

## ChiliProject - Feature # 169: [PATCH] hiding form pages from search engines

<b>Status:</b>	Closed	<b>Priority:</b>	Normal
<b>Author:</b>	Yuki Sonoda	<b>Category:</b>	User interface
<b>Created:</b>	2011-02-10	<b>Assignee:</b>	Eric Davis
<b>Updated:</b>	2011-02-17	<b>Due date:</b>	
<b>Remote issue URL:</b>	<a href="http://www.redmine.org/issues/7582">http://www.redmine.org/issues/7582</a>		
<b>Affected version:</b>			
<b>Description:</b>	<p>Form pages like /issues/new are not worth to be indexed by search engines. And moreover it is sometimes confusing for visitors from search engine. When you have a question about chiliproject and you search about it, what can you do if /issues/new appears?</p> <p>It happens when these form pages are opened for anonymous user. It actually happened at redmine.ruby-lang.org once. So I wrote the attached patch. This patch adds a meta element as follows in some pages:</p> <pre>&lt;pre&gt; &lt;meta name="ROBOTS" content="NOINDEX,FOLLOW,NOARCHIVE" /&gt; &lt;/pre&gt;</pre>		

### Associated revisions

**2011-02-14 03:17 am - Eric Davis**

[#169] Add a ROBOTS meta tag to several forms to hide from web spiders

Based on the patch by Yuki Sonoda

### History

**2011-02-10 09:52 am - Felix Schäfer**

I guess having something like @\*/new@ in the robots.txt wouldn't work, would it?

**2011-02-10 01:07 pm - Yuki Sonoda**

According to <http://www.robotstxt.org/robotstxt.html>, robots.txt does not support glob. So we can not expect \*/new works fine.

**2011-02-10 11:36 pm - Eric Davis**

I think this is a good idea. I'd like to improve on it a little bit though by making the robot\_exclusion\_tag take options for the content section (e.g. @robot\_exclusion\_tag("NOINDEX,FOLLOW,NOARCHIVE")@ or @robot\_exclusion\_tag("NOINDEX,NOFOLLOW")@). Then we (or plugins) could have more control over the indexing options for each page.

Thoughts?

**2011-02-11 07:30 am - Felix Schäfer**

Eric Davis wrote:

> Thoughts?

What about making @NOINDEX,FOLLOW,NOARCHIVE@ the default and calling the method with any collection of @(NO)SOMETHING@ overrides the default for that keyword?

**2011-02-11 07:01 pm - Eric Davis**

This was my idea. It let us have more control of what the actual content is in case the meta tag allows other values later.

```
<pre><code class="ruby">
def robot_exclusion_tag(content="NOINDEX,FOLLOW,NOARCHIVE")
  "<meta name='ROBOTS' content=#{content} />"
end
```

```
end
</code></pre>
```

**2011-02-11 09:31 pm - Felix Schäfer**

Eric Davis wrote:

> This was my idea. It let us have more control of what the actual content is in case the meta tag allows other values later.

No, I meant having @NOINDEX,FOLLOW,NOARCHIVE@ be the default, and if you call it with @INDEX@ to get @INDEX,FOLLOW,NOARCHIVE@. I just realized that's overengineering it though, I like your proposal :-)

**2011-02-11 11:54 pm - Eric Davis**

- Target version set to 1.1.0 *â€” Bell*

- Assignee set to Eric Davis

Yea I thought about doing keywords too but then we would have to maintain a list of valid ones. Hence the idea of just using a simple string.

I'll add and modify this patch. I think it's minor enough for 1.1.0.

**2011-02-14 02:20 am - Eric Davis**

- Status changed from Open to Ready for review

I've modified Yuki Sonoda's patch and the code is ready for review.

<https://github.com/chiliproject/chiliproject/pull/7>

**2011-02-14 07:05 am - Felix Schäfer**

Looks good to me. I'll merge it by the time I'm around a more stable connection if you haven't done so until then.

**2011-02-14 09:33 am - Holger Just**

I still like Felix' idea of having defaults and being able to gradually overwrite them. This could be done like this:

```
<pre><code class="ruby">
# Add a HTML meta tag to control robots (web spiders)
#
# @param [optional, String] changed content of the ROBOTS tag.
# defaults to no index, follow, and no archive
def robot_exclusion_tag(content="")
  default_content = { "INDEX" => "NO",
                    "FOLLOW" => "",
                    "ARCHIVE" => "NO" }

  args = content.upcase.split(",").inject({}) do |args, arg|
    value = arg.gsub(/^(NO)/, "")
    args[value] = $1 || ""
  end
  default_content.merge(args).collect{ |k, v| v+k }.join(",")
end
</code></pre>
```

**2011-02-14 11:07 pm - Eric Davis**

Holger Just wrote:

> I still like Felix' idea of having defaults and being able to gradually overwrite them.

Just seems like a lot of code to me that might not be used that often.

**2011-02-15 06:28 am - Felix Schäfer**

Eric Davis wrote:

> Just seems like a lot of code to me that might not be used that often.

That's what I meant with "don't overengineer it" ;-) I think the simple version is fine, avoiding to have to write it all out for those few times you need other params is not worth it.

**2011-02-17 01:12 am - Eric Davis**

- *Status changed from Ready for review to Closed*

Merged into master for 1.1.0. Thank you for the patch Yuki Sonoda.

## Files

---

robot_exclusion.patch	2 kB	2011-02-10	Yuki Sonoda
-----------------------	------	------------	-------------