**RedMine Information**

<https://www.meta-inf.hu/pages/viewpage.action?pageId=40730668>

**Redmine2confluence-wiki**

[**https://github.com/vile/redmine2confluence-wiki**](https://github.com/vile/redmine2confluence-wiki)

* Execute ExportWiki.export\_all from exportwiki.rb in the redmine environment. This will create set of folders with all the project data
* Edit redmine\_spaces.py file to
  + provide reference to Jira ticket link conversion or another ticketing system (it will replace \g<1> with original redmine ID)
  + configure mapping of redmine projects to confluence spaces. Different projects can be mapped to the same space. in this case conflicting pages will be ignored
* Modify 'import\_confluence.py' to configure:
  + user name
  + password
  + confluence URL
* Run the python script 'import\_confluence.py' in the exported data folder (root level)

## Examples 1:

Exporting a single project by name from Redmine 2 / Rails 3:

$ cd /PATH/TO/redmine

$ script/rails console production

> require("/PATH/TO/redmine2confluence-wiki/export\_wiki.rb")

> ExportWiki.export\_wiki("/tmp", Project.find\_by\_name("PROJECT NAME").wiki)

## Example 2:

This is how we used it with Redmine 2.2.0, Ruby 1.8.7 and Rails 3.2.9.

First I had to find the rails command:

# find / -name rails

Next execute the script:

# cd /PATH/TO/redmine

# RAILS\_ENV=production /var/lib/gems/1.8/bin/rails console

> load "exportwiki.rb"

> ExportWiki.export\_all()

All the pages will be saved under /tmp/redmine.

Move the page to the location of the script.

# mv /tmp/\* .

Next I tried with a test Confluence server. Some pages might be refused by Confluence. I split the file in two till I find the line that causes the problem. Put the line in redmine\_space.py.

I also created a script that generates a SQL script to be run by Postgresql to fix page authors. The process is described here:

<https://confluence.atlassian.com/display/CONFKB/How+to+change+the+creator+of+a+page>

# ./create\_sql\_script.py

This will result in a script called 'fix-author.sql' to be run on the Confluence server.

**Exporting redmine wiki through SQL**

Here's a SQL query to fetch Redmine's wiki data:

SELECT

    wiki\_pages.id AS page\_id,

    wiki\_pages.parent\_id AS parent\_page\_id,

    wiki\_pages.title AS page\_name,

    wiki\_contents.text AS page\_content,

    users.login AS author,

    projects.identifier AS project\_key,

    projects.name AS project\_name

FROM wiki\_pages

LEFT JOIN wiki\_contents ON wiki\_contents.page\_id = wiki\_pages.id

LEFT JOIN users ON wiki\_contents.author\_id = users.id

LEFT JOIN wikis ON wiki\_pages.wiki\_id = wikis.id

LEFT JOIN projects ON wikis.project\_id = projects.id

ORDER BY page\_id ASC;

# Exporting redmine wiki pages

[**http://stbuehler.de/blog/article/2011/06/04/exporting\_redmine\_wiki\_pages.html**](http://stbuehler.de/blog/article/2011/06/04/exporting_redmine_wiki_pages.html)

# I wanted to export all wiki pages of a redmine project as plain text (the original textile code) with a small yaml header:

~/redmine $ RAILS\_ENV=production ./script/console -s

Loading production environment in sandbox (Rails 2.3.11)

Any modifications you make will be rolled back on exit

NOTE: SourceIndex.new(hash) is deprecated; From /usr/lib/ruby/gems/1.8/gems/rails-2.3.11/lib/rails/vendor\_gem\_source\_index.rb:100:in `new'.

>> def export\_text(p)

>> c = p.content\_for\_version(nil)

>> "---\ntitle: #{p.title}\nupdated\_on: #{c.updated\_on}\n---\n\n#{c.text}"

>> end

=> nil

>> def export\_wiki(dir, wiki)

>> dir = dir + "/" + wiki.project.identifier

>> Dir.mkdir(dir)

>> wiki.pages.each { |p|File.open(dir + "/" + p.title + ".textile", "w") { |f| f.write(export\_text(p)) } }

>> true

>> end

=> nil

>> Project.all

=> [#<Project id: 2, name: "Debian Server", ... , identifier: "debianserver", ...

>> export\_wiki("/tmp", Project.find(2).wiki)

=> true

Now “/tmp/debianserver/” contains all the wiki pages.



**The ruby functions without irb shell:**

def export\_text(p)

c = p.content\_for\_version(nil)

"---\ntitle: #{p.title}\nupdated\_on: #{c.updated\_on}\n---\n\n#{c.text}"

end

def export\_wiki(dir, wiki)

dir = dir + "/" + wiki.project.identifier

Dir.mkdir(dir)

wiki.pages.each { |p|File.open(dir + "/" + p.title + ".textile", "w") { |f| f.write(export\_text(p)) } }

true

end

**Bulk Import**

Script to bulk import textile files into Confluence Wiki. Very useful to convert a redmine wiki to Confluence. To export redmine wiki pages to textile you can use <http://stbuehler.de/blog/article/2011/06/04/exporting_redmine_wiki_pages.html> We used this and "<https://bitbucket.org/simbtecnologia/github-to-jira>" to migrate from github/redmine tool…

|  |
| --- |
| [**import\_confluence.py**](https://gist.github.com/tim-jansen/6263586#file-import_confluence-py)  """ |
|  |

|  |
| --- |
| Script to import multiple directories with textile files into Confluence Wikis. Can be used with OnDemand instances. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| To use as redmine migration tool, you need to export wiki pages in textile format. One way is described in: http://stbuehler.de/blog/article/2011/06/04/exporting\_redmine\_wiki\_pages.html |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ~/redmine $ RAILS\_ENV=production ./script/console -s |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| def export\_text(p) |
|  |

|  |
| --- |
| c = p.content\_for\_version(nil) |
|  |

|  |
| --- |
| "#{c.text}" |
|  |

|  |
| --- |
| end |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| def export\_wiki(dir, wiki) |
|  |

|  |
| --- |
| dir = dir + "/" + wiki.project.identifier |
|  |

|  |
| --- |
| Dir.mkdir(dir) |
|  |

|  |
| --- |
| wiki.pages.each { |p|File.open(dir + "/" + p.title + ".textile", "w") { |f| f.write(export\_text(p)) } } |
|  |

|  |
| --- |
| true |
|  |

|  |
| --- |
| end |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Project.all |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| export\_wiki("/tmp", Project.find(2).wiki) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| """ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import sys |
|  |

|  |
| --- |
| import os |
|  |

|  |
| --- |
| import re |
|  |

|  |
| --- |
| from xmlrpclib import ServerProxy, Fault |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| # Connects to confluence server with username and password |
|  |

|  |
| --- |
| site\_URL = "https://yoursite.atlassian.net/wiki/" |
|  |

|  |
| --- |
| server = ServerProxy(site\_URL + "/rpc/xmlrpc") |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| username = "confluence\_username" |
|  |

|  |
| --- |
| pwd = "confluence\_password" |
|  |

|  |
| --- |
| token = server.confluence2.login(username, pwd) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| spaces = { |
|  |

|  |
| --- |
| "your\_redmine\_project\_name":"CONFLUENCECODE" |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| def save(space, filename, pagename): |
|  |

|  |
| --- |
| # Retrives text from a file |
|  |

|  |
| --- |
| f = open(filename, 'r') |
|  |

|  |
| --- |
| content = f.readlines() |
|  |

|  |
| --- |
| content = '\n'.join(content[1:]) |
|  |

|  |
| --- |
| content = re.subn(r'([{}])',r'\\\1',content, flags=re.S)[0] |
|  |

|  |
| --- |
| content = re.subn(r'@(.\*?)@',r'{{\1}}}',content)[0] |
|  |

|  |
| --- |
| content = re.subn(r'<pre>(.\*?)</pre>',r'{code}\1{code}',content, flags=re.S)[0] |
|  |

|  |
| --- |
| f.close() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| # Creates a new page to insert in the new space from text file content |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| content = server.confluence2.convertWikiToStorageFormat(token, content) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| newpage = {"title":pagename, "space":space, "content":content} |
|  |

|  |
| --- |
| server.confluence2.storePage(token, newpage) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| for project\_dir, subdirs, files in os.walk('.'): |
|  |

|  |
| --- |
| try: |
|  |

|  |
| --- |
| project\_name = project\_dir.split('/')[1] |
|  |

|  |
| --- |
| except IndexError: |
|  |

|  |
| --- |
| continue |
|  |

|  |
| --- |
| for filename in files: |
|  |

|  |
| --- |
| if filename[-8:] == '.textile': |
|  |

|  |
| --- |
| pagename = filename[:-8].replace('\_',' ').strip() |
|  |

|  |
| --- |
| else: |
|  |

|  |
| --- |
| continue |
|  |

|  |
| --- |
| try: |
|  |

|  |
| --- |
| server.confluence2.getPage(token,spaces[project\_name],pagename) |
|  |

|  |
| --- |
| except KeyError: |
|  |

|  |
| --- |
| print "Project %s not mapped" % project\_name |
|  |

|  |
| --- |
| break |
|  |

|  |
| --- |
| except Fault: |
|  |

|  |
| --- |
| pass |
|  |

|  |
| --- |
| else: |
|  |

|  |
| --- |
| print "Page %s exists" % pagename |
|  |

|  |
| --- |
| continue |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| print "Saving page %s"% pagename |
|  |

|  |
| --- |
| save(spaces[project\_name], os.path.join(project\_dir,filename),pagename) |
|  |

|  |
| --- |
|  |
|  |

server.confluence2.logout(token)

# Importing data from Redmine

## About importing from Redmine

You can install additional importer that supports Redmine from [Atlassian Marketplace](https://marketplace.atlassian.com/plugins/com.atlassian.jira.plugins.jira-importers-redmine-plugin)

It will work with JIRA Importers Plugin 5.0.2 and above. Requires JIRA 5.0 or above.

Also please refer to this official [documentation](https://confluence.atlassian.com/jira/importing-data-from-redmine-313459706.html).

<https://confluence.atlassian.com/adminjiraserver072/importing-data-from-redmine-829827218.html>

Enable

In your Redmine instance, ensure that REST support is enabled: open **Administration > Settings > Authentication** tab and select the **"Enable REST web service"** check box.