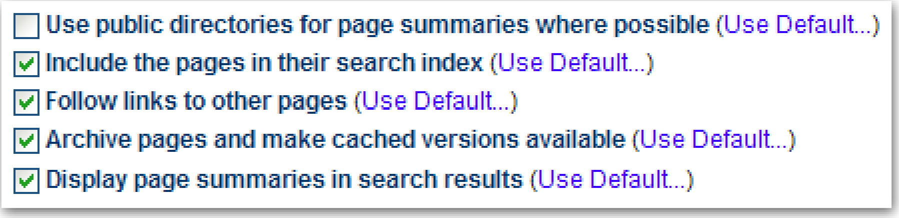
Search Engine Optimisation (SEO)

Theme Builder is the only theme that lets you optimise Confluence for Internet search engines...

Control the Robots!

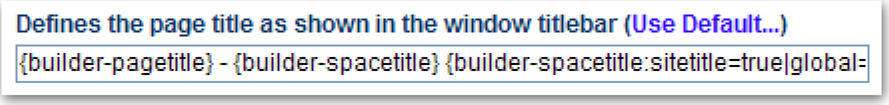
Theme Builder now allows you to easily control search engine robots/crawlers.



☐ Use public directories for page summaries where possible (Use Default...)
☑ Include the pages in their search index (Use Default...)
☑ Follow links to other pages (Use Default...)
☑ Archive pages and make cached versions available (Use Default...)
☑ Display page summaries in search results (Use Default...)

Document Title Customisation

Whether you want a static title, a dynamic title or a mixture of both, Theme Builder gives you complete control:



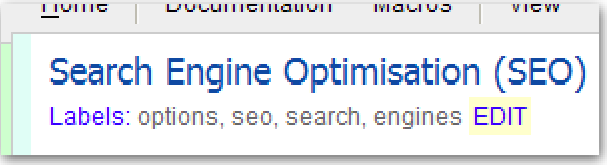
Defines the page title as shown in the window titlebar (Use Default...)

{builder-pagetitle} - {builder-spacetitle} {builder-spacetitle:sitetitle=true|global=

The title can be just static text or even include macros to output the site title, space title and page title.

Automatic Keywords

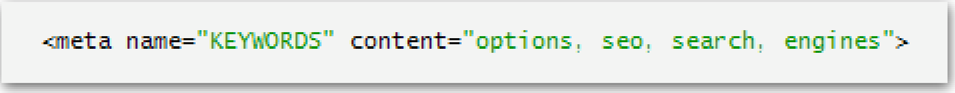
Add labels to your wiki content or blog posts:



Search Engine Optimisation (SEO)

Labels: options, seo, search, engines EDIT

And enable the option to automatically create meta keywords in the HTML:



```
<meta name="KEYWORDS" content="options, seo, search, engines">
```

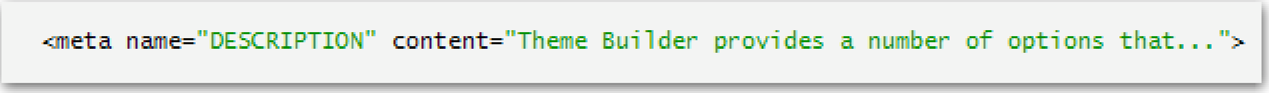
This is particularly useful if you have enterprise search appliances or web services which look for specific terms in your metadata vocabulary.

Customisable Page Summary

You can define the page description by simply wrapping content in the excerpt macro:

```
{excerpt}Theme Builder provides a number of options that...{excerpt}
```

Will result in the following being added to the HTML:

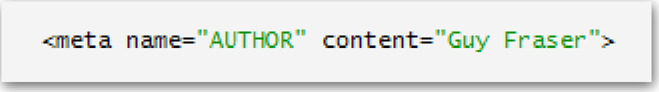


```
<meta name="DESCRIPTION" content="Theme Builder provides a number of options that...">
```

If you don't specify an excerpt, the wiki will generate one automatically.

Automatic Author

Easily output the Author meta tag based on the person who created the page:



```
<meta name="AUTHOR" content="Guy Fraser">
```

Deferred Panel Loading

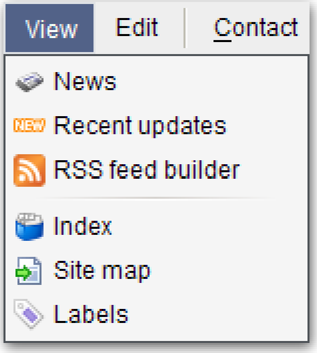
Demote the importance of boilerplate content in panels such as the header, footer and sidebars by optionally loading any of them after the main page content:

A search engine will see a very different structure to the page than an end-user - finally a win, win situation!

You can also move in-page content to other locations at runtime - this allows you to put less important text further down the page and then move it further up the page or in to sidebars, etc., when rendered in the browser.

Search-Friendly Navigation

Builder has some pretty extreme forms of navigation, such as menus and tree-views, and we've gone to great lengths to make sure that navigation works well with search engines:



```
...
<li class="menuitem">View
 <ul class="submenu">
  <li class="menuitem">
   <a href="/pages/viewrecentblogposts.action?key=Builder">News</a>
  </li>
  ...
  <li class="menuitem">
   <a href="/labels/listlabels-heatmap.action?key=Builder">Labels</a>
  </li>
 </ul>
</li>
...
```

Menus are output in to the HTML as an unordered list of links, not as JavaScript, ensuring that search engines can understand the menus.

The tree-view, on the other hand, is purely JavaScript to prevent loads of non-related links from getting prioritised by the search engines.

Navigation macros allow tooltips to be specified (alt and title attributes) to further improve quality of indexing by some search engines.

Intelligent "nofollow" Directives

There's some stuff you just don't want Internet search engines to index and this is the only theme understands that requirement.

As well as outputting `nofollow` directives automatically for certain links (such as an Edit link), you can optionally add the directive to any link generated by the menulink macro and compound-menuitem macro (the macros that are used for almost all of the navigation).

Cleaner, More Semantic HTML

We've spent even more time cleaning up the HTML output by the theme - we've simplified it, we've validated it and we've made it more semantic than ever before.

We've reduced HTML clutter, such as inline scripts and styles, to an absolute minimum (some "cruft" is still required by Confluence) leaving the vast majority of the markup dedicated to content, navigation and basic layout.