllisLab's own black lists]
- Nation from IP [either for blacklisting or for showing national identity]

**RSS** – provides templates for various form of RSS [RSS2 & Atom]

**Plugins** – not part of EE. Provide new tags for the EE language – i.e. enhance templates. Currently over 90 plugins [many from EE team]. Not nearly as many as WordPress, mainly because so much functionality is already in the main EE modules.

**Extensions** – not part of EE. More fundamental than plugins. These use clearly defined hooks in code to actually change functionality, for example in the Control Panel.

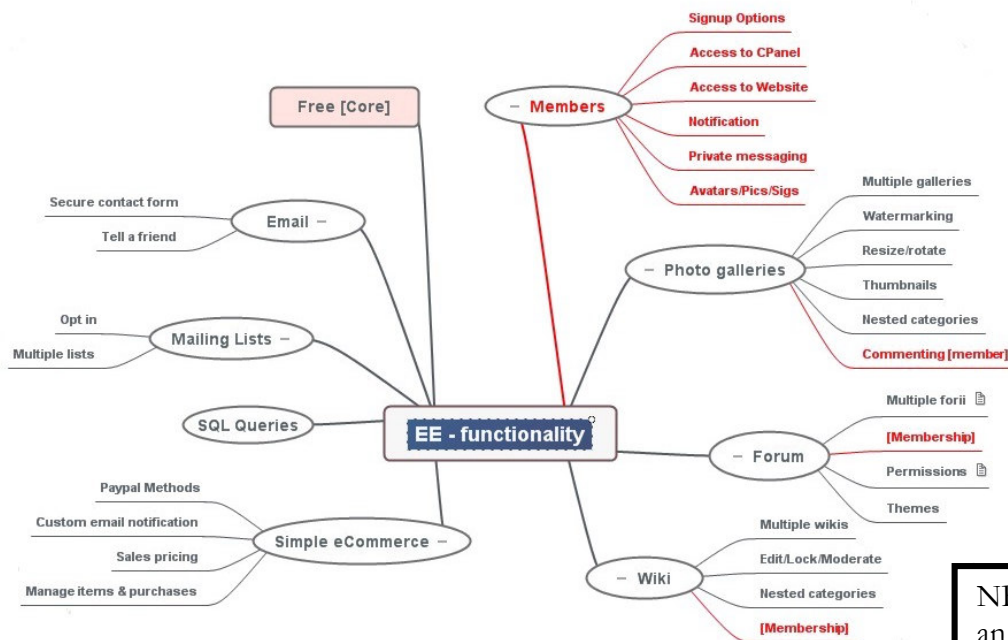
**Themes** – well, they do exist, but just aren't essential. When you first install EE, you will be asked to select a theme. You can use it, but personally, I have just ignored it. It is so easy to cut & paste in your own HTML/CSS, and make that define the site layout.

**Search** – built in, ready to customise templates. Simple or advanced options for more control.

**Backup**: one step backup of entire database [this includes everything – templates, entries, permissions etc] only things are external files such as graphics, MP3's and Flash.

## 4.2 EE Full versions

The full version offers a lot of functions that in other progs requires integrating third party stuff, with all the problems of incompatibilities, memberships, style differences.



NB The Forum is an addon module – \$50 or \$100

The full version contains a LOT of extra modules which offer significant functionality.

**Membership:** Allows you to set up multiple member groups. Each member account requires a unique name and unique email address. Each member account must belong to one member group. There are all sorts of options for controlling signup [captcha's, activation etc].

### 1) Limiting access to control panel

- different member groups can be restricted to what they can do in the CP
- different member groups can be restricted to what they can even see in the CP.

Very useful for beginners you can so simplify the CP, that they cannot even see the template, membership or admin tabs where they could really do damage. You can limit them to which weblogs they can see, and even restrict them to only editing their own entries.

You can set up custom status flags for the entries, so low level users can only submit articles, and require a higher level user to authorise before being published – a proper workflow regime.

2) Limiting access to final website content

you can set limit to specific pages for different member groups – access is on a per template basis [fine grained]

**Photo Galleries:** create and display multiple galleries, with thumbnails, batch upload, commenting, watermarking, resizing and rotating.

**Forum** [separately purchased module]

**Wiki** [multiple wiki's]

**Simple eCommerce** [PayPal only]

**SQL queries:** allows you to query the database directly [not update/delete]. Obviously need knowledge of dbase structure, but that is well documented

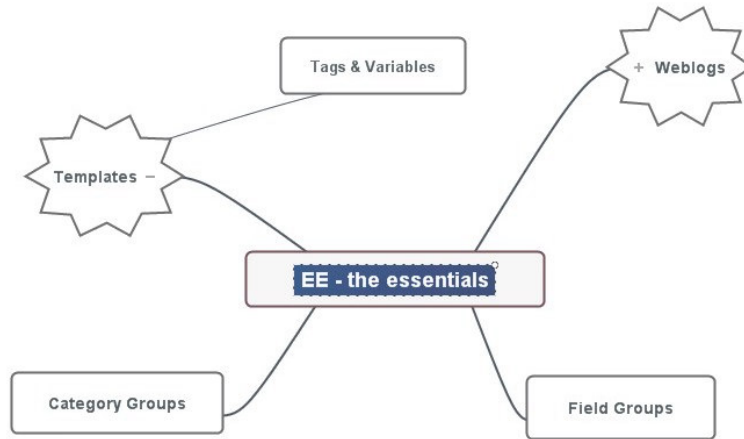
**Mailing List:** multiple lists, opt in variety [user types email addr and is then emailed link to follow]

**Email** – secure contact forms [your email address not visible to spammers]



## 5. EE – the fundamentals

We now leave general descriptions and talk about the real heart of EE:



In particular, we will be talking about:

- Templates
- Weblogs
- Field groups and custom fields
- Categories

And how they are tied together with Tags and Variables of different kinds.

Along the way we will cover embedding templates [like SSI but better], search engine friendly URLs, embed variables and the magic behind templates [i.e. `segment_3` and its friends]

## 5.1 Templates: an introduction

Basically a template is an HTML page<sup>5</sup>. It can also be a CSS stylesheet or Javascript file or even an XML or RSS page.

Each templates lives in a template group – for ease of organisation. So your individual ‘product’ pages could all live in a ‘product\_range’ group. You can have as many groups as you want, and as many templates in each group. .

Obviously you could build the entire site in templates, just as a collection of static pages, but that’s rather missing the point of a CMS! So we will start to show how to use templates to your advantage.

First though, each template group has an ‘index’ template – this is the default template for the group – which can be important when we will see how a template name is related to its url.

### 5.1.1 Templates and site urls

The default url for an EE site is:

```
www.mysite.com/index.php6.
```

If you want to view a particular template in your browser, then you simply append the template group name and the template name., i.e:

```
www.mysite.com/index.php/product_range/item_a
```

If you omitted the final segment of the address, e.g:

```
www.mysite.com/index.php/product_range
```

then you are shown to the default [index] page for that template group<sup>7</sup>.

Note that urls contain no obscure characters – nice and human/search engine friendly.

### 5.1.2 Editing templates

You can edit templates in the CP. Alternatively it is easy to edit them in you favourite HTML/CSS editor with all those nice validation and other features, and then FTP the templates to prescribed folders on the server [these folders will have same names as template groups].

### 5.1.3 Comments

Commenting is simple - EE comments won’t appear at all in final rendered page.

```
{!-- this is a comment - can be multi line --}
```

Similar to an HTML comment, except using curly braces {...}. which is main reason for mentioning them. If you see { } in a template, then inside it is something EE – either a comment, an embed, a tag or a variable.

---

<sup>5</sup> We will see how this definition expands, later on [section 5.1.4], but this will do for now.

<sup>6</sup> There are documented ways, using the .htaccess file, of suppressing the ‘index/php’ bit of the url.

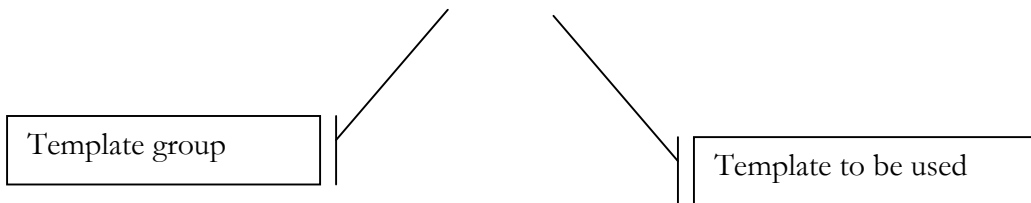
<sup>7</sup> Alternatively, you can select a default template to be displayed in case of an incomplete url [for future reference this is in CP: Templates: Global Template Preferences: 404 page.]

### 5.1.4 Embed templates: basics

This is where we finally start to do something useful. So far we have defined a template as being an HTML page. Actually it can be also part of a page, just like a Server Side Include [SSI].

So you can build up a page from various snippets of standard stuff, with just the different text actually in the master template, using the following syntax:

```
{embed="my_embeds/nav_main"}
```



As a personal preference I keep my standard site wide embeds (templates which I embed), in clearly named template groups – such as ‘embeds\_in\_head’ and ‘embeds\_in\_body’.

By contrast, specific embeds that will only be used by one template or those in one template group, I keep in the same template group. However that organisation choice is entirely up to you and your style.

Here is a template that generates a simple HTML page, using a number of embedded templates to do the heavy lifting:

```
{embed="embeds_in_head/doctype" }
<head>
  <title>My Site</title>
  {embed="embeds_in_head/meta_tags" }
  {embed="embeds_in_head/links_css" }
  {embed="embeds_in_head/links_favicon" }
</head>

<body id="pid_literature">
  {embed="embeds_in_body/nav_main" }

  <h1>What the Dickens</h1>
  <p>It is a far, far, better thing...</p>

  {embed="embeds_in_body/content_sec" }
  {embed="embeds_in_body/footer" }
</body>
</html>
```

Note how clearly this reads. Of course, you can embed templates within templates within templates within... but do be sensible!

## 5.2 Weblogs

So far all the text has been embedded in a template, but that still defeats the idea of a CMS. You would still need, for example, a separate template for each product.

What we want to do is have a generic template, tell it which product to display, and then let the database pull in the appropriate data for that product. Or, if you are doing something more blog based, we might want to get the template to display the summaries of the 10 most entries, etc etc.

And this is where the weblog comes in. Weblogs hold the data

### 5.2.1 Weblog is not a web-log

**First of all, whenever you come across the word ‘weblog’ in EE documentation of forums, DON’T think BLOG, just think ‘data container’. This knowledge is absolutely crucial to the correct understanding of EE.**

More exactly, an EE weblog is a table in the database sense. So an ‘entry’ in the weblog, is a row in the table. Even better, you can determine the table structure.

Now an entry in an EE weblog, might indeed be for a blog entry, but it might also be the technical data for a product - that choice is yours.

You can define as many weblogs [tables] as you want, with unlimited entries.

### 5.2.2 Field groups define weblog structure

**This is one of the best parts of EE.** You can completely define the structure [columns] in an EE Weblog. You do this by setting up a **field group**. A field group consists of one or more fields, over which you have a lot of control.

You can define a fields name, order in the group [i.e. effectively column number] and type, among other things. Field types include:

- Text input [number of characters]
- Text area [number of lines of text]
- Drop down list
- Date
- Related field [links to another weblog of the programmers choice]. The user, when creating/editing an entry, can then can select any single entry in that weblog.

With the first two types, you can allow EE to add the necessary xhtml [e.g. <p>...</p> around paragraphs] or you can elect for no formatting.

With this flexibility you can set up fields that are appropriate for different tasks, and not have to shoehorn everything into a few predefined slots. Thus in my product weblog [table], the field group has over sixty fields, to handle and allow individual access to specific items of data. By contrast, the ‘news’ weblog only required 3 or 4 fields.

**One field group can be used to define the structure of many weblogs  
But each weblog is completely defined by one field group only!**

### 5.3 Categories

Once weblog entries have been created, you will almost certainly want to categorise them in different ways. Well, you just set up category groups. Each group consists of one or more categories, which can be nested as deeply as you want.

You define the name for each category item, and its level in the group, EE will assign it a unique ID.

Once the category groups have been created, you can assign one **or more**<sup>8</sup> category groups to a weblog. Then when creating or editing an weblog entry, you then assign one or more categories from any of these groups to the entry.

You can [if wanted] then use these categories to elect which entries are displayed, but that is getting a bit ahead of ourselves.

---

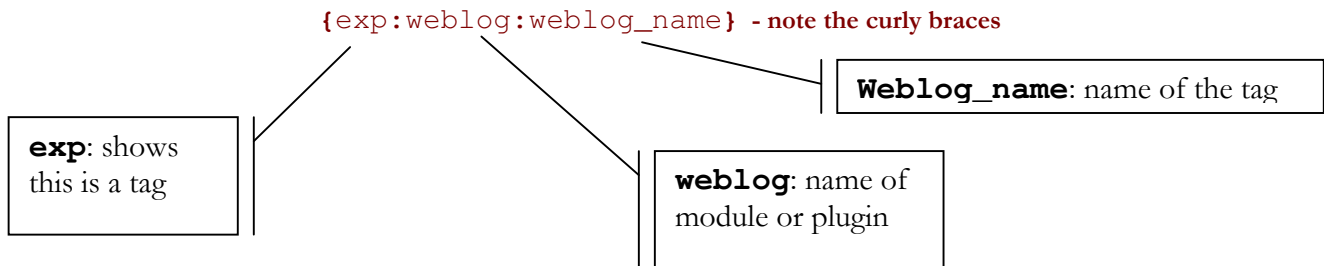
<sup>8</sup> Until v1.5 you could only assign 1 category group to a weblog – which was very frustrating. Now that limitation has been removed.

## 5.4 Templates & Weblogs – joining the two

So – how do we get a template to display stuff from a weblog!! That is the \$64,000 question. And the answer is, very easily, using EE Tags.

### 5.4.1 EE Tags

An EE tag is an instruction to EE to do something. Each module in EE and each plugin has its own tags. All tags have the same format, so it's dead easy to spot them! Here is one of the simplest:



This is a single tag, which just returns the title of the current weblog. Many tags come in pairs, which surround some other code. One of the most useful is also from the weblog module.

```
<body>
<h1>My Products</h1>
{exp:weblog:entries}           << start of tag
  <p>this is an entry</p>
{/exp:weblog:entries}        << end of tag
</body>
```

This loops through all entries in all weblogs, spitting out the html for the line 'this is an entry' – not very useful after all? However we can control this tag using tag parameters and determine what it pulls out using the custom field names and variables.

```
{exp:weblog:entries weblog="weblog_a" category="C11|C15" sort="asc" }
  <h2>{title}</h2>
  <p>{f_para1}</p>
{/exp:weblog:entries}
```

parameters

This time we loop through weblog\_a only, finding all entries in categories C11 or C15, and generate the html that displays the title as an h2 header and then whatever is in custom field f\_para1 as a paragraph. All of this is sorted with most recent entry first. Simple huh!

There are all sorts of parameters that can be employed to get the data that you want. [*see handout of EE tags*] – just don't have the time to go through these.

You can also use a variety of variables between the tags. We have used custom field names, but there are also some predefined items create specific info. One useful one is the {title\_permalink} variable. It is quite likely that you will want to create a link to display the full entry.

```
{exp:weblog:entries weblog="weblog_a" category="C11|C15" sort="asc"}
  <h2>{title}</h2>
  <p>{f_para1}</p>
  <a href="{title_permalink="myproducts/generic_product"}">
    read more...</a>
{/exp:weblog:entries}
```

This will then generate a nice link for each specific entry – without you needing to specify the entry name manually.

If anyone chooses to click on one of these links, then EE will display that specific weblog entry using the ‘generic\_products’ template from the ‘myproducts’ template group.

Thus in 5 lines we have the basis of a blog ☺. There are lots of other tags, but no time to go into them now. [ [http://expressionengine.com/docs/quick\\_reference.html](http://expressionengine.com/docs/quick_reference.html) ]

## 5.5 EE Variables

EE variables are useful little things that dramatically increase flexibility and maintainability of your code.

The first thing to understand is that EE variables aren't really variable, or at least they are read-only! So you can access them, but not alter them. Even variables you define won't change value. Some even act as little functions, transforming data.

### 5.5.1 Tag specific variables

We have already come across tag specific variables – these are specific to a particular tag pair, and can only live/work within that tag. See the quick reference chart [ [http://expressionengine.com/docs/quick\\_reference.html](http://expressionengine.com/docs/quick_reference.html) ]

### 5.5.2 EE variables

EE provides a number of types of 'variables', which do quite different things:

#### 5.5.2.1 Standard variables – [for information]

Provide information about some current operation/setting of the system or users, e.g.:

- {elapsed\_time} – time taken to render a specific template for this particular view
- {username} – name of current logged in user

#### 5.5.2.2 Path variables - [for portability!]

To my way of thinking these are really functions, but whatever they are, they are really useful! **Use them right from the start – they will ensure that your site is easily portable!**

Path variables transform 'template\_group/template' information into full rendered path, so if we simply write:

```
{path="products/product_a"}
```

this is transformed, by EE, in the final rendered page to:

```
http://www.mysite.com/index.php/products/product\_a
```

The obvious use is in a hyperlink:

```
<a href="{path="template_group/template"}">...</a>
```

These are therefore similar to the {title\_permalink} variables we have seen before, except path variables don't need to be used inside {exp:weblog:entries} tag. Makes it really easy to create a site at one address and move it to another, because they pick up the current site address from one central location.

There is a special version of this for style sheets.

```
<link rel="stylesheet" ...  
href="{stylesheet=stylesheets/mycss}" />
```

If you use this EE makes no attempt to look for PHP in the template, and other stuff that is needful for stylesheets.



NB it does not seem to matter whether you ‘quote’ the template\_group/template structure or not.

### 5.5.3 Dynamically assigned global variables [for reusability]

**These are user defined variables.** These only apply within the template that there were defined in. Why they are called global I’m not sure!

You can define a variable in a template [generally best to define them at the beginning], using the `assign_variable` format:

```
{assign_variable: my_weblog="weblog_a"}
```

You can then use these else where in code in the template. Makes it so much easy to maintain or to reuse templates, e.g:

```
{exp:weblog:entries weblog="{my_weblog}"
category="C11|C15" sort="asc"}
```

Note that DAGV’s have to be encased in curly brackets to let EE know that you are using a variable, and not just a string with that value.

I repeat: ‘Dynamically assigned global variables’ **only** apply within the template they were defined for – they do NOT apply to any embedded templates or vice versa!

### 5.5.4 User defined global variables [for ease of maintenance]

These really are globally defined variables [or constants anyway ☺]! The can hold any static information, e.g. names, telephone nos, addresses or even HTML in them.

Don’t try and put dynamic stuff in them, e.g. PHP or EE Tags, as they won’t be run. Sadly you can’t use them as parameters in tags, as they are one of the last things parsed.

### 5.5.5 Conditional variables [for power]

Very useful – you can test other variables or custom fields and act accordingly: Same format as PHP conditionals – here is the standard ‘if’ structure, testing a custom field.

```
{if "{f_paral}" != ""}
  {f_paral}
{/if}
```

here is the ‘if/else’ structure

```
{if "{f_paral}" != ""}
  {f_paral}
{if:else}
  <p>it was the worst of times</p>
{/if}
```

Notes:

- the ‘{if:’ prefix to the ‘else’ – EE requires it to identify the control structure
- {if:elseif is also available
- Standard operators ==, !=, <, <=, > & >= available
- Can use Boolean operators to make advanced conditionals

## 5.6 Extending your templates

So far we have looked at the template as holding the HTML for some or all of a webpage, or a style sheet or JavaScript file. We have seen how we get nice simple urls from EE, via the ‘template\_group/template’ naming scheme. We have seen how to use EE tags to pull weblog entries [or parts of] into the final page.

All this is quite obvious – you could see the code and make a good guess as to what it does. However EE templates also have some none obvious stuff. So lets see some of the deeper magic in EE templates, and how to use it to your benefit.

### 5.6.1 Single entry pages and urls

Earlier we saw the `{title_permalink}` variable generated the path to display a specific entry in a template. The title or entry id<sup>9</sup> will appear in segment 3 of the url, e.g:

```
www.mysite.com/info.php/myproducts/any_product/filter311
```



Note how nice and readable the url still is and how search engine friendly.

Even more cunning, is the fact that when EE parses the url, it will recognise that segment three contains an entry title or id. will then supply info from that entry<sup>10</sup> only.

Therefore if the template uses any EE tags, you don’t need to specifically name the entry – EE holds it in mind. Such a template is known as a single entry page. In fact, some tags only work on single entry pages – for example the ‘next entry’ and ‘previous entry tags:

```
{exp:weblog:prev_entry ...}
{exp:weblog:next_entry ...}
```

NB a page which shows many entries is cunningly known as a multi-entry page!

### 5.6.2 Multy-entry pages and urls

Segment 3 can hold other things than entry titles. For example, if you call a multi-entry page, and segment 3 contains a category ID, EE will recognise that.

Thus if we generate<sup>11</sup> a list of links to display distributors in a specific part of the world [each region has its own category]:

```
{exp:weblog:categories weblog="{master_weblog_name}"
style="linear"}
<p><a href="{path=commerce/distributors}">
  {category_description}</a>{/exp:weblog:categories}
```

<sup>9</sup> We would get the entry id if we had used the `{permalink}` variable instead.

<sup>10</sup> By using the ‘dynamic=’off’ parameter in an EE tag, on a single entry page, you can force EE to look at all the other entries.

<sup>11</sup> Note the use of the ‘categories’ tag to generate the list of links.

on the rendered page this gives a list:

```
<a href="/.../commerce/distributors/C1">Europe</a>
<a href="/.../commerce/distributors/C2">U.S.A.</a> etc
```

If you click one of these links, EE recognises a category ID in seg 3, and will render multi-entry distributors page showing only distributors from Europe or USA etc, **without need** to specify this in any tag.

There is yet more magic - if segment 3 is neither an entry identifier or a category ID, then EE knows this. The segment can be used by you as a variable {segment\_3} inside the template. In fact you can use up to 9 segment variables.

### 5.6.3 Template embed variables

So far embedded templates have been used for invariant HTML, even if spiced up with EE, i.e. they will be the same no matter where embedded.

However you can pass ‘variables’ to templates. These are known as ‘embed variables’ for obvious reasons.

Don’t be fooled though – they pass by value, and as with any other EE ‘variable’ can’t be changed inside the embedded template. But, given that caveat, are very useful. Let’s see how it can be used:

Parent template:

```
{embed="embeds_in_body/header_h1" h1title="small product"}
```

Child template:

```
{if "{embed:h1title}" != ""}
  <h1>{embed:h1title}</h1>
{if:else}
  <h1>product</h1>
{/if}
```

This will generate an h1 header saying ‘small product’ if the variable is named in the parent [not necessary].

I could, in the parent template, have used a custom field rather than a simple string.

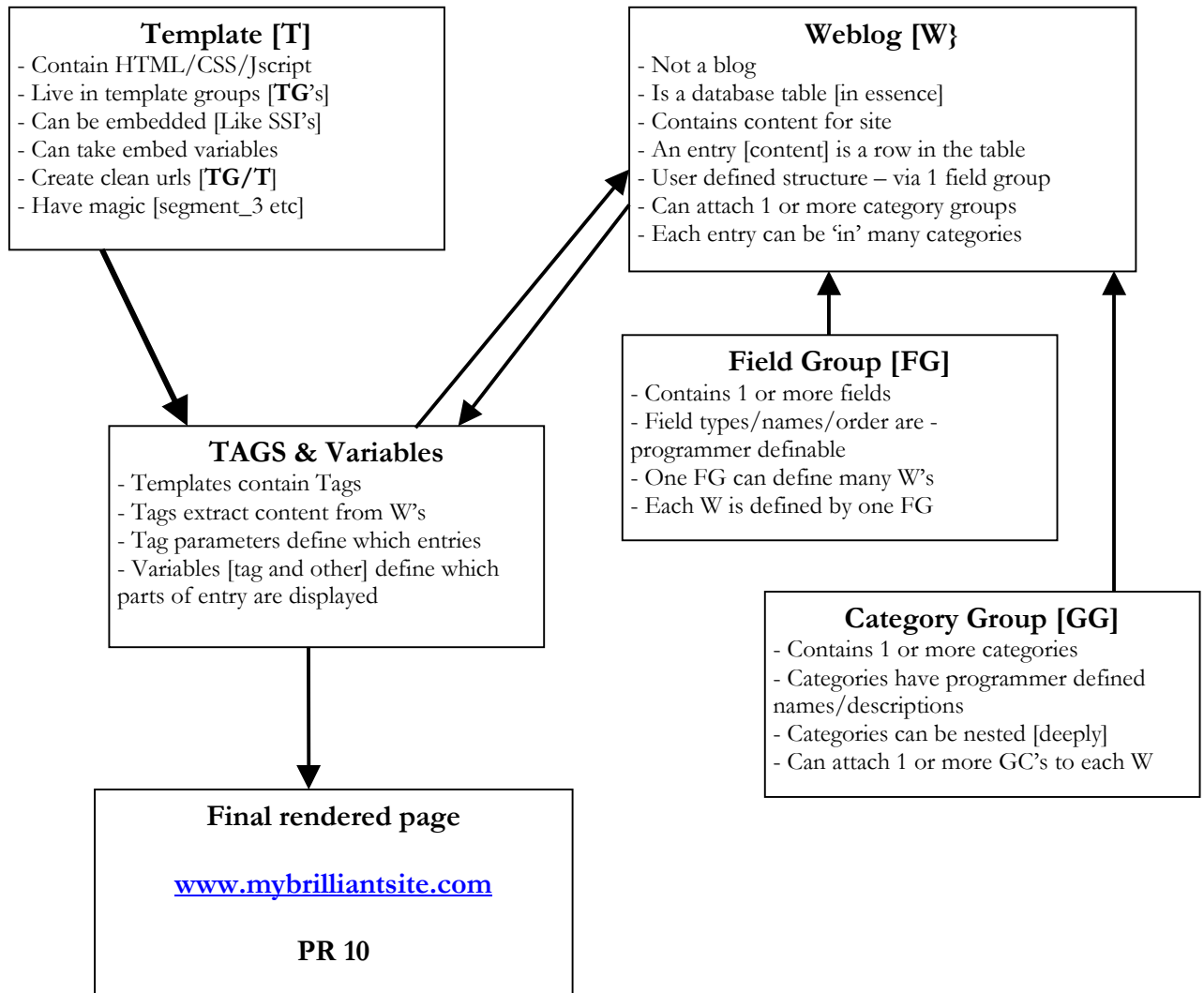
You can have as many embed variables as you want. Order is not important. They are not explicitly declared in the child template.

### 5.6.4 PHP in templates

You can use PHP inside a template. It does need to be turned on [by default, it is off – for security]. You don’t need to enable PHP parsing for parent or child templates.

You can use PHP to generate EE. I’ve used this a little and it rather cracked me up, to think I was using PHP to write EE which generates PHP – somewhat recursive!

**Bringing it all together:**



**6. Summary**

There is so much more I could cover – Status groups – for example. But time does not permit, so I will just finish now with a summary of my impressions about EE

- It is 'the Designers Friend' – works extremely well with standards compliant HTML/CSS
- Provides a blank sheet – doesn't force you into any design mould
- Don't need to be a PHP Guru to use it – but those features are there if needed
- Powerful, flexible & secure
- Easy to backup
- Commercial product with outstanding support
- Does have a learning curve, but it's not overwhelming
- Do plan your fields, categories, weblogs & templates when doing a serious site

**Don't forget 'Weblogs aren't web-logs' and 'variables aren't variable' ☺**

## 7. Appendix: EE Resources

Some useful EE related links:

<b>Main EllisLab places</b>	<b>url</b>
EllisLab site	<a href="http://www.ellislab.com">www.ellislab.com</a>
EllisLab webhosting [sister company]	<a href="http://www.enginehosting.com">www.enginehosting.com</a>
Knowledge base	<a href="http://expressionengine.com/knowledge_base">http://expressionengine.com/knowledge_base</a>
Forums	<a href="http://expressionengine.com/forums">http://expressionengine.com/forums</a>
On line docs	<a href="http://expressionengine.com/docs">http://expressionengine.com/docs</a>
On line Wiki	<a href="http://expressionengine.com/wiki">http://expressionengine.com/wiki</a>
Plugin List	<a href="http://expressionengine.com/downloads/addons/category/plugins">http://expressionengine.com/downloads/addons/category/plugins</a>
Module list	<a href="http://expressionengine.com/downloads/addons/category/modules">http://expressionengine.com/downloads/addons/category/modules</a>
Extensions list	<a href="http://expressionengine.com/downloads/addons/category/extensions">http://expressionengine.com/downloads/addons/category/extensions</a>
Development blog	<a href="http://expressionengine.com/blog">http://expressionengine.com/blog</a>
<b>References</b>	
EE Quick reference chart	<a href="http://expressionengine.com/docs/quick_reference.html">http://expressionengine.com/docs/quick_reference.html</a>
Jamboree [EE celebration]	<a href="http://www.jambor-ee.com">www.jambor-ee.com</a>
<b>Example Sites</b>	
pMachine \$15000 shootout	<a href="http://www.pmachine.com/shootout">www.pmachine.com/shootout</a> [can't find this on new site]
EE showcase	<a href="http://expressionengine.com/wiki/Powered_by_ExpressionEngine">http://expressionengine.com/wiki/Powered_by_ExpressionEngine</a>
Veerles blog	<a href="http://veerle.duoh.com">http://veerle.duoh.com</a>
Erskine Corp	<a href="http://www.erskinecorp.com">www.erskinecorp.com</a>
Buxton Opera House	<a href="http://www.buxtonoperahouse.org.uk">www.buxtonoperahouse.org.uk</a>
Methods arts	<a href="http://www.methodarts.com">http://www.methodarts.com</a>
My first EE site – because it introduced me to EE and was used to illustrate this talk	<a href="http://www.magnum.com">www.magnum.com</a>

If you want to see some comparisons between various CMS's [wordpress, drupal, joomla, drupal] then a starting point might be the following forum thread:

<http://www.expressionengine.com/forums/viewthread/38244/>

Of course this is written by EE enthusiasts, but they do seem to have used the others in some depth [perhaps why they are enthusiasts!].

There is also an interesting interview with the people in the EE technical team at:

<http://www.jambor-ee.com/features/entry/questions-for-pmachine/#comments>

It includes stuff on some future things and also their favourite elements in EE.

## 8. Appendix: template and tags example

This is a simple example showing code for two templates, one embedded within the other using embed variables. In the parent template we use an EE tag to generate a number of entries, and include permalinks and pagination. We also use conditional variables, user defined variables & EE comments.

I have excluded lots of stuff, such as the doctype, meta tags, title etc, from the HTML code that I would normally use! I do however include a style sheet link to show the EE way.

### ----- master template -----

```
{assign_variable: master_weblog="w_weblog_products"} {!-- weblog used in this template --}
{assign_variable: summary_entry="products_summary"} {!-- weblog entry for summary text --}
{assign_variable: template_single_entry="info/product"} {!-- single entry template --}

<html>
<head>
  {!-- CSS is held in a template, so note use of template_group/template name form --}
  <link rel="stylesheet" type="text/css" media="screen" href="{stylesheet=stylesheets/mycss}" />
</head>

<body>
  {!-- Using an embedded template to generate header – extracting text from a named entry
  in the weblog – passing names of weblog and entry as embed variables --}
  {embed="embeds_in_body/header" w1blog="{master_weblog}" w1entry="{summary_entry}"}

  {!-- now using an EE tag pair to display a number of entries [excluding the summary entry], with
  pagination top and bottom, we show 4 entries at a time --}
  {exp:weblog:entries weblog="{master_weblog}" url_title="not {summary_entry}" limit="4" paginate="both" }

  <h2>{title}</h2> {!-- using entry title for the h2 headers --}

  <p>{f_product_summary}</p> {!-- summary paragraph for this product --}

  {!-- permalink to entry: note use of user defined variable {template_single_entry} to
  provide the necessary template_group/template info: could have used that directly --}
  <p class="permalink">
    <a href="{title_permalink="{template_single_entry}"}" >read more</a>
  </p>

  {!-- pagination code – can be anywhere inside the weblog:entries tag pair
  note the EE variables {current_page} etc, that apply inside the pagination markers --}
  {paginate}
  <p class="paginate">Page {current_page} of {total_pages} pages {pagination_links}</p>
  {/paginate}
  {/exp:weblog:entries}

</body>
</html>
```

*[embedded template on next page]*

----- header template [embedded in parent] -----

```
{assign_variable: h1_default="Good Stuff"}

<div id="header">

{!-- call up one named entry from named weblog: these have been passed as embed variables --}
{exp:weblog:entries weblog="{embed:w1blog}" url_title="{embed:w1entry}"}
  <h1>
    {!-- if selected field has contents then use that as header, otherwise use the default string --}
    {if "{f_long_title}" != ""}
      {f_long_title}
    {if:else}
      {h1_default}
    {/if}
  </h1>

{/exp:weblog:entries}
</div>
```

## 9. Appendix – Magnom product notes

[a few notes about the product which required me to hunt for a content management system in the first place]

### 9.1 contaminants in industrial fluids

Industry runs on fluids of various kinds – water, oil, fuel, hydraulic, transmission, cutting & cleaning fluids amongst others. These fluids can be contaminated with various kinds of particles; of which ferrous [iron/steel] particles are the worst for causing damage.

Suprisingly, it is the smaller particles [1-50microns] that are most damaging, especially in especially in modern close tolerance machines. These particles are just the size to be able to get into the clearances between moving parts, then get stuck and ground along like abrasives – gouging out more of the same [chain reaction of wear] – leading to eventual massive failure. If your car engine has ever seized up, then this may well be the reason [apart from an oil leak of course!]

The obvious solution is to use conventional barrier [paper] filters. But these are either are either too coarse, or if fine enough clog quickly or restrict the flow rate significantly. Barrier filters can split, dumping huge amounts of these materials into the key components, and are no use in high vibration environments or where the flow is pulsing or can even reverse direction. Finally they can strip out additives that have been put in the fluid to enhance performance!

One solution is to place a magnet into the fluid. However, this is not enough, as the accumulated particles are still in the main fluid flow, so as material accumulates, the outer layers are prone to ‘wash off’, where clumps are forced off by the fluid pressure.

### 9.2 Magnom filter

The Magnom™ filter is a ring shaped magnetic filter which has carefully shaped field plates at either end. The plates have a number of clear channels round the circumference. The fluid passes unrestricted through the channels [the central hole is used for a mandrel to mount a chain of these units in series]. The magnetic fields lines however pull the particles into ‘dead-waters’ between the plates, where they happily accumulate without being subject to wash off.

The Magnom is very effective & is used to protect equipment in:

- a large range of industries [e.g. from huge industrial coal crushing plants to F1 racing]
- clean a wide range of fluids [water, oil, fuel, hydraulics, cutting fluids etc]
- wide range of operating environments [from ‘fit and forget’ solutions in hydraulics to systems needing daily changing, such as metal work shops].

### 9.3 Website goals

We had to make sense of all the different products, industries and applications. I realised that I had to provide lots of different ways in to access the appropriate information:



- distributors would want to look up a particular products
- industry consultants seeing how they might they could use Magnom
- engineers looking for a solution to an urgent problem with a specific fluid
- others would be interested in real life applications
- people wanting to understand the particle problem and how the Magnom works

So rather than just set up a list of products, we wanted several high level types of page:

- Product page – full data table + links to relevant industry, fluid and case studies
- Industry page – notes on that industry explaining how Magnom could be used + links to appropriate products and case studies
- Fluid page – notes on that fluid explaining how Magnom could be used + links to appropriate products and case studies

So that users could come into the site with whatever was of concern to them, see how Magnom could help, and then be lead quickly to the best product for their task.

Plus, of course, case study pages [with links to product, industry and application], lists of distributors, technical notes, email signup etc etc