SMARTY USES AND XOOPS

1. What is Smarty?

Not to re-invent the wheel, the following is what the author has to say about Smarty:

> "Although Smarty is known as a "Template Engine", it would be more accurately described as a "Template/Presentation Framework." That is, it provides the programmer and template designer with a wealth of tools to automate tasks commonly dealt with at the presentation layer of an application. I stress the word *Framework* because Smarty is not a simple tag-replacing template engine. Although it can be used for such a simple purpose, its focus is on quick and painless development and deployment of your application, while maintaining high-performance, scalability, security and future growth." - smarty.php.net

Some of the notable features of Smarty (from smarty.php.net):

Caching: Smarty provides fine-grained caching features for caching all or parts of a rendered web page, or leaving parts uncached.

Configuration Files: Smarty can assign variables pulled from configuration files.

Variable Modifiers: The content of assigned variables can easily be adjusted at display-time with modifiers.

Template Functions: Many functions are available to the template designer to handle tasks such as generating HTML code segments.

Plugins: Almost every aspect of Smarty is controlled through the use of plugins. They are generally as easy as dropping them into the plugin directory and then mentioning them in the template or using them in the application code.

Debugging: Smarty comes with a built-in debugging console so the template designer can see all of the assigned variables and the programmer can investigate template-rendering speeds.

2. Xoops Smarty

Xoops adopted Smarty in Xoops2. With the adoption of Smarty, Xoops also saw the branching-out of e-Xoops. Some of Xoops users decided against using smarty and continued the development of Xoops1 and renamed it as e-Xoops.

Xoops' use of Smarty tags is a slightly different. Xoops uses bracket **<>** to enclose smarty tag. So instead of **{$name}**, Xoops uses **<{$name}>**

Xoops Pre-Set Theme/Template Smarty Tags

<{$xoops_charset}>
Inserts the characterset info (meta tag info) e.g "iso-8859-1", "UTF-8"

<{$xoops_langcode}>
Content Language e.g. "DE", "EN"

<{$xoops_meta_keywords}>
Inserts the keyword list from the Meta/Footer settings (meta tag info)

<{$xoops_meta_description}>
Inserts the meta tag site description

<{$meta_copyright}>
Inserts the meta tag copyright text

<{$meta_robots}>
Inserts the W3C robot meta tag info

<{$meta_rating}>
Inserts the meta tag rating information

<{$xoops_sitename}>
Inserts the site name

<{$xoops_pagetitle}>
Inserts the page title

<{$xoops_theme}>
Theme's name in directory "/themes/"
e.g "default", "xt2"

<{$xoops_themecss}>
Inserts the style.css
e.g "http://www.xoops.org/themes/default/style.css"

(note, any combination of the above can be arranged to
fill out meta tag info to your liking)

<{$xoops_url}>
Inserts the site URL
e.g "http://www.xoops.org"

<{$xoops_js}>
Inserts XOOPS javascript code

<{$xoops_banner}>
Displays banners

<{foreach item=block from=$xoops_lblocks}>
<{$block.title}>
<{$block.content}>
<{/foreach}>
Code snippet to display the left blocks

<{if $xoops_showcblock == 1}>
Tests for system settings to display center blocks

<{foreach item=block from=$xoops_ccblocks}>
<{$block.title}>
<{$block.content}>
<{/foreach}>
Code snippet to show the center-center blocks

<{foreach item=block from=$xoops_clblocks}>
<{$block.title}>
<{$block.content}>
<{/foreach}>
Code snippet to show the center-left blocks

<{foreach item=block from=$xoops_crblocks}>
<{$block.title}>
<{$block.content}>
<{/foreach}>
Code snippet to show the center-right blocks

<{$xoops_contents}>
Show the news and other contents

<{if $xoops_showcblock == 1}>
Test for showing center blocks

<{if $xoops_showrblock == 1}>
Test for showing right blocks

<{foreach item=block from=$xoops_rblocks}>
<{$block}>
<{/foreach}>
Code snippet to show right blocks

<{$xoops_footer}>
Shows the footer

<{$xoops_isadmin}>
True if the visitor is Administrator

<{$xoops_isuser}>
True if the visitor is a logged in user

<{$xoops_userid}>

User ID of the member

<{$xoops_uname}>
Uname for the member

The uses of smarty with Xoops are numerous. We will not attempt to summarize every single use but will cover a few that we believe it is simple to do but offers vast arrays of uses in extending the functions of Xoops. It is our hope that you could use the same logic and most importantly your imagination to fine-tune your own Xoops application.

WARNING: Please backup your files before you attempt to make any changes, so you can reverse back to your original file in case of the tips are not working for some reasons. Although we have tested the tips, we do not guarantee that the following tips will work for you.

3. Run-timer Modifier and Xoops Use

One of the Smarty features mentioned previously is that the content of assigned smarty variables can be adjusted at display-time. This option gives you the ultimate control on where and how you want to display your contents.

Block fine-tuning
Xoops block displays are fixed with modules being assigned to. If you have no starting module, the block assigned in the top page theoretically are also associated with a module which you may call it “Top Page” module. This is just a clarification if someone objects to the block-module association. Xoops pulls the association information from block_module_link table.

If you want to assign, let us say, a right block to a module and want it only displayed in the module top page – the index page. Without modification, Xoops under current settings will display the right block to every pages of the module.

You could overwrite the Smarty variable controlling the showing of any right blocks.

In the default theme, Xoops checks if the Smarty variable **<{$xoops_showrblock}>** is set to “1” in order to render the right blocks. Knowing this, you could overwrite the Smarty variable value to “0” to prevent the right blocks being shown on certain pages of the module. [**It may not work if your page is cached**]

Add the following to the page that you don’t want the right blocks to show.
$xoopsTpl->assign('xoops_showrblock', 0);
before
include XOOPS_ROOT_PATH.'/footer.php';

By doing this, you effectively prevent the display of right blocks on this particular page.

Page Title <{$xoops_pagetitle}>
Page title plays a very important role in searching engine ranking. The stock Xoops distribution by default assigns the module name as the page title for every pages of a particular module. This definitely will not help you improve your ranking. Google and other search engines rely heavily on page titles in page ranking all other factors considered. A page from PR5 site may rank well ahead of a similar page from a PR7 site due to the simple fact that the page from PR5 has a page title with all relevant "key words" while the PR7 page does not.

There are several hacks of implementing page titles. It may sound a daunting task in assigning your own page title. It is really not complicated if you understand the working of Smarty.

You can just simply overwrite the Smarty variable at display-time with your own page title.

For example, if we want to display an individual download name as the page title under singlefile.php.

We know from reading the file that $dtitle is assigned as the title of an individual download name. Knowing this, now we can assign it as page title.

Add
$xoopsTpl->assign('xoops_pagetitle', $dtitle);
before
include XOOPS_ROOT_PATH.'/include/comment_view.php';
include XOOPS_ROOT_PATH.'/footer.php';

We now have a unique page title for every download. Very simple? It is and should be.

Meta keywords
The value of meta keywords in searching engine rankings is highly debatable. Google pays no attention to meta keywords. Yahoo! and MSN may or may not assign weight to keywords. Even they do, the weight may be much less than those of your page titles.

Although it may not help in most cases, there is no harm in implementing custom keywords. In case that you want to implement custom keywords for your pages, the followings are steps that you could accomplish the feature.

It will require modifications to theme.html file and individual pages.

Let's say we also want to include the download name in the metakeywords. What we want is simply "add" the download name into the Xoops metakeywords. We will assign a Smarty variable for the download name and add as part of theme.html file.

Here are the steps:

Add (singlefile.php)
$xoopsTpl->assign('xoops_customkeywords', $dtitle);
before
include XOOPS_ROOT_PATH.'/include/comment_view.php';
include XOOPS_ROOT_PATH.'/footer.php';

Now notice that we have just created a Smarty variable called **xoops_customkeywords** with a value of $dtitle.

Next, Add **BOLD** (in theme.html)

<{if $xoops_customkeywords != ' '}>
<meta name="keywords" content="<{$xoops_customkeywords}>,
<{$xoops_meta_keywords}>" />
<{else}>
<meta name="keywords" content="<{$xoops_meta_keywords}>" />
<{/if} >

EXTRA ATTENTION:
<{if $xoops_customkeywords != **' '**}>
Notice the use of two single quotes, not double quotes. Otherwise you might have blank page problems.

This tells Xoops to check if **xoops_customkeywords** Smarty variable has been assigned. If it is, then include it in addition to the default keywords. If not, use the default keywords.

Now you have customized keywords for every pages of your site.

As the name implies, it is customizable so you decide whatever information you wish to include in the keywords.

You could simply use the xoops_pagetitle smarty variable instead of creating a new variable called **xoops_customkeywords.** But in some cases you do not want to include pagetitle as part of your keywords because pagetitle is usually too long. The meaning of “keywords” tends to get lost with a very long pagetitle.

4. Use of Right Blocks

Default Xoops theme block displays are as follows

LeftBlocks **CenterCenter** **RightBlocks**
CenterLeft **CenterRight**
Contents

You could move the order of **Contents** if you wish by modifying the theme.html file.

Here we will show the use of Right Blocks with your Xoops site. If you have **no use** of the right blocks, then you can use right blocks in a variety of ways. The followings are two examples

(they are just ideas, to accomplish them you would need make necessary changes to your theme file, such as change current two-columns display to 3-coloumns):

3-Columns Center-Block

Left **Center-Center**
Center-Left **Right-Blocks** **Center-Right**
Contents

2 Center-Center Blocks

Left **Center-Center**
Center-Left **Center-Right**
Right-Blocks
Contents

You may try as many combinations as you wish.

The only limitation of Xoops is that you can't create your own custom blocks and you will not be able to use the right blocks in other ways if your right blocks are in use.

If we were able to create custom blocks then Xoops would truly be the most flexible system in presenting information in blocks. You could literally put a block to anywhere you wish if custom blocks were allowed. **Imagine that!**

5. Try Your Own

As usual, we are hoping the information will be useful and you will be able to customize and use it on your own sites.

Xoops has been proven to be eXtensible and customizable (with exception of the rigid rendering/assigning blocks) and you have the ultimate control in your page design. Smarty just makes the job much easier.

Please post suggestions and feedback on Xoops-Tips.com site.