

Redmine - Feature #2084

Customizable "protocols" to generate external links

2008-10-24 11:56 - Erwan Ducroquet

Status: New	Start date: 2008-10-24
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category: Wiki	Estimated time: 0.00 hour
Target version:	
Resolution:	
Description	
<p>Allows to define "prototypes" along with an URL prefix, so it will be easy in markup to reference external resource without needing a copy/paste of a full URL</p> <p>Example with proto='mywiki' ; URL prefix="http://wiki.company.com/display/space/" :</p> <p>markup : "Check This":mywiki:somewikipage formatted into : Check This</p> <p>Other examples :</p> <p>"Great language":wp:Ruby_(Programing_Language) Great language</p> <p>:wp:Redmine Redmine</p> <p><i>note:</i> also same feature asked for CustomFields</p>	
Related issues:	
Related to Redmine - Feature # 2083: CustomField of type "external-link-to" w...	Closed 2008-10-24
Related to Redmine - Feature # 2953: application launch URLs not rendered as...	New 2009-03-11

History

#1 - 2008-10-24 11:57 - Erwan Ducroquet

see also #2083

#2 - 2008-10-24 14:48 - Erwan Ducroquet

For info : I've currently hardcoded such feature by adding a method in wiki_formatting.rb, but it's not very pretty ...

```
##- Erwan Ducroquet 2008-10-23 generate links depending on proto
def inline_specific_link_do(all, start, proto, url, title)
  case proto
  when 'stp'
    "#{start}<a class='external' href='http://wiki.fr.world.socgen/display/STP/#{url}'>#{title}</a>"
```

```

when 'ccl'
"#{start}<a class='external' href='http://srvparstp20.fr.world.socgen/cc/view,label,#{url}'>#{title}</a>"
when 'wp'
"#{start}<a class='external' href='http://en.wikipedia.org/wiki/#{url}'>#{title}</a>"
when 'google'
"#{start}<a class='external' href='http://www.google.com/search?q=#{url}'>#{title}</a>"
when 'file'
"#{start}<a class='external' href='file://#{url}'>#{title}</a>"
else
all
end
end

```

#-- Erwan Ducroquet 2008-10-23 add own way to handle specific URL prototypes : STP wiki links, files, ClearCase source, Rules, ext Reviews

#-- format is :repository:resource or "title":repository:resource

```

def inline_specific_link(text)
text.gsub!(/\^[^"\n]+\\"{1}:[a-z]+):([w_\V\\:~!+@()]+)/ do
inline_specific_link_do( $&, ", $2, $3, $1)
end
text.gsub!(/\^[^"\n]+\\"{1}:[a-z]+):([w_\V\\:~!+@()]+)/ do
inline_specific_link_do( $&, ", $1, $2, $2)
end
end

```