



## Beech - Theming/Skinning MindTouch

**developer.mindtouch.com is a community contributed help site for MindTouch Core only. Supported MindTouch TCS documentation can be found at [help.mindtouch.us](http://help.mindtouch.us)**

**Using MindTouch TCS (SAAS)? Find MindTouch TCS skinning tutorials at [help.mindtouch.us](http://help.mindtouch.us)**

- Changing the background color of your MindTouch Help Center
- Hide the left hand navigation on a MindTouch article
- Hiding a section on a MindTouch article

**Note: This documentation only pertains to MindTouch Core**

**9/7/10** - MindTouch 2010 now ships with MindTouch 2010. BeechBeta is the MindTouch supported and much improved version of beech. Documentation will be updated to support BeechBeta.

**2/23/10** - New [beech.zip](#) available for download. Added PageRss(),

### Comments:

There are **38** comments on this page. [Contribute now](#)

### Tutorial:



### Views:

### Beech Resources

- [Beech Design Forge](#)
  - [Colombia](#)
  - [Mediterraneo](#)
  - [Windjammer](#)
- [Beech PHP Function Reference](#)
  - [Beech PHP Function Reference - Message Function](#)
  - [Beech PHP Function Reference - Page Functions](#)
  - [Beech PHP Function Reference - Site Functions](#)
  - [Beech PHP Function Reference - Title Functions](#)
  - [Beech PHP Function Reference - User Functions](#)
- [FAQ: Beech Theme](#)
- [Step By Step Tutorial to Skin/Theme MindTouch using Beech](#)

## Goals & Objectives

1. Provide much more granular control over the layout
2. Make PHP skinning functions understandable
3. Provide a more bare-bones HTML layout for designers to customize as they see fit

4. Require designers to modify just one CSS file
5. Make navigating the theme file system more intuitive

## Implemented Changes

### 1. Separated most of the PHP & HTML

Formerly only one markup file existed for each theme and its name matched that of the theme, ace.php, fiesta.php, deuce.php. Each of the aforementioned files were saturated with markup, PHP and JavaScript, much of which was not important to a designer. Additionally because these files were so cluttered visually parsing them was difficult, even for the trained eye. To solve this problem I moved the HTML content from the markup file to a new file called HTML.php. The new separated file is intended to provide designers with a more accurate and manageable skinning experience. I also located HTML.php inside the skin folder to provide more granular control over each skin. With this change now each skin can have custom markup that is independent of other skins.

### 2. Restructured the theme files

When creating this new theme I wanted to make accessing its files more intuitive. With the other themes there are multiple files in numerous folders that are inserted into the skin. Custom CSS modifications were always stacked on top of a large underlying layer of MindTouch prescribed CSS. This approach limited flexibility and forced a very sculpted look & feel upon designers. In order to solve this problem I removed the base CSS files (`_common.css`, `_style.css`) in exchange for one CSS file, `_style.css`, which resides in the skin folder. I also shifted the `_print.css`, `html.php` and all images to the skin level. Again, this provides much more granular control over the look & feel of each individual skin and also eliminates any MindTouch defined CSS dependencies.

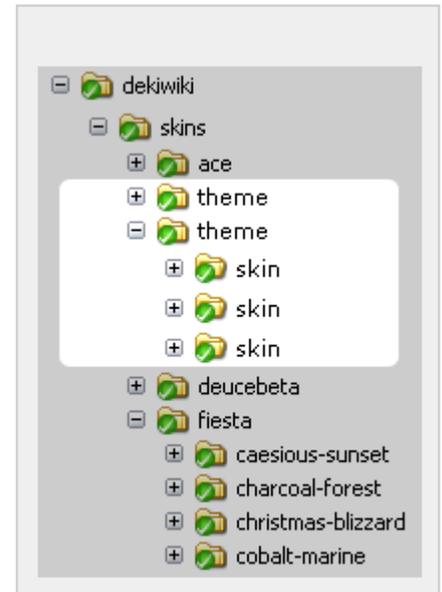
### 3. Renamed all the PHP skinning functions

When I set out to change the theming process I immediately noticed that the PHP skinning functions in each markup file were inconsistent and could be challenging to a non-developer. For instance there were references to `wfMsg()`, `$this->html()`, `$this->haveData()`, `$wgTitle`, `$wgUser` and many more. While programmatically these functions all make sense they add an unnecessary burden on a designer. As an alternative I gave these functions new pseudo names that follow a simple pattern and consistent naming schema. For instance:

1. `$this->SiteLogo();`
2. `$this->PageTags();`
3. `$this->PageEdit();`
4. `$this->PageAdd();`
5. `$this->PageTitle();`

### 4. Consolidated CSS files

Initially when we built MindTouch themes we planned for CSS modifications to be made as an addition to the existing CSS. While this enabled very rapid theme development it also limited the flexibility of the design and greatly raised the skill level required to completely customize MindTouch. Additionally it also made many assumptions about how MindTouch should look and generally limited the scope of customization to aesthetic CSS such as colors, fonts, etc. As a solution I decided to completely remove the base CSS files (`_common.css` and `_style.css`) and only reference the `_style.css` in the skin folder. I will add, however, that `_content.css` will remain, as content styling has proven to be the most challenging portion of creating a new



MindTouch skin. The `_content.css` file is pre-populated with default content CSS but can be modified on a per skin basis.

### 5. **Eliminated unnecessary structure**

In order to make theming more flexible there were some deeper consideration required for the PHP skinning functions. Whereas the older PHP skinning functions contained a large amount of predefined HTML I opted to remove most of this HTML and place it directly in `HTML.php`. This may seem fairly useless but it does shift the control of the HTML into

the hands of the designer. For instance I removed extra wrapper `<div>`, `<li>` and `<span>` elements that were added for CSS identification or to meet some assumed function. Ultimately now you can decide if your logout button is in a dropdown list, wrapped in `<div class="logout-wrapper">` or just by itself.

