

Differences from Builder 1.x

This document explains the changes made to menus and their notation between Theme Builder 1.x and 2.x...

Theme Builder 1.x Menu Notation

In Theme Builder 1, menus were defined as simple unordered lists like the one shown below:

```
* [Builder]
** [Builder 1.x]
** [Builder 2.x]
```

The notation above would produce a menu something like the following if placed on a normal page within your space:

- [Documentation]
 - [Builder 1.x](#)
 - [Builder 2.x](#)

And if placed in the "Custom pre-menu" or "Custom post-menu" sections of the [Theme Configuration](#) screen it would result in a menu bar at the top of the screen similar to this:

Unknown macro: {menubar}
Unknown macro: {wikimenu}

- [Builder]
 - [Builder 1.x](#)
 - [Builder 2.x](#)

Theme Builder 2.x Menu Notation

In Theme Builder 2 and above you can still use the simple unordered list notation, however it must be wrapped in the [menubar macro](#) and [wikimenu macro](#) as shown below:

```
{menubar}
{wikimenu}
* [Builder]
** [Builder 1.x]
** [Builder 2.x]
{wikimenu}
{menubar}
```

menubar and wikimenu

The [menubar macro](#) turns an unordered (bullet) list of links in to a functional menu. The [wikimenu macro](#) processes a normal Confluence wiki notation unordered list to ensure it's in the correct format for processing by the [menubar macro](#).

The neat thing about this is that you can now put the menu anywhere - for example, you put it in a sidebar, the footer or even on a wiki page (as shown above).

Note: The [menubar macro](#) can only be used in locations that are themed with the Theme Builder plugin (as it includes the JavaScript and CSS files required to make the menus work).

menuitem, menuseparator, menu, submenu, sub-submenu

While the wiki notation is often really easy to work with, there are some downsides - most notably when a user does not have access to the item you are linking to which results in a "dead" (non-clickable) menu item. As such, we highly recommend using the new [Menu Macros](#) when defining menus in Theme Builder 2:

```
{menubar}
  {menuitem}[Home]{menuitem}
  {menuseparator}
  {menu}[Builder]
    {menuitem}[Builder 1.x]{menuitem}
    {menuitem}[Builder 2.x]{menuitem}
  {menu}
{menubar}
```

We've indented the items above to make it easier to see the hierarchical structure of the menu - this has no effect on the actual menu, so it's optional, but we highly recommend it as it makes your menu notation easier to read.

The result:

```
Unknown macro: {menubar}
Unknown macro: {menuitem}
[Home]
Unknown macro: {menuseparator}
Unknown macro: {menu}
[Builder]
Unknown macro: {menuitem}
Builder 1.x
Unknown macro: {menuitem}
Builder 2.x
```

The [menuitem macro](#) replaces the *'s in the wiki notation and to indent the list (create a submenu) the [menu macro](#) is used (and for subsequent indentation levels the [submenu macro](#) and [sub-submenu macro](#) are used).

The [menuitem macro](#) has a very useful feature - it will hide any dead links in your menus automatically.

The [menuseparator macro](#) replaces the ---- used to create a divider in the menu. Should two menu separators be found next to each other (eg. because the menuitem macro has hidden inactive links between them) the surplus separator(s) will automatically be removed.

Likewise, if an entire menu or sub-menu is found to be empty, it will be removed.

menulink

The [menulink macro](#) provides an easy way to link to all common features within Confluence. For example, instead of using wiki notation to define the link to the home page like this:

```
{menuitem}[Home]{menuitem}
```

You can use the [menulink macro](#) as follows:

```
{menuitem}{menulink:home}Home Page{menulink}{menuitem}
```

The key benefit in this scenario is that it doesn't matter what the home page is called, or even if it's renamed at a later date, the [menulink macro](#) will still find it and link to it.

Another key benefit of the [menulink macro](#) is that it strictly adheres to Confluence permissions and user privileges and therefore provides a very reliable way of linking to Confluence features.

For more information, see [menulink macro](#).

viewmenu and editmenu

Theme Builder 2 also introduced the [viewmenu macro](#) and [editmenu macro](#) - these allow you to quickly insert the default View and Edit menus anywhere within your menu structure.

Furthermore, you can now fully customise the View and Edit menus to completely customise them or separate items out in to different menus, etc. For more information see [Theme Configuration 2.x](#).