


INTRODUCTION

{META}		{CUSTOM_HEADER}	
	{GAL_NAME}		
	{GAL_DESCRIPTION}		
	{SYS_MENU}		
	{SUB_MENU}		
{LANGUAGE_SELECT_FLAGS}			
{LANGUAGE_SELECT_LIST}{THEME_SELECT_LIST}			
{ADMIN_MENU}{GALLERY}			
{VANITY}		{CUSTOM_FOOTER}	

What the heck is this?????!!!!

This, my friend, is the *template.html* from Coppermine's sample theme.

Coppermine themes depend on three files (*style.css*, *template.html*, and *theme.php*) and one folder (*images*), which can contain several files (i.e. images). You can have more, but you **must** have these four "core" items. I'll explain more about this as I go on.

Where to start...where to start...where to start...

Well, since I've already got the template up top here, why don't we look at that?

To begin with, what are those squiggly thingies? The **brackets** are there to tell Coppermine "Hey! I've been put together somewhere else but this is where you can put me now". You can move these around; put them anywhere on the page you'd like.

DO NOT REMOVE THEM

If you remove them, parts of your gallery won't work.

So what are they?!

Let's take a look and see if I can explain these "tags" for you.

INTRODUCTION

- {META} This is the built in meta data for you gallery.
- {CUSTOM_HEADER} What does the name say to you?
- {GAL_NAME} This is the gallery name you enter from the “General Settings” selection of the configuration.
- {GAL_DESCRIPTION} Again, from the “General Settings” of the configuration.
- {SYS_MENU} This is the menu that includes *Home, Login, My Gallery*, etc.
- {SUB_MENU} This menu contains *Album List, Last Uploads, Last Comments*, etc.
- {LANGUAGE_SELECT_FLAGS} This is the list of flags that are displayed for language selections.
- {LANGUAGE_SELECT_LIST} A drop down text menu for language selection.
- {THEME_SELECT_LIST} A drop down list to change themes.
- {ADMIN_MENU} The menu that contains *Config, Categories, Albums*, etc.
- {GALLERY} This is the actual picture section.

Make a copy of the sample theme and play around with their placement. You can see how it will look by entering *URL to your photo page/index.php?theme=sample*.

Go ahead...I'll wait...

Wasn't that interesting. Did you get things all out of whack? Don't worry. We'll clean it up as we go along.

Before we go any further, I must stress the point that you will need at least some *basic* knowledge of how CSS works.

**If you don't have any experience working with CSS,
stop reading this and gain some knowledge there first.**

I could explain it but that's a totally different animal deserving of its own zoo.

Okay, since you're still here, I'll assume you'll have an idea of what we talk about from this point forward.

INTRODUCTION

You don't really need to know PHP in order to design your own theme in Coppermine. That's one of the cool things about using stuff like this. All that programming stuff is already done.

Remember those squiggly thingies ...That's right! The brackets. There in the PHP code of the **theme.php** file as well. And they serve the same purpose.

Now, it's kind of hard to separate discussing the **theme.php** file from the **style.css** file since they work hand in hand to generate the overall webpage. So first, I think I'll break down what each part of the code covers. Then we'll go back and look at how the **style.css** affects how it's displayed.

Here we go...

The Credit

```
/******  
Coppermine Photo Gallery  
*****  
Copyright (c) 2003-2006 Coppermine Dev Team  
v1.1 originally written by Gregory DEMAR  
  
This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.  
*****  
Coppermine version: 1.4.10  
$Source$  
$Revision: 3275 $  
$Author: gaugau $  
$Date: 2006-09-03 12:10:47 +0200 (So, 03 Sep 2006) $  
*****/
```

This is the credits section of the code. DO NOT TAKE THIS OUT! These guys have worked HARD on this and they deserve ALL the credit they can get.

THANKS COPPERMINE DEV. TEAM!

The Setup

```
// The following terms can be defined in theme.php  
// ("THEME_HAS_RATING_GRAPHICS", 1) : The location for the ratings graphics will  
// be directed to the themes images folder.  
// RUBBISH : images/rating0.gif  
// POOR : images/rating1.gif  
// FAIR : images/rating2.gif  
// GOOD : images/rating3.gif  
// EXCELLENT : images/rating4.gif  
// GREAT : images/rating5.gif  
// ("THEME_HAS_NAVBAR_GRAPHICS", 1); : The location for the navbar graphics will  
// be directed to the themes images folder.  
// Back to thumbnails : images/thumbnails.gif  
// Picture Information : images/info.gif  
// Slideshow : images/slideshow.gif  
// Report to admin : images/report.gif  
// Ecard : images/ecard.gif  
// Previous : images/prev.gif  
// Next : images/next.gif
```

THEME.PHP

```
// ('THEME_HAS_FILM_STRIP_GRAPHIC', 1) : The location for the film strip graphics will
// be directed to the themes images folder.
// tile : images/tile.gif
// ('THEME_HAS_FILM_STRIP_GRAPHICS', 1) : The location for the film strip graphics will
// be directed to the themes images folder.
// tile on the top : images/tile1.gif
// tile on the bottom : images/tile2.gif
// ('THEME_HAS_NO_SYS_MENU_BUTTONS', 1) : When present the system won't attempt to
// replace {BUTTONS} in the SYS_MENU template
// The entire block needs to be present like in Coppermine 1.3 themes
// ('THEME_HAS_NO_SUB_MENU_BUTTONS', 1) When present the system won't attempt to
// replace {BUTTONS} in the SUB_MENU template
// The entire block needs to be present like in Coppermine 1.3 themes
// ('THEME_IS_XHTML10_TRANSITIONAL', 1) : If theme is defined as
// XHTML10_TRANSITIONAL the VANITY footer will be enabled
// if the theme has a {VANITY} token in its template.html. Don't enable this if you have
// modified the code! See the
// docs/theme.html documentation for validation methodology.
```

HOLY COW! What's all this!

Don't be alarmed. The real meat of ALL of this is:

```
('THEME_HAS_RATING_GRAPHICS', 1);
('THEME_HAS_NAVBAR_GRAPHICS', 1);
('THEME_HAS_FILM_STRIP_GRAPHIC', 1);
('THEME_HAS_FILM_STRIP_GRAPHICS', 1);
('THEME_HAS_NO_SYS_MENU_BUTTONS', 1);
('THEME_HAS_NO_SUB_MENU_BUTTONS', 1);
('THEME_IS_XHTML10_TRANSITIONAL', 1);
```

Whew! That's a lot less to look at.

What do these lines do? These lines are used if you want to define you own **custom** graphics and/or menu system.

Coppermine comes with standard rating graphics, film strip graphics and menu system. But it also allows you the capability of customizing one or all of them. If you use ANY of them, make sure you remove the comment indicators (`//`), add the word **define**, and end the line with a semi-colon (`;`). If you don't do these three things, it won't work and you may get errors that you don't understand.

```
define('THEME_HAS_RATING_GRAPHICS', 1);
```

This allows you to define you own ratings graphics in the “images” directory of your theme.



These graphics **MUST BE** named:

THEME.PHP

rating0.gif
rating1.gif
rating2.gif
rating3.gif
rating4.gif
rating5.gif

```
define('THEME_HAS_NAVBAR_GRAPHICS', 1);
```

When you are viewing individual pictures from an album, there is a navigation bar just above the picture.

The icons in this “NAVBAR” are the “NAVBAR_GRAPHICS”.



Once again, have to be in the “images” directory and they have SPECIFIC NAMES:

thumbnails.gif
info.gif
slideshow.gif
report.gif
ecard.gif
prev.gif
next.gif

```
define('THEME_HAS_FILM_STRIP_GRAPHIC', 1);
```

This is the graphic that’s repeated to create the strip that you see above and below the list of pictures below the single picture you are viewing from your album (huh?).



{PICTURE} {PICTURE}



OH!

This one must be named

tile.gif

```
define('THEME_HAS_FILM_STRIP_GRAPHICS', 1);
```

THEME.PHP

Same as above, but this one will use one graphic for the top strip and a different graphic for the bottom.

tile1.gif
tile2.gif

```
define('THEME_HAS_NO_SYS_MENU_BUTTONS', 1);
```

This overrides the standard layout of the system menu and allows you to specify the layout and/or graphics of the *system* menu.

More on this in a little later.

```
define('THEME_HAS_NO_SUB_MENU_BUTTONS', 1);
```

This overrides the standard layout of the *sub* menu and allow you to specify the layout and/or graphics of the *sub* menu.

Again, more on this a little later.

```
define('THEME_IS_XHTML10_TRANSITIONAL',1);
```

Just as the text implies. This will specify to the internet world that you theme is XHTML10 compliant.

Don't use this one until you have verified the compliance of your theme!

Okay. Let's start customizing...

```
// Creates buttons from a template using an array of tokens
// this function is used in this file it needs to be declared before being called.
function assemble_template_buttons($template_buttons,$buttons) {
    $counter=0;
    $output="";

    foreach ($buttons as $button) {
        if (isset($button[4])) {
            $spacer=$button[4];
        } else {
            $spacer="";
        }

        $params = array(
            '{SPACER}' => $spacer,
            '{BLOCK_ID}' => $button[3],
            '{HREF_TGT}' => $button[2],
            '{HREF_TITLE}' => $button[1],
            '{HREF_LNK}' => $button[0]
        );
        $output.=template_eval($template_buttons, $params);
    }
}
```

THEME.PHP

```
return $output;
}

// Creates an array of tokens to be used with function assemble_template_buttons
// this function is used in this file it needs to be declared before being called.
function addbutton(&$menu,$href_lnk,$href_title,$href_tgt,$block_id,$spacer) {
    $menu[]=array($href_lnk,$href_title,$href_tgt,$block_id,$spacer);
}

// HTML template for sys_menu
$template_sys_menu = <<<EOT
    {BUTTONS}
EOT;

// HTML template for sub_menu
$template_sub_menu = $template_sys_menu;

if (!defined('THEME_HAS_NO_SYS_MENU_BUTTONS')) {

    // HTML template for template sys_menu spacer
    $template_sys_menu_spacer = "::";

    // HTML template for template sys_menu buttons
    $template_sys_menu_button = <<<EOT
    <!-- BEGIN {BLOCK_ID} -->
        <a href="{HREF_TGT}" title="{HREF_TITLE}">{HREF_LNK}</a> {SPACER}
    <!-- END {BLOCK_ID} -->
    EOT;

    // HTML template for template sys_menu buttons
    // {HREF_LNK}{HREF_TITLE}{HREF_TGT}{BLOCK_ID}{SPACER}

    addbutton($sys_menu_buttons,'{HOME_LNK}','{HOME_TITLE}','{HOME_TGT}','home',$tem
    plate_sys_menu_spacer);

    addbutton($sys_menu_buttons,'{MY_GAL_LNK}','{MY_GAL_TITLE}','{MY_GAL_TGT}','my
    _gallery',$template_sys_menu_spacer);

    addbutton($sys_menu_buttons,'{MEMBERLIST_LNK}','{MEMBERLIST_TITLE}','{MEMBER
    LIST_TGT}','allow_memberlist',$template_sys_menu_spacer);

    addbutton($sys_menu_buttons,'{MY_PROF_LNK}','{MY_PROF_TITLE}','{MY_PROF_TGT}','
    my_profile',$template_sys_menu_spacer);

    addbutton($sys_menu_buttons,'{ADM_MODE_LNK}','{ADM_MODE_TITLE}','{ADM_MODE
    _TGT}','enter_admin_mode',$template_sys_menu_spacer);

    addbutton($sys_menu_buttons,'{USR_MODE_LNK}','{USR_MODE_TITLE}','{USR_MODE_T
    GT}','leave_admin_mode',$template_sys_menu_spacer);

    addbutton($sys_menu_buttons,'{UPL_PIC_LNK}','{UPL_PIC_TITLE}','{UPL_PIC_TGT}','uplo
    ad_pic',$template_sys_menu_spacer);

    addbutton($sys_menu_buttons,'{REGISTER_LNK}','{REGISTER_TITLE}','{REGISTER_TGT}'
    ,register',$template_sys_menu_spacer);
```


THEME.PHP

```
addbutton($sys_menu_buttons, '{FAQ_LNK}', '{FAQ_TITLE}', '{FAQ_TGT}', 'faq', $template_sys_menu_spacer);
    addbutton($sys_menu_buttons, '{LOGIN_LNK}', '{LOGIN_TITLE}', '{LOGIN_TGT}', 'login', "");

addbutton($sys_menu_buttons, '{LOGOUT_LNK}', '{LOGOUT_TITLE}', '{LOGOUT_TGT}', 'logout', "");
    // Login and Logout don't have a spacer as only one is shown, and either would be the last option.

    $params = array('{BUTTONS}' =>
assemble_template_buttons($template_sys_menu_button, $sys_menu_buttons));
    $template_sys_menu = template_eval($template_sys_menu, $params);
}

if (!defined('THEME_HAS_NO_SUB_MENU_BUTTONS')) {

    // HTML template for template sub_menu spacer
    $template_sub_menu_spacer = $template_sys_menu_spacer;

    // HTML template for template sub_menu buttons
    $template_sub_menu_button = $template_sys_menu_button;

    // HTML template for template sub_menu buttons
    // {HREF_LNK}{HREF_TITLE}{HREF_TGT}{BLOCK_ID}{SPACER}

    addbutton($sub_menu_buttons, '{CUSTOM_LNK_LNK}', '{CUSTOM_LNK_TITLE}', '{CUSTOM_LNK_TGT}', 'custom_link', $template_sub_menu_spacer);

    addbutton($sub_menu_buttons, '{ALB_LIST_LNK}', '{ALB_LIST_TITLE}', '{ALB_LIST_TGT}', 'album_list', $template_sub_menu_spacer);

    addbutton($sub_menu_buttons, '{LASTUP_LNK}', '{LASTUP_TITLE}', '{LASTUP_TGT}', 'lastup', $template_sub_menu_spacer);

    addbutton($sub_menu_buttons, '{LASTCOM_LNK}', '{LASTCOM_TITLE}', '{LASTCOM_TGT}', 'lastcom', $template_sub_menu_spacer);

    addbutton($sub_menu_buttons, '{TOPN_LNK}', '{TOPN_TITLE}', '{TOPN_TGT}', 'topn', $template_sub_menu_spacer);

    addbutton($sub_menu_buttons, '{TOPRATED_LNK}', '{TOPRATED_TITLE}', '{TOPRATED_TGT}', 'toprated', $template_sub_menu_spacer);

    addbutton($sub_menu_buttons, '{FAV_LNK}', '{FAV_TITLE}', '{FAV_TGT}', 'favpics', $template_sub_menu_spacer);

    addbutton($sub_menu_buttons, '{SEARCH_LNK}', '{SEARCH_TITLE}', '{SEARCH_TGT}', 'search', "");

    $params = array('{BUTTONS}' =>
assemble_template_buttons($template_sub_menu_button, $sub_menu_buttons));
    $template_sub_menu = template_eval($template_sub_menu, $params);
}
```

THEME.PHP

This is used by the Coppermine templates to generate the standard menus.

If you're going to create custom menus you can safely remove all of the above code.

To create customized system menus, we're going to need to customize the code.

Menus

There are four (4) menus in Coppermine.

- The **System Menu** (remember {SYS_MENU}?)
- The **Sub Menu** (a.k.a. {SUB_MENU})
- The **Administration Menu** (silly enough called {ADMIN_MENU})
- The **User Admin Menu** (used by users to edit their own settings)

You can create a customized “look and feel” for your system menu if you follow a few simple rules...that's **rules**. As in “**THESE MUST BE FOLLOWED**”.

Let's look at each one separately...

The System Menu – {SYS_MENU}

Home :: Register :: Login

First, we need to tell Coppermine that we are making a template for the system menu (as well as making a note to others that this is what we're doing):

```
// HTML template for sys menu
$template_sys_menu = <<<EOT
```

Now, because we are creating a *custom* menu, we need to tell Coppermine where each part is. This is done by “tagging” each item of the menu. We need a “BEGIN” tag and an “END” tag.

```
<!-- BEGIN xxxxx -->
      (CODE)
<!-- END xxxxx -->
```

How does this look in “real” code? Let's see.

THEME.PHP

```
<!-- BEGIN home -->
<a href="{HOME_TGT}" title="{HOME_TITLE}">{HOME_LNK}</a>
<!-- END home -->
```

This code is telling Coppermine to set up a link for Coppermine's home page. But this, as it is, does not do anything different than the default mode. However, if we were to add the HTML tag to this code

```
<!-- BEGIN home -->
<li><a href="{HOME_TGT}" title="{HOME_TITLE}">{HOME_LNK}</a></li>
<!-- END home -->
```

Coppermine would display

- [HOME](#)

instead of

[HOME ::](#)

Nothing really amazing in and of itself. But, when you combine this with some CSS set up in your stylesheet and add some definitions, etc, you can get your system menu to look just about anyway you'd like it to.



Cool, huh.

Okay, what are the tags?

```
<!-- BEGIN home -->
<a href="{HOME_TGT}" title="{HOME_TITLE}">{HOME_LNK}</a>
<!-- END home -->
<!-- BEGIN my_gallery -->
<a href="{MY_GAL_TGT}" title="{MY_GAL_TITLE}">{MY_GAL_LNK}</a>
<!-- END my_gallery -->
<!-- BEGIN allow_memberlist -->
<a href="{MEMBERLIST_TGT}" title="{MEMBERLIST_TITLE}">{MEMBERLIST_LNK}</a>
<!-- END allow_memberlist -->
<!-- BEGIN my_profile -->
<a href="{MY_PROF_TGT}" title="{MY_PROF_LNK}">{MY_PROF_LNK}</a>
<!-- END my_profile -->
<!-- BEGIN faq -->
<a href="{FAQ_TGT}" title="{FAQ_TITLE}">{FAQ_LNK}</a>
<!-- END faq -->
<!-- BEGIN enter_admin_mode -->
<a href="{ADM_MODE_TGT}" title="{ADM_MODE_TITLE}">{ADM_MODE_LNK}</a>
<!-- END enter_admin_mode -->
<!-- BEGIN leave_admin_mode -->
<a href="{USR_MODE_TGT}" title="{USR_MODE_TITLE}">{USR_MODE_LNK}</a>
<!-- END leave_admin_mode -->
```

THEME.PHP

```
<!-- BEGIN upload_pic -->
<a href="{UPL_PIC_TGT}" title="{UPL_PIC_TITLE}">{UPL_PIC_LNK}</a>
<!-- END upload_pic -->
<!-- BEGIN register -->
<a href="{REGISTER_TGT}" title="{REGISTER_TITLE}">{REGISTER_LNK}</a>
<!-- END register -->
<!-- BEGIN login -->
<a href="{LOGIN_TGT}" title="{LOGIN_LNK}">{LOGIN_LNK}</a>
<!-- END login -->
<!-- BEGIN logout -->
<a href="{LOGOUT_TGT}" title="{LOGOUT_LNK}">{LOGOUT_LNK}</a>
<!-- END logout -->
```

Then you **MUST** end the section with

EOT;

and don't forget the semicolon!

The Sub Menu – {SUB_MENU}

Album list :: Last uploads :: Last comments :: Most viewed :: Top rated :: My Favorites :: Search

Like the {SYS_MENU} we need to tell Coppermine that we are defining our own sub menu and provide tagging so Coppermine will know where to begin and end each part of the menu.

```
// HTML template for sub menu
$template_sub_menu = <<<EOT
<!-- BEGIN custom_link -->
<a href="{CUSTOM_LNK_TGT}" title="{CUSTOM_LNK_TITLE}">{CUSTOM_LNK_LNK}</a>
<!-- END custom_link -->
<!-- BEGIN album_list -->
<a href="{ALB_LIST_TGT}" title="{ALB_LIST_TITLE}">{ALB_LIST_LNK}</a>
<!-- END album_list -->
<a href="{LASTUP_TGT}" title="{LASTUP_LNK}">{LASTUP_LNK}</a>
<a href="{LASTCOM_TGT}" title="{LASTCOM_LNK}">{LASTCOM_LNK}</a>
<a href="{TOPN_TGT}" title="{TOPN_LNK}">{TOPN_LNK}</a>
<a href="{TOPRATED_TGT}" title="{TOPRATED_LNK}">{TOPRATED_LNK}</a>
<a href="{FAV_TGT}" title="{FAV_LNK}">{FAV_LNK}</a>
<a href="{SEARCH_TGT}" title="{SEARCH_LNK}">{SEARCH_LNK}</a>
EOT;
```

“Hey! Part of that’s not ‘tagged’!!”

That’s okay. As long as you provide some sort of formatting for these lines (like the aforementioned HTML tags). If you don’t provide ANY formatting, you’ll simply get a line of textual links.

Remember. The whole point of all of this is to *customize*.

THEME.PHP

The Administration Menu – {ADMIN_MENU}

Config	Categories	Albums	Groups	Users	Ban Users	Review Comments	Sort my pictures	Batch add files	Admin Tools	My profile	Documentation
--------	------------	--------	--------	-------	-----------	-----------------	------------------	-----------------	-------------	------------	---------------

Okay. This one's got quite a bit of stuff in it, so pay close attention.

// HTML template for gallery admin menu

```
$template_gallery_admin_menu = <<<EOT
<!-- BEGIN admin_approval -->
<a href="editpics.php?mode=upload_approval" title="{UPL_APP_TITLE}">{UPL_APP_LNK}</a>
<!-- END admin_approval -->
<a href="admin.php" title="{ADMIN_TITLE}">{ADMIN_LNK}</a>
<a href="catmgr.php" title="{CATEGORIES_TITLE}">{CATEGORIES_LNK}</a>
<a href="albmgr.php{CATL}" title="{ALBUMS_TITLE}">{ALBUMS_LNK}</a>
<a href="groupmgr.php" title="{GROUPS_TITLE}">{GROUPS_LNK}</a>
<a href="usermgr.php" title="{USERS_TITLE}">{USERS_LNK}</a>
<a href="banning.php" title="{BAN_TITLE}">{BAN_LNK}</a>
<a href="reviewcom.php" title="{COMMENTS_TITLE}">{COMMENTS_LNK}</a>
<!-- BEGIN log_ecards -->
<a href="db_ecard.php" title="{DB_ECARD_TITLE}">{DB_ECARD_LNK}</a>
<!-- END log_ecards -->
<a href="picmgr.php" title="{PICTURES_TITLE}">{PICTURES_LNK}</a>
<a href="searchnew.php" title="{SEARCHNEW_TITLE}">{SEARCHNEW_LNK}</a>
<a href="util.php" title="{UTIL_TITLE}">{UTIL_LNK}</a>
<a href="profile.php?op=edit_profile" title="{MY_PROF_TITLE}">{MY_PROF_LNK}</a>
<!-- BEGIN documentation -->
<a href="{DOCUMENTATION_HREF}" title="{DOCUMENTATION_TITLE}"
target="cpg_documentation">{DOCUMENTATION_LNK}</a>
<!-- END documentation -->
EOT;
```