

Anatomy of the Default Theme

Mike Muin

Anatomy of the Default Theme

Mike Muin

Published 2004-05-21

Copyright © 2004

Table of Contents

1. Introduction	1
1.1. Themes vs. Templates	1
1.2. Theme Directory	1
1.3. Smarty	1
2. HTML Files	3
2.1. theme.html	3
2.1.1. Diagram of the Default Theme	3
2.1.2. DOCTYPE	3
2.1.3. Meta Tags	4
2.1.4. Title	4
2.1.5. Links	5
2.1.6. JavaScript	5
2.1.7. Header Table	5
2.1.8. Content Table - Left Column	6
2.1.9. Content Table - Center Blocks	7
2.1.10. Content Table - Center-Center Blocks	7
2.1.11. Content Table - Center-Left Blocks	8
2.1.12. Content Table - Center-Right Blocks	8
2.1.13. Content Table - Module Contents	9
2.1.14. Content Table - Right Column	9
2.1.15. Footer Table	10
2.2. theme_blockleft.html	10
2.3. theme_blockcenter_c.html	11
2.4. theme_blockcenter_l.html	11
2.5. theme_blockcenter_r.html	12
2.6. theme_blockright.html	12
3. CSS Files	13
3.1. style.css	13
3.2. styleMAC.css	16
3.3. styleNN.css	16
4. Images	19
4.1. logo.gif	19
4.2. hbar.gif	19
4.3. poweredby.gif	19
5. Basic Theme-Building Tips	21
5.1. Installing a Theme	21
5.2. Smarty Debugging	21
Index	23



List of Figures

2.1. Basic Layout	3
-------------------------	---



Chapter 1. Introduction

The downloaded XOOPS package contains 3 themes: **default**, **x2t** and **phpkaox**. Of these three, the default theme is the most commonly used. It is the theme used by www.xoops.org

Most custom themes are modified versions of the XOOPS default theme. It is highly advised to study the structure of the XOOPS default theme before starting to create custom themes.

This document describes the basic structure of the default theme. Although some basic theme-building tips are given, this document is not an extensive theme-building tutorial.

1.1. Themes vs. Templates

In **XOOPS**, *themes* and *templates* are 2 different things

Themes provide the basic structure and overall layout of the *whole* site. Unless heavily modified with Smarty tags, Style Sheets or PHP codes, themes dictate the common elements found in all pages of the site, e.g. site colors, location of site logo, banners, width of left column and presence of a header bar or footer bar.

Templates, on the other hand, control layout and design of page elements. Almost all page elements, unless rendered from the PHP script, are controlled by templates. These include both *blocks*, e.g. Main Menu, User Menu, Newest Members, Recent Topics and custom blocks, and *module content*, e.g. index page of News, layout of NewBB posts and layout of individual link information in MyLinks module.

Themes are modified and edited outside of the XOOPS Admin Control Panel, often using an HTML editor and FTP client. Templates are modified from within the XOOPS Admin Control Panel through the Templates Administration.

These two concepts provide flexibility in designing a XOOPS site. Themes provide gross control of the whole site, while templates provide fine manipulations of page details.

1.2. Theme Directory

The XOOPS Theme Directory is the `themes/` folder from the main XOOPS directory. Any folder within that directory is considered a theme. With the default installation, there are 3 folders available that represent the 3 themes: `default/`, `phpkaox/` and `x2t/`.

It is important to remember that when you install your theme inside the Theme Directory, the folder name of your theme becomes the name of the theme from within the XOOPS Admin Control Panel.

1.3. Smarty

XOOPS uses the Smarty Template Engine and XOOPS themes have Smarty tags embedded in the HTML code. Smarty tags are identified by the delimiters `<{ and }>`. If you see variables, conditionals or functions inside those delimiters, they are considered Smarty tags.

Smarty tags speed up site loading because the content of these tags have already been processed and compiled. So instead of creating a script to call out the site url, the username or even the theme folder, a Smarty tag is used.

The great thing about using Smarty tags inside the `theme.html` is that it allows better control of layout by having a Smarty tag represent a content instead of a php script. Smarty tags can be placed in between HTML tags to enable formatting.

As you go through the document, take note of the different Smarty tags used and how they are used in relation to HTML codes.

Learn more about Smarty at <http://smarty.php.net>.



Chapter 2. HTML Files

The **XOOPS** default theme contains 6 HTML files: `theme.html`, `theme_blockleft.html`, `theme_blockcenter_c.html`, `theme_blockcenter_l.html`, `theme_blockcenter_r.html` and `theme_blockright.html`.

Each file will be discussed in detail.

2.1. theme.html

Undoubtedly, the `theme.html` is the most important HTML file in all **XOOPS** themes. This file dictates the overall design, structure and layout of your site.

2.1.1. Diagram of the Default Theme

Below is a diagrammatic representation of the basic layout of the `theme.html` of the Default theme. It has *3 main tables* for the header, content and footer and *1 nested table* for the center blocks. Take note that there are actually *no spaces* in between the tables once displayed on the browser.

With a basic understanding of HTML, it is possible to manipulate the Header table, the Content table and the Footer table independent from each other.

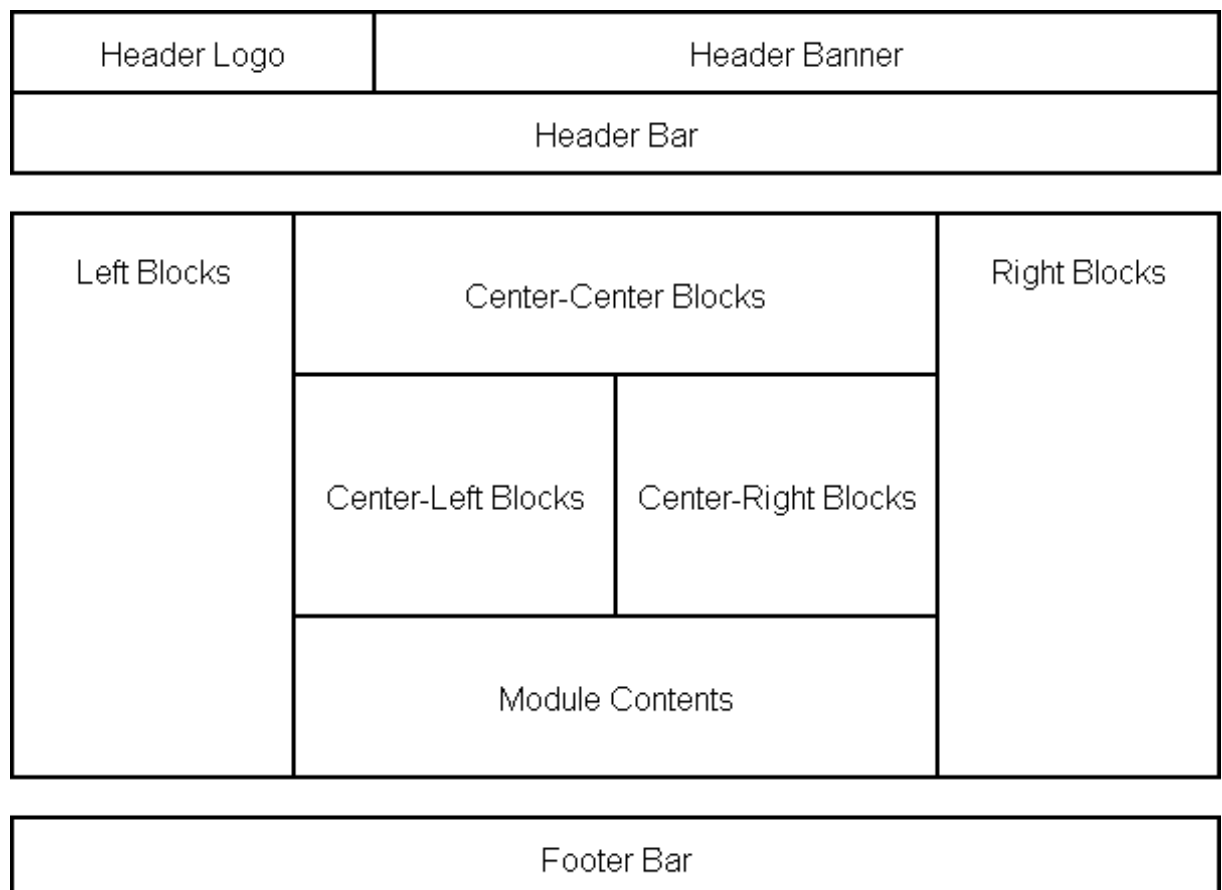


Figure 2.1. Basic Layout

2.1.2. DOCTYPE

Line 1-2:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="<{$xoops_langcode}>"><meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">" lang="<{$xoops_langcode}>">
```

Each HTML document should have a Document Type Definition. XOOPS themes adhere to XHTML 1.0 Transitional. XHTML is a family of current and future document types and modules that reproduce, subset, and extend HTML 4



Theme-Building Tips

- The DOCTYPE format adheres to recommended standards. There are very few reasons to change this.
- Keep it in mind when custom-coding or editing themes and templates. This is important for HTML validation.

2.1.3. Meta Tags

Line 3-12:

```
<head>
<meta http-equiv="content-type" content="text/html;
charset=<{$xoops_charset}>" />
<meta http-equiv="content-language" content="<{$xoops_langcode}>" />
<meta name="robots" content="<{$xoops_meta_robots}>" />
<meta name="keywords" content="<{$xoops_meta_keywords}>" />
<meta name="description" content="<{$xoops_meta_description}>" />
<meta name="rating" content="<{$xoops_meta_rating}>" />
<meta name="author" content="<{$xoops_meta_author}>" />
<meta name="copyright" content="<{$xoops_meta_copyright}>" />
<meta name="generator" content="XOOPS" />
```

The Meta tags form the first part of the head of the HTML and are mostly generated from Smarty tags. It should not be edited directly within the theme.html file. Meta information should be changed from the **System Admin > Preferences > Meta Tags and Footer**.



Theme-Building Tips

- Initial XOOPS installation automatically populates the database with XOOPS-specific Meta Tags. Revise the Meta Tags before the site goes public.
- Choose Meta Tags wisely. With the current XOOPS theme system, meta tags are implemented site wide. Meta-tags cannot be implemented per page.

2.1.4. Title

Line 13:

```
<title><{$xoops_sitename}> - <{$xoops_pagetitle}></title>
```

There are 2 Smarty tags in the title:

- <{\$xoops_sitename}> - This is the *Site name* entered in the **System Admin > Preferences > General Settings**. This Smarty tag can be used anywhere in the theme.html or template files where the name of the

site is needed.

- `<{$xoops_pagetitle}>` - This is a dynamically derived Smarty tag that will change as the page changes. The default content of the Smarty tag is the *Site Slogan* from **System Admin > Preferences > General Settings**. If a module is specified as the front page, or when browsing the site within a module, the module name will be used and displayed instead of the Site Slogan.



Theme-Building Tips

- `<{$xoops_sitename}>` is an important Smarty tag to remember when building custom themes and editing templates. They can be used where the Site Name is needed and can be embedded within HTML tags for formatting.

2.1.5. Links

Line 14-18:

```
<link href="<{$xoops_url}>/favicon.ico" rel="SHORTCUT ICON" />
<link rel="stylesheet" type="text/css" media="all"
href="<{$xoops_url}>/xoops.css" />
<link rel="stylesheet" type="text/css" media="all"
href="<{$xoops_themecss}>" />
<!-- RMV: added module header -->
<{$xoops_module_header}>
```

This is the part where the favicon.ico and Cascading Style Sheets are linked. This is the first instance where the important `<{$xoops_url}>` Smarty tag is used.

- `<{$xoops_url}>` - This is the site URL as defined in the `mainfile.php`. This Smarty tag can be used anywhere in the `theme.html` or template files where the site URL is needed.
- `<{$xoops_themecss}>` - This Smarty tag defines the location of the specific `style.css` of the theme chosen as default. It has little use within the body part of the `theme.html`.



Theme-Building Tips

- `<{$xoops_url}>` is an important Smarty tag to remember when building custom themes and editing templates. They can be used where the Site URL is needed and can be embedded within HTML tags for absolute path calls (images and links) and formatting.

2.1.6. JavaScript

Line 19-23:

```
<script type="text/javascript">
<!--
<{$xoops_js}>
//-->
</script>
</head>
```

This part loads the XOOPS JavaScript functions.

2.1.7. Header Table

Line 24-34:

```
<body>
  <table cellspacing="0">
    <tr id="header">
      <td id="headerlogo"><a href="<{$xoops_url}>/"></a></td>
      <td id="headerbanner"><{$xoops_banner}></td>
    </tr>
    <tr>
      <td id="headerbar" colspan="2"> </td>
    </tr>
  </table>
```

The header table forms the first part of the body of the HTML. It contains the site logo, space for banners and the image bar separating the header from the content. The formatting of the table and each table cell is handled by the `style.css` of the default theme.

Aside from `<{$xoops_url}>`, two Smarty tags are found:

- `<{$xoops_imageurl}>` - This Smarty tag represents the path to your default theme directory, e.g. `http://www.yoursite.com/theme/default/`. This Smarty tag can be used anywhere in the `theme.html` or template files where the path to the theme directory is needed.
- `<{$xoops_banner}>` - This Smarty tag retrieves the banners from **System Admin > Banners**



Theme-Building Tips

- `<{$xoops_imageurl}>` can be used to define paths for images and files found within your theme directory. It is not exclusive for images. Also remember that it already has the trailing slash in it.

2.1.8. Content Table - Left Column

Line 36-45:

```
<table cellspacing="0">
  <tr>
    <td id="leftcolumn">
      <!-- Start left blocks loop -->
      <{foreach item=block from=$xoops_lblocks}>
        <{include file="default/theme_blockleft.html"}>
      </foreach>
      <!-- End left blocks loop -->
    </td>
```

This piece of code starts the all-important Content Table. This is where all block and module contents are displayed. The Content Table tag opens at Line 36 and closes at Line 106. There are many codes in between that will dictate the look of the site.

The Content Table has only 1 row with 3 cells. The first cell contains the left column contents. The layout is controlled by the `style.css` using the id "leftcolumn".

Within that table cell are Smarty tags that calls a loop and displays the different blocks for the Left Column. The layout and structure of each left column block is handled by the `theme_blockleft.html` file and the `style.css`.

Specific details on the left column blocks is provided in the `theme_blockleft.html` section.



Theme-Building Tips

- You can totally eliminate the use of a different HTML file for the left blocks by copying the code inside the `theme_blockleft.html` file and pasting them inside the `<{foreach}>` Smarty loop.
- If you plan to continue using the `theme_blockleft.html` file using the `<{include}>` Smarty tag, remember to change the directory to suit your theme folder. The default directory in the tag is the “default” folder, i.e. `default/theme_blockleft.html`. If your theme folder is “mynewtheme”, then the include function should state `mynewtheme/theme_blockleft.html`.

2.1.9. Content Table - Center Blocks

Line 47-50:

```
<td id="centercolumn">

    <!-- Display center blocks if any -->
    <{if $xoops_showcblock == 1}>
```

The second cell within the Content Table is the Center Column. The layout and format settings are controlled by the `style.css` using the id “centercolumn”.

The Smarty tag `<{if $xoops_showcblock == 1}>` checks if there are center blocks to be displayed. If there are, the HTML and Smarty code in between the conditionals will be called, i.e. the center blocks will be displayed.

Line 85:

```
</if}>
    <!-- End display center blocks -->
```

The `<{if}>` Smarty tag is closed at Line 85 with an `</if}>` Smarty tag. This part ends all calls made to the different center blocks.

Remember that there are 3 types of center blocks: **center-center**, **center-left** and **center-right**. Specific details on these types are provided below.

2.1.10. Content Table - Center-Center Blocks

Line 52-63:

```
<table cellpadding="0">
    <tr>
        <td id="centerCcolumn" colspan="2">

            <!-- Start center-center blocks loop -->
            <{foreach item=block from=$xoops_ccblocks}>
                <{include file="default/theme_blockcenter_c.html"}>
            </foreach}>
            <!-- End center-center blocks loop -->

        </td>
    </tr>
```

The center blocks are all contained in a nested table inside the **Center Column** (`<td id="centercolumn">`). The nested table, let's call it the **Center Blocks** table as opposed to the Center Column table, has 2 rows.

The first row has one cell (`<td id="centerCcolumn" colspan="2">`) whose layout is done through

the `style.css`. This row contains the Smarty tags that call the **Center-Center** blocks through the `<{foreach}>` Smarty loop tag.

Notice that in include function calls the `theme_blockcenter_c.html` file. This file dictates the look and layout of the center-center block, specifically how the block title and content are displayed.



Theme-Building Tips

- You can totally eliminate the use of a different HTML file for the center-center blocks by copying the code inside the `theme_blockcenter_c.html` file and pasting them inside the `<{foreach}>` Smarty loop.
- If you plan to continue using the `theme_blockcenter_c.html` file using the `<{include}>` Smarty tag, remember to change the directory to suit your theme folder, as mentioned before.
- The td style id “centerCcolumn” acts on the cell that “houses” the center-center blocks, and not the center-center blocks themselves. For changes on the center-center block layout, edit the `theme_blockcenter_c.html` file.

2.1.11. Content Table - Center-Left Blocks

Line 64-71

```
<tr>
    <td id="centerLcolumn">
        <!-- Start center-left blocks loop -->
        <{foreach item=block from=$xoops_clblocks}>
            <{include file="default/theme_blockcenter_l.html"}>
        </foreach>
        <!-- End center-left blocks loop -->
    </td>
</tr>
```

The second row of the **Center Blocks** table has 2 cells. The first cell contains the Smarty loop that calls on the **Center-Left** blocks. The cell layout is managed through td style id “centerLcolumn”. As with the center-center blocks, a Smarty include tag calls on a separate file, `theme_blockcenter_l.html`, for the layout of the center-left blocks.



Theme-Building Tips

- Check Theme-Building Tips (Theme-Building Tips [8]) for the center-center blocks. They should be very similar.

2.1.12. Content Table - Center-Right Blocks

Line 73-83:

```
</td><td id="centerRcolumn">
    <!-- Start center-right blocks loop -->
    <{foreach item=block from=$xoops_crblocks}>
        <{include file="default/theme_blockcenter_r.html"}>
    </foreach>
    <!-- End center-right blocks loop -->
</td>
</tr>
</table>
```

The second cell of the second row of the Center Blocks table contains the Smarty loop that calls on the **Center-Right** blocks. The cell layout is managed through td style id "centerRcolumn". As with the center-center blocks, a Smarty include tag calls on a separate file, `theme_blockcenter_r.html`, for the layout of the center-right blocks.

In summary, the Center Blocks table, which is a nested table, has this basic layout.

Center-Center Blocks	
Center-Left	Center-Right



Theme-Building Tips

- Check Theme-Building Tips for the center-center blocks. They should be very similar.
- Understanding the layout of the Center Blocks table allows you to do some basic XOOPS theme customizations by manipulating the table layout.

2.1.13. Content Table - Module Contents

Line 88-91:

```
<div id="content">
    <{$xoops_contents}>
</div>
</td>
```

This short HTML code, a div, is a very important part of the theme. This dictates where your **module contents** are displayed in relation to the center blocks. The default placement is *below* the Center Blocks table.

Example of module contents are the news articles in the News Module, the Forum table of the NewBB module, and the Contact Form of the Contact Us Module.



Theme-Building Tips

- Just cut and paste this div on top of the Center Blocks table, or more specifically *above the conditional* that calls the Center Block table, if you want the blocks to be below the module contents.

2.1.14. Content Table - Right Column

Line 93-106:

```
<{if $xoops_showrblock == 1}>
    <td id="rightcolumn">
        <!-- Start right blocks loop -->
        <{foreach item=block from=$xoops_rblocks}>
            <{include file="default/theme_blockright.html"}>
        </foreach>
        <!-- End right blocks loop -->
    </td>
</if>
</tr>
</table>
```

This HTML code displays the Right Blocks. It starts with a conditional: if you have right blocks to show (`<{if $xoops_showrblock == 1}>`), then create a cell for that (`<td id="rightcolumn">`), start the Smarty tag loop and include the `theme_blockright.html` file.

As with the Left Blocks, the included file dictates the layout of the Right Blocks.



Theme-Building Tips

- Check Theme-Building Tips for the Left and Center blocks. They should be very similar.

2.1.15. Footer Table

Line 108-114:

```
<table cellpadding="0">
  <tr id="footerbar">
    <td><a href="http://www.xoops.org/" target="_blank"></a></td>
  </tr>
</table>
</body>
</html>
```

The footer table ends the body of the HTML and the `theme.html` file



Theme-Building Tips

- Customize the table as you wish. Add more details (copyright information, contact details) that are relevant to your site.
- Take note that the footer table contains an image (`poweredby.gif`) and not text. The default theme does not use the Footer information from the XOOPS Admin Control Panel. The Footer information can be called and placed in the theme by using the Smarty tag `<{ $xoops_footer }>` within the Footer Table.

2.2. theme_blockleft.html

Line 1-2:

```
<div class="blockTitle"><{ $block.title }></div>
<div class="blockContent"><{ $block.content }></div>
```

This HTML file dictates the general layout of the **Left Blocks** which gives them a consistent look. This is called by the `theme.html` file through a Smarty `<{include}>` tag. The block title and the block content are separate divs.

Remember that **the specific layout of the block contents**, e.g. images used in a block, layout of links in User Menu, custom blocks, are handled by **block templates** and can be managed in the **Templates Administration** of the Admin Control Panel.

The Smarty tags used are:

- `<{ $block.title }>` - This Smarty tag retrieves the Block Title, e.g. User Menu, Login, Main Menu, Who's Online.
- `<{ $block.content }>` - This Smarty tag retrieves the Block Content, e.g. the links in User Menu, the

text and users in Who's Online block.



Theme-Building Tips

- Customize the HTML code as you wish as long as you use the Smarty tags mentioned to place the block title and block contents. A common customization is to convert the divs to a single table with 2 rows.

2.3. theme_blockcenter_c.html

Line 1-6:

```
<div style="padding: 5px;">
  <fieldset>
    <legend class="blockTitle"><{$block.title}></legend>
    <div class="blockContent"><{$block.content}></div>
  </fieldset>
</div>
```

This HTML file dictates the general layout of the **Center-Center Blocks** which gives them a consistent look. This is called by the `theme.html` file through a Smarty `<{include}>` tag.

All center blocks are placed inside a div and enclosed in a fieldset. The block title becomes the legend of that **fieldset**. The block content is placed in a nested div inside the fieldset.

It uses the Smarty tags `<{$block.title}>` and `<{$block.content}>`.



Theme-Building Tips

- The rendering of fieldsets often differ from browser to browser. Although it is possible to change its look through the `style.css` by creating a class for the fieldset, some theme-builders rely on divs or tables to display the block title.
- Take note that the padding of the divs of the center blocks are embedded in the tag rather than linked from the `style.css`. Edit the file directly for changes in the displayed padding of the block.

2.4. theme_blockcenter_l.html

Line 1-6:

```
<div style="padding: 0px 0px 0px 8px;">
  <fieldset>
    <legend class="blockTitle"><{$block.title}></legend>
    <div class="blockContent"><{$block.content}></div>
  </fieldset>
</div>
```

This HTML file dictates the general layout of the **Center-Left Blocks** which gives them a consistent look. This is called by the `theme.html` file through a Smarty `<{include}>` tag.

As with the center-center blocks, the center-left blocks are placed inside a div and enclosed in a fieldset. The block title becomes the legend of that **fieldset**. The block content is placed in a nested div inside the fieldset

It uses the Smarty tags `<{$block.title}>` and `<{$block.content}>`.



Theme-Building Tips

- The rendering of fieldsets often differ from browser to browser. Although it is possible to change its look through the `style.css` by creating a class for the fieldset, some theme-builders rely on divs or tables to display the block title.
- Take note that the padding of the divs of the center blocks are embedded in the tag rather than linked from the `style.css`. Edit the file directly for changes in the displayed padding of the block.

2.5. theme_blockcenter_r.html

Line 1-6:

```
<div style="padding: 0px 5px 0px 0px;">
  <fieldset>
    <legend class="blockTitle"><{$block.title}></legend>
    <div class="blockContent"><{$block.content}></div>
  </fieldset>
</div>
```

This HTML file dictates the general layout of the **Center-Right Blocks** which gives them a consistent look. This is called by the `theme.html` file through a Smarty `<{include}>` tag.

As with the center-center and center-left blocks, the center-right blocks are placed inside a div and enclosed in a **fieldset**.

It uses the Smarty tags `<{$block.title}>` and `<{$block.content}>`.



Theme-Building Tips

- The rendering of fieldsets often differ from browser to browser. Although it is possible to change its look through the `style.css` by creating a class for the fieldset, some theme-builders rely on divs or tables to display the block title.
- Take note that the padding of the divs of the center blocks are embedded in the tag rather than linked from the `style.css`. Edit the file directly for changes in the displayed padding of the block.

2.6. theme_blockright.html

Line 1-2:

```
<div class="blockTitle"><{$block.title}></div>
<div class="blockContent"><{$block.content}></div>
```

This HTML file dictates the general layout of the **Right Blocks**. If the conditional to show right blocks are true, the `theme.html` file calls this file through a Smarty `<{include}>` tag.



Theme-Building Tips

- Customize the HTML code as you wish as long as you use the Smarty tags mentioned to place the block title and block contents. A common customization is to convert the divs to a single table with 2 rows.

Chapter 3. CSS Files

3.1. style.css

The `style.css` Cascading Style Sheet is the main style sheet of a theme. This is loaded in all user-side pages, be it the Home page, the News module or the View Account page.

Line 1-17

```
body {color: black; background: white; margin: 0; padding: 0;}

table {width: 100%; margin: 5; padding: 5; font-size: small}
table td {padding: 0; border-width: 0; vertical-align: top; font-family:
Verdana, Arial, Helvetica, sans-serif;}

a {color: #666666; text-decoration: none; font-weight: bold;
background-color: transparent;}
a:hover {color: #ff6600;}

h1 {}
h2 {}
h3 {}
h4 {}
h5 {}
ul { margin: 2px; padding: 2px; list-style: decimal inside; text-align: left;}
li { margin-left: 2px; list-style: square inside; color: #2F5376}

input.formButton {}
```

These load the general styles to page elements. As can be seen, this is where the body and link attributes of the pages are determined.

Line 19-31:

```
.item {border: 1px solid #cccccc;}
.itemHead {padding: 3px; background-color: #2F5376; color: #FFFFFF;}
.itemInfo {text-align: right; padding: 3px; background-color: #efefef}
.itemTitle a {font-size: 130%; font-weight: bold; font-variant: small-caps;
color: #ffffff; background-color: transparent;}
.itemPoster {font-size: 90%; font-style:italic;}
.itemPostDate {font-size: 90%; font-style:italic;}
.itemStats {font-size: 90%; font-style:italic;}
.itemBody {padding-left: 5px;}
.itemText {margin-top: 5px; margin-bottom: 5px; line-height: 1.5em;}
.itemText:first-letter {font-size: 133%; font-weight: bold;}
.itemFoot {text-align: right; padding: 3px; background-color: #efefef}
.itemAdminLink {font-size: 90%;}
.itemPermaLink {font-size: 90%;}
```

These are style classes specific for the News module and how the News articles are rendered.

Line 33:

```
th {background-color: #2F5376; color: #FFFFFF;
padding : 2px; vertical-align : middle;
font-family: Verdana, Arial, Helvetica, sans-serif;}
```

This is a general style attribute for table headers. This is important to consider when tables are used in blocks and modules, e.g. forums, recent posts block, contact us.

Line 35-36:

```
td#headerbanner {width: 100%; background-color: #2F5376;
```

```
vertical-align: middle; text-align:center;}
td#headerbar {border-bottom: 1px solid #dddddd;
background-image: url(hbar.gif);}
```

These style ids are specific for the **Header Table**. Note that the **header bar image** (`hbar.gif`) is called from the `style.css` as a background image and not directly from the `theme.html`.

Line 38-41:

```
td#leftcolumn {width: 170px; border-right: 1px solid #cccccc;
font-size:12px;}
td#leftcolumn th {background-color: #2F5376; color: #FFFFFF;
vertical-align: middle;}
td#leftcolumn div.blockTitle {padding: 3px; background-color: #dddddd;
color: #639ACE; font-weight: bold;}
td#leftcolumn div.blockContent {padding: 3px; line-height: 120%;
line-height: 120%;}
```

These style ids and classes are specific for the **Left Column**. Note the style attributes used for the **Left Block Titles** and **Left Block Content**.

Line 44-57:

```
td#centercolumn {font-size: 12px;}

td#centercolumn th {background-color: #2F5376; color: #FFFFFF;
vertical-align: middle;}
td#centerCcolumn {padding: 0px 3px 1px 3px;}
td#centerCcolumn legend.blockTitle {padding: 3px; color: #639ACE;
font-weight: bold; margin-top: 0px; margin-right: 0px; margin-left: 0px;}
td#centerCcolumn div.blockContent {border-left: 1px solid #cccccc;
border-right: 1px solid #cccccc; border-bottom: 1px solid #dddddd;
padding: 3px; margin-right: 0px; margin-left: 0px; margin-bottom: 2px;
line-height: 120%;}

td#centerLcolumn {width: 50%; padding: 0px 3px 0px 0px;}
td#centerLcolumn legend.blockTitle {padding: 3px; color: #639ACE;
font-weight: bold; margin-top: 0px;}
td#centerLcolumn div.blockContent {border-left: 1px solid #cccccc;
border-right: 1px solid #cccccc; border-bottom: 1px solid #dddddd;
padding: 3px; margin-left: 3px; margin-right: 2px;
margin-bottom: 2px; line-height: 120%;}

td#centerRcolumn {width: 50%; padding: 0px 3px 0px 0px;}
td#centerRcolumn legend.blockTitle {padding: 3px; color: #639ACE;
font-weight: bold; margin-top: 0px;}
td#centerRcolumn div.blockContent {border-left: 1px solid #cccccc;
border-right: 1px solid #cccccc; border-bottom: 1px solid #dddddd;
padding: 3px; margin-left: 2px; margin-right: 3px;
margin-bottom: 2px; line-height: 120%;}
```

These are the style ids and classes of the all the center blocks: the **center-center blocks** (`centerCcolumn`), the **center-left block** (`centerLcolumn`) and the **center-right blocks** (`centerRcolumn`).

Please note that the font-size of the center blocks and module content is determined here (`centercolumn`).

Line 59:

```
div#content {text-align: left; padding: 8px;}
```

This is the specific div that contains the module contents. The text-align attributes and padding are defined here.

Line 61-64:

```
td#rightcolumn {width: 170px; border-left: 1px solid #cccccc;
font-size:12px;}
```

```
td#rightcolumn th {background-color: #2F5376; color: #FFFFFF;
  vertical-align: middle;}
td#rightcolumn div.blockTitle {padding: 3px; background-color: #dddddd;
  color: #639ACE; font-weight: bold;}
td#rightcolumn div.blockContent {padding: 3px; line-height: 120%;}
```

These style ids and classes are specific for the **Right Column**, if it is to be displayed. Note the style attributes used for the **Right Block Titles** and **Right Block Content**.

Line 66:

```
tr#footerbar {text-align:center; background-image: url(hbar.gif);}
```

This is the id used for the **Footer Table**. Take note that the background image is the same with the Header Bar. It is declared here in the `style.css` and not in the `theme.html`.

Line 68-72:

```
td#mainmenu a {background-color: #e6e6e6; display: block;
  margin: 0; padding: 4px;}
td#mainmenu a:hover {background-color: #ffffff;}
td#mainmenu a.menuTop {padding-left: 3px; border-top: 1px solid silver;
  border-right: 1px solid #666666; border-bottom: 1px solid #666666;
  border-left: 1px solid silver;}
td#mainmenu a.menuMain {padding-left: 3px; border-right: 1px solid #666666;
  border-bottom: 1px solid #666666; border-left: 1px solid silver;}
td#mainmenu a.menuSub {padding-left: 9px; border-right: 1px solid #666666;
  border-bottom: 1px solid #666666; border-left: 1px solid silver;}
```

These are the styles used for the **Main Menu** block. As can be seen, the `style.css` differentiates between the topmost link, the main links and the submenu links.

Line 74-77:

```
td#usermenu a {background-color: #e6e6e6; display: block;
  margin: 0; padding: 4px; border-right: 1px solid #666666;
  border-bottom: 1px solid #666666; border-left: 1px solid silver;}
td#usermenu a:hover {background-color: #ffffff;}
td#usermenu a.menuTop {border-top: 1px solid silver;}
td#usermenu a.highlight {background-color: #fcc;}
```

These are the styles used for the **User Menu** block. The last line (`a.highlight`) states the color of the **Inbox** link when there are private messages for the user.

Line 81-87:

```
.outer {border: 1px solid silver;}
.head {background-color: #c2cdd6; padding: 5px; font-weight: bold;}
.even {background-color: #dee3e7; padding: 5px;}
.odd {background-color: #E9E9E9; padding: 5px;}
.foot {background-color: #c2cdd6; padding: 5px; font-weight: bold;}
tr.even td {background-color: #dee3e7; padding: 5px;}
tr.odd td {background-color: #E9E9E9; padding: 5px;}
```

These are styles used for tables. The odd and even values are the ones used alternately in most **XOOPS** tables to show the different rows and columns.

Line 89-91:

```
div.errorMessage { background-color: #FFCCCC; text-align: center;
  border-top: 1px solid #DDDDFF; border-left: 1px solid #DDDDFF;
  border-right: 1px solid #AAAAAA; border-bottom: 1px solid #AAAAAA;
  font-weight: bold; padding: 10px;}
div.confirmMsg { background-color: #DDFFDF; color: #136C99;}
```



```

text-align: center; border-top: 1px solid #DDDDFF;
border-left: 1px solid #DDDDFF; border-right: 1px solid #AAAAAA;
border-bottom: 1px solid #AAAAAA; font-weight: bold; padding: 10px;}
div.resultMsg { background-color : #CCCCCC; color: #333333;
text-align: center; border-top: 1px solid silver;
border-left: 1px solid silver; font-weight: bold;
border-right: 1px solid #666666; border-bottom: 1px solid #666666;
padding: 10px;}

```

These are styles used for Error and Confirmation Messages produced by the system. There is almost no reason to customize these.

Line 93-94:

```

div.xoopsCode { background: #FFFFFF; border: 1px inset #000080;
font-family: "Courier New",Courier,monospace; padding: 0px 6px 6px 6px;}
div.xoopsQuote { background: #FFFFFF; border: 1px inset #000080;
font-family: "Courier New",Courier,monospace; padding: 0px 6px 6px 6px;}

```

These are styles used for the **Codes** and **Quotes** found in the NewBB module or the module comments.

Line 97-108:

```

.comTitle {font-weight: bold; margin-bottom: 2px;}
.comText {padding: 2px;}
.comUserStat {font-size: 10px; color: #2F5376; font-weight:bold;
border: 1px solid silver; background-color: #ffffff; margin: 2px;
padding: 2px;}
.comUserStatCaption {font-weight: normal;}
.comUserStatus {margin-left: 2px; margin-top: 10px; color: #2F5376;
font-weight:bold; font-size: 10px;}
.comUserRank {margin: 2px;}
.comUserRankText {font-size: 10px;font-weight:bold;}
.comUserRankImg {border: 0;}
.comUserName {}
.comUserImg {margin: 2px;}
.comDate {font-weight: normal; font-style: italic; font-size: smaller}
.comDateCaption {font-weight: bold; font-style: normal;}

```

These are styles used in the NewBB module and module comments to show information about the user. These information are usually found under the username when viewing a forum post.

3.2. styleMAC.css

Line 1-3:

```

@import url(style.css);
/* Very short Mac-specific additions/changes here (if
any) */

```

These files are used to improve CSS rendering between different browsers. This file allows fine-tuning of CSS definitions for **Mac** browsers.

3.3. styleNN.css

Line 1-3:

```

@import url(style.css);
/* Very short Gecko-specific additions/changes here (if
any) */

```

These files are used to improve CSS rendering between different browsers. This file allows fine-tuning of CSS

definitions for **Netscape** browsers, including **Mozilla**, **Mozilla Firebird** and **Mozilla Firefox**.



Chapter 4. Images

4.1. logo.gif



This is the image found inside the **Header Table**.

4.2. hbar.gif



This is the image used for both the **Header Bar** and the **Footer Table**.

4.3. poweredby.gif

A horizontal bar with a light gray background and a thin border, containing the text "POWERED BY XOOPS (C) 2002 THE XOOPS PROJECT" in a small, dark font.

This is the image used as the footer information in the **Footer Table**.



Chapter 5. Basic Theme-Building Tips

5.1. Installing a Theme

The most common way of installing a theme is through FTP. If your theme files (`theme.html`, `style.css`, images) are already inside a folder, upload that folder inside the Theme Directory (`. . themes/`). If they are not in a folder, create a new folder inside the Theme Directory and upload the theme files to that new folder. Remember that **the name of the folder** becomes **the name of the theme** from within the XOOPS Admin Control Panel.

Once you've uploaded the theme and the theme files, go back to **Administration Menu -> System Administration -> Preferences -> General Settings** and choose your theme from the "Selectable themes" dropdown box. Save changes.

An extra step is to delete all cached files from within the `templates_c/` folder using an FTP client. It is safe to delete all files in that folder, but it is prudent to leave the `index.html` file in there. This extra step ensures that you are looking at your new theme instead of cached files of the old theme.

If you made changes to your theme after it is installed, go to the **Administration Menu -> System Administration -> Preferences -> General Settings** and set "Update module template .html files from themes/your theme/templates directory?" to YES. This will allow XOOPS to look for new or changed theme files in the Theme Directory.

Go to your home page and check out your new theme. If everything is okay, go back to **Administration Menu -> System Administration -> Preferences -> General Settings** and set "Update module template .html files from themes/your theme/templates directory?" to NO.

5.2. Smarty Debugging

A useful tool for Theme Building is Smarty Debugging. Go to **Administration Menu -> System Administration -> Preferences -> General Settings** and set "Debug mode" to **Smarty Templates Debug**. Remember to do this only for a local setup or a non-production site since a pop-up will be created with a list of Smarty variables available.

It is a general rule that most Smarty variables listed by the Smarty Debug popup when the home page is loaded are available for use inside the `theme.html`. In simpler terms, when you load the home page, most Smarty variables from the popup can be used throughout the site as a Smarty tag embedded in the `theme.html` file.



Index