Redmine - Feature #306
Full Text Search of files

2007-03-29 08:49 - Ross Manning

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Search engine	**Estimated time:**	0.00 hour
Target version:	4.1.0		
Resolution:			

Description

It would be great if we could get an option to search the contents of attached files/documents.

Related issues:

Duplicated by Redmine - Feature # 818: Fulltext search include content of files	**Closed**	**2008-03-09**
Duplicated by Redmine - Feature # 4862: Search engine doesn't look inside doc...	**Closed**	**2010-02-17**

History

#1 - 2009-07-17 12:48 - Oleg Lozinskij

Ross Manning wrote:

> *It would be great if we could get an option to search the contents of attached files/documents.*

Any updates on this feature?

Cheers!

#2 - 2009-07-19 00:30 - Jens Goldhammer

Maybe you can use ActAsSolr (http://acts-as-solr.rubyforge.org/) or ActAsFerret (http://rm.jkraemer.net/projects/activity/aaf) for it...

#3 - 2009-11-18 12:53 - S Reid

Has anyone tried the above, or have any other suggestions ?

#4 - 2010-02-17 19:49 - Jean-Philippe Lang

- Category set to Search engine

#5 - 2010-03-03 09:15 - Emmanuel Bastien

FYI, I was pleased with acts_as_xapian features, even if I'm not sure that there is still a maintainer for this project.

#6 - 2013-01-29 15:43 - Anonymous

+100 for this one.

This is a really interesting feature!

#7 - 2013-03-25 20:27 - Dipan Mehta

+1. Very much useful

#8 - 2013-06-07 15:52 - Terence Mill

It's worth to have a look at RSolr whih can easily integrate with an solr instance embedded in jruby.

```
if RUBY_PLATFORM =~ /java/

  require 'rsolr-direct'
  ::RSolr.load_java_libs

  ::Sunspot.session.instance_variable_set(:@connection,
    ::RSolr.connect(
      :direct,
      :solr_home => File.join(RAILS_ROOT, 'solr'),
      :data_dir =>  File.join(RAILS_ROOT, 'solr', 'data', RAILS_ENV)
    )
  )

end
```

#9 - 2017-06-19 03:56 - Jens Krämer

- File 0001-moves-shellout-method-to-Utils-Shell.patch added
- File 0002-implements-fulltext-extraction-for-attachments.patch added
- File 0003-store-fulltext-in-the-attachment-model-and-make-it-s.patch added
- File 0004-turns-attachment-search-on-by-default.patch added
- File 0005-simplify-wording-for-search-form-options-de-en.patch added

I've been looking into this for Planio recently and we would like to contribute our code for this feature.

Our patch adds a fulltext column to the attachments table, which is filled with the plain text representation of the attachment, as far as possible, after the attachment has been created. As of now, the following text extractors are implemented:

- XML based office documents (LibreOffice / OpenOffice / MS Office) (through RubyZIP / Nokogiri)
- Old binary MS office formats (using the external catdoc, catppt and xls2csv commands)
- PDF (using pdf2text)
- RTF (uses the external unrtf command)
- plain text, CSV

Other formats could be added, i.e. to extract image metadata through imagemagick.
External commands come with sensible defaults and can be configured through configuration.yml. The whole feature may be turned off in the same place.
The fulltext is added to the list of columns searched when attachment search is active. We also chose to enable attachment search by default and changed the wording of the option slightly to reflect the fact that now also attachment contents will be searched.

We think this is a "good enough" solution for many if not most Redmine installations, compared to more complex external indexing solutions.

Given that most attachment uploads happen asynchronously through Javascript the added processing time for text extraction should be barely noticed by the user. Going further one could think about pushing that work into an ActiveJob worker so administrators can decide if text extraction should happen inline or if they want to set up i.e. DelayedJob or any other deferred processing backend for this.

Please let me know what you think!

#10 - 2017-06-21 10:51 - Jens Krämer

- File 0002-implements-fulltext-extraction-for-attachments.patch added

here is an updated version of patch 0002, with improved exception handling in the TextExtractor.

#11 - 2017-06-21 11:15 - Jens Krämer

- File 0001-moves-shellout-method-to-Utils-Shell.patch added
- File 0002-implements-fulltext-extraction-for-attachments.patch added
- File 0003-store-fulltext-in-the-attachment-model-and-make-it-s.patch added
- File 0004-turns-attachment-search-on-by-default.patch added
- File 0005-simplify-wording-for-search-form-options-de-en.patch added

updated all 5 patches and rebased against current trunk

#12 - 2017-06-21 11:17 - Jan from Planio www.plan.io

- File deleted (0001-moves-shellout-method-to-Utils-Shell.patch)

#13 - 2017-06-21 11:17 - Jan from Planio www.plan.io

- File deleted (0003-store-fulltext-in-the-attachment-model-and-make-it-s.patch)

#14 - 2017-06-21 11:17 - Jan from Planio www.plan.io

- File deleted (0004-turns-attachment-search-on-by-default.patch)

#15 - 2017-06-21 11:17 - Jan from Planio www.plan.io

- File deleted (0005-simplify-wording-for-search-form-options-de-en.patch)

#16 - 2017-06-21 11:17 - Jan from Planio www.plan.io

- File deleted (0002-implements-fulltext-extraction-for-attachments.patch)

#17 - 2017-06-21 11:17 - Jan from Planio www.plan.io

- File deleted (0002-implements-fulltext-extraction-for-attachments.patch)

#18 - 2017-06-21 11:21 - Jan from Planio www.plan.io

- Target version set to Candidate for next major release

This feature was high on the priorities list at Planio, so I assume it would be popular with Redmine users as well. I'd like to propose this feature for the next major release.

By the way, many of the binaries needed for text extraction are available on Windows as well. So this feature should be mostly cross-platform:

- pdftotext: http://www.foolabs.com/xpdf/download.html
- unrtf: http://gnuwin32.sourceforge.net/packages/unrtf.htm

Only catdoc/catppt/xls2csv are afaik not available on Windows, but they are only required for the "old" binary MS Office formats (doc, ppt, xls), the newer XML based formats (docx, pptx, xlsx) are supported using rubyzip & nokogiri and don't need external binaries.

#19 - 2017-06-23 05:31 - Jens Krämer

- File 0006-moves-text-extraction-to-an-ActiveJob-Job.patch added

Here's an additional patch which moves the text extraction into an ActiveJob job. By default these are executed inline, so the behaviour does not change. However users can now set up DelayedJob or one of the other possible ActiveJob backends to benefit from text extraction in the background.

#20 - 2017-06-24 05:08 - Go MAEDA

The patch from Planio is very insteresting. Thank you for sharing the patch.

I tried the patch but attachments.fulltext column was not updated in my environment. I found an error "uninitialized constant Redmine::TextExtractor::ZippedXmlHandler::Zip" in the log. Could you let me know how can I fix the problem?

log/development.log:

```
[ActiveJob] [ExtractFulltextJob] [b85b3211-0396-4ffa-a99d-ff877ea2ad6f] Performing ExtractFulltextJob from Inline(text_extraction) with arguments: 110
[ActiveJob] [ExtractFulltextJob] [b85b3211-0396-4ffa-a99d-ff877ea2ad6f]   Attachment Load (0.2ms)  SELECT  "attachments".* FROM "attachments" WHERE "attachments"."id" = ? LIMIT 1  [["id", 110]]
[ActiveJob] [ExtractFulltextJob] [b85b3211-0396-4ffa-a99d-ff877ea2ad6f] error in fulltext extraction: uninitialized constant Redmine::TextExtractor::ZippedXmlHandler::Zip
[ActiveJob] [ExtractFulltextJob] [b85b3211-0396-4ffa-a99d-ff877ea2ad6f] Performed ExtractFulltextJob from Inline(text_extraction) in 28.16ms
[ActiveJob] Enqueued ExtractFulltextJob (Job ID: b85b3211-0396-4ffa-a99d-ff877ea2ad6f) to Inline(text_extraction) with arguments: 110
```

about the attachment:

```
2.3.3 :001 > Attachment.find(110)
  Attachment Load (0.3ms)  SELECT  "attachments".* FROM "attachments" WHERE "attachments"."id" = ? LIMIT 1  [["id", 110]]
 => #<Attachment id: 110, container_id: 33, container_type: "Issue", filename: "test.docx", disk_filename: "170624114527_test.docx", filesize: 133663, content_type: "application/vnd.openxmlformats-officedocument.word...", digest: "0215fd360f2759b605151f171741b1a503f77d2bda5234d4ea...", downloads: 0, author_id: 1, created_on: "2017-06-24 02:45:27", description: "", disk_directory: "2017/06", fulltext: nil>
```

bin/about:

```
Environment:
  Redmine version             3.3.3.devel.16682
  Ruby version                2.3.3-p222 (2016-11-21) [x86_64-darwin16]
  Rails version               4.2.8
  Environment                 development
```

```
  Database adapter            SQLite
SCM:
  Subversion                  1.9.5
  Darcs                       2.12.0
  Mercurial                   3.8.4
  Cvs                         1.12.13
  Bazaar                      2.7.0
  Git                         2.11.0
  Filesystem
Redmine plugins:
  no plugin installed
```

#21 - 2017-06-25 02:46 - Go MAEDA

Go MAEDA wrote:

> *I tried the patch but attachments.fulltext column was not updated in my environment. I found an error "uninitialized constant Redmine::TextExtractor::ZippedXmlHandler::Zip" in the log. Could you let me know how can I fix the problem?*

The workaround for the error:

```
diff --git a/lib/redmine/text_extractor.rb b/lib/redmine/text_extractor.rb
index 33f922f8d..ad78f69e5 100644
--- a/lib/redmine/text_extractor.rb
+++ b/lib/redmine/text_extractor.rb
@@ -1,3 +1,5 @@
+require 'zip'
+
 module Redmine
   class TextExtractor
```

#22 - 2017-07-03 17:30 - Mischa The Evil

I like (the proposed implementation of) this feature. +1 from me...

#23 - 2017-07-07 09:16 - Go MAEDA

- Target version changed from Candidate for next major release to 4.1.0

I think this is very important and long awaited feature.
Let's discuss implementing this feature for 3.5.0.

#24 - 2017-07-27 14:58 - Go MAEDA

- Subject changed from Full Text Search of files? to Full Text Search of files

#25 - 2017-08-26 10:16 - Mitsuyoshi Kawabata

+1

#26 - 2017-08-26 10:19 - Hirofumi Kadoya

+1

#27 - 2017-08-29 05:06 - okkez _

I have considered to implement this feature, too.
This patch is nice and great work.

How about support plain text only at first step?
I want to customize and extend text extraction method via plugins or something.

#28 - 2017-10-20 21:21 - Kush Suryavanshi

+1. It will be great if this happens 3.5.0

#29 - 2018-10-26 07:37 - Jens Krämer

- File 0001-implements-fulltext-extraction-for-attachments.patch added
- File 0002-store-fulltext-in-the-attachment-model-and-make-it-s.patch added
- File 0003-turns-attachment-search-on-by-default.patch added
- File 0004-simplify-wording-for-search-form-options-de-en.patch added
- File 0005-moves-text-extraction-to-an-ActiveJob-Job.patch added

Since the creation of this patch, the text extraction logic has been extractacted into the Plaintext Gem . I now rebased the whole patch series on current master and changed it to use that gem, significantly reducing the size of the patch.

#30 - 2019-02-18 09:42 - Kouhei Sutou

I'm confirming these patches on master.

We need at least the following changes:

We need to resolve conflict for Gemfile in 0001-implements-fulltext-extraction-for-attachments.patch.

```
diff --git a/Gemfile b/Gemfile
index ffc51245b..5c7254824 100644
--- a/Gemfile
+++ b/Gemfile
@@ -14,6 +14,7 @@ gem "csv", "~> 3.0.1" if RUBY_VERSION >= "2.3" && RUBY_VERSION < "2.6"
 gem "nokogiri", (RUBY_VERSION >= "2.3" ? "~> 1.10.0" : "~> 1.9.1")
 gem "i18n", "~> 0.7.0"
 gem "rbpdf", "~> 1.19.6"
+gem "plaintext"

 # Windows does not include zoneinfo files, so bundle the tzinfo-data gem
 gem 'tzinfo-data', platforms: [:mingw, :x64_mingw, :mswin]
```

See https://github.com/kou/redmine/commit/b95407c15ed157d59066e00809310537d1fd5585.patch for resolved patch.

We need to use ActiveRecord::Migration[4.2] in migration:

```
diff --git a/db/migrate/20170613064930_add_fulltext_to_attachments.rb b/db/migrate/20170613064930_add_fulltext_to_attachments.rb
index c3d9ca5063..393dedd5f6 100644
--- a/db/migrate/20170613064930_add_fulltext_to_attachments.rb
+++ b/db/migrate/20170613064930_add_fulltext_to_attachments.rb
@@ -1,4 +1,4 @@
-class AddFulltextToAttachments < ActiveRecord::Migration
+class AddFulltextToAttachments < ActiveRecord::Migration[4.2]
   def change
     add_column :attachments, :fulltext, :text, :limit => 4.megabytes # room for at least 1 million characters / approx. 80 pages of english text
   end
```

https://github.com/kou/redmine/commit/3743bc877fe7855684fb1582a22d9f018119451f

We need to fix expected values in tests:

```
diff --git a/test/jobs/extract_fulltext_job_test.rb b/test/jobs/extract_fulltext_job_test.rb
index ed4b666069..06cf3dfca4 100644
--- a/test/jobs/extract_fulltext_job_test.rb
+++ b/test/jobs/extract_fulltext_job_test.rb
@@ -1,6 +1,7 @@
 require 'test_helper'

 class ExtractFulltextJobTest < ActiveJob::TestCase
+  fixtures :issues, :users

   def test_should_extract_fulltext
     att = nil
@@ -17,7 +18,8 @@ def test_should_extract_fulltext
     ExtractFulltextJob.perform_now(att.id)

     att.reload
-    assert att.fulltext.include?("this is a text file for upload tests\r\nwith multiple lines")
+    assert_equal("this is a text file for upload tests with multiple lines",
+                 att.fulltext)
   end

 end
diff --git a/test/unit/attachment_test.rb b/test/unit/attachment_test.rb
index 7e7edad1bf..84e8f15cef 100644
--- a/test/unit/attachment_test.rb
+++ b/test/unit/attachment_test.rb
@@ -509,6 +509,7 @@ def test_should_extract_fulltext
       :author => User.find(1),
       :content_type => 'text/plain')
     a.reload
-    assert a.fulltext.include?("this is a text file for upload tests\r\nwith multiple lines")
+    assert_equal("this is a text file for upload tests with multiple lines",
+                 a.fulltext)
   end
```

```
  end
```

https://github.com/kou/redmine/commit/6f043dcee92b0767d61139783f8ec73ef0019279

What should we do to merge this into master?

I think that the followings are remained tasks:

- Use "20180923091604" or larger prefix for db/migrate/20170613064930_add_fulltext_to_attachments.rb

- Adjust styles
  - e.g.: Don't put an empty line before the last end:

```
diff --git a/app/jobs/extract_fulltext_job.rb b/app/jobs/extract_fulltext_job.rb
index aaa716d7d..a9e54c591 100644
--- a/app/jobs/extract_fulltext_job.rb
+++ b/app/jobs/extract_fulltext_job.rb
@@ -9,5 +9,4 @@ class ExtractFulltextJob < ActiveJob::Base
       att.update_column :fulltext, text
     end
   end
-
 end
```

  - e.g.: Use && rather than and:

```
diff --git a/app/jobs/extract_fulltext_job.rb b/app/jobs/extract_fulltext_job.rb
index aaa716d7d..33992f010 100644
--- a/app/jobs/extract_fulltext_job.rb
+++ b/app/jobs/extract_fulltext_job.rb
@@ -2,9 +2,9 @@ class ExtractFulltextJob < ActiveJob::Base
   queue_as :text_extraction

   def perform(attachment_id)
-    if att = Attachment.find_by_id(attachment_id) and
-      att.readable? and
-      text = Redmine::TextExtractor.new(att).text
+    if (att = Attachment.find_by_id(attachment_id)) &&
+      att.readable? &&
+      (text = Redmine::TextExtractor.new(att).text)

       att.update_column :fulltext, text
     end
```

  - e.g.: Don't omit parentheses:

```
diff --git a/lib/redmine/configuration.rb b/lib/redmine/configuration.rb
index c72a2707a..9aed7ca3e 100644
--- a/lib/redmine/configuration.rb
+++ b/lib/redmine/configuration.rb
@@ -66,7 +66,7 @@ module Redmine
       end

       if text_extractors = @config['text_extractors']
-        Plaintext::Configuration.load YAML.dump text_extractors
+        Plaintext::Configuration.load(YAML.dump(text_extractors))
```

```
        end

        check_regular_expressions


 -  Remove "should_" prefix from test name:       diff --git a/test/jobs/extract_fulltext_job_test.rb b/test/jobs/extract_fulltext_job_test.rb
index 06cf3dfca..6a00ed67e 100644
--- a/test/jobs/extract_fulltext_job_test.rb
+++ b/test/jobs/extract_fulltext_job_test.rb
@@ -3,7 +3,7 @@ require 'test_helper'
 class ExtractFulltextJobTest < ActiveJob::TestCase
   fixtures :issues, :users

-  def test_should_extract_fulltext
+  def test_extract_fulltext
     att = nil
     Redmine::Configuration.with 'enable_fulltext_search' => false do
       att = Attachment.create(
diff --git a/test/unit/attachment_test.rb b/test/unit/attachment_test.rb
index 84e8f15ce..05a9315f7 100644
--- a/test/unit/attachment_test.rb
+++ b/test/unit/attachment_test.rb
@@ -502,7 +502,7 @@ class AttachmentTest < ActiveSupport::TestCase
     puts '(ImageMagick convert not available)'
   end

-  def test_should_extract_fulltext
+  def test_extract_fulltext
     a = Attachment.create(
       :container => Issue.find(1),
       :file => uploaded_test_file("testfile.txt", "text/plain"),



-  Make the max extracted text size customizable. Because I have some texts (such as log files) larger than 4MiB. I want to search larger texts.
diff --git a/config/configuration.yml.example b/config/configuration.yml.example
index 117d88d56..6af7ad839 100644
--- a/config/configuration.yml.example
+++ b/config/configuration.yml.example
@@ -218,6 +218,12 @@ default:
   #
   # enable_fulltext_search: false

+  # The maximum text size by text extraction for fulltext search.
+  #
+  # 4MiB by default.
+  #
+  # max_text_size: 4194304
+
   # Text extraction helper programs.
   #
   # commands should write the resulting plain text to STDOUT. Use __FILE__ as
diff --git a/lib/redmine/text_extractor.rb b/lib/redmine/text_extractor.rb
```

```
index 8d2f9e69c..34b7a361f 100644
--- a/lib/redmine/text_extractor.rb
+++ b/lib/redmine/text_extractor.rb
@@ -8,8 +8,10 @@ module Redmine
     # returns the extracted fulltext or nil if no matching handler was found
     # for the file type.
     def text
-      Plaintext::Resolver.new(@attachment.diskfile,
-                              @attachment.content_type).text
+      resolver = Plaintext::Resolver.new(@attachment.diskfile,
+                                         @attachment.content_type)
+      resolver.max_plaintext_bytes = Redmine::Configuration["max_text_size"] || 4.megabytes
+      resolver.text
     rescue Exception => e
       Rails.logger.error "error in fulltext extraction: #{e}"
       raise e unless e.is_a? StandardError # re-raise Signals / SyntaxErrors etc
```

- Use 2 ** 32 - 1 for attachments.fulltext limit. The current 4.megabytes is a bit meaningless. Because Active Record uses mediumtext for 4.megabytes for MySQL. mediumtext accepts almost 16MiB (2 ** 16 - 1). It doesn't limit to 4MiB. Active Record uses text for PostgreSQL. It doesn't have no limit. If we use 2 ** 32 - 1, MySQL uses longtext. It accepts almost 4GiB (2 ** 32 - 1). We need more 1 byte for each longtext column value than mediumtext column value. mediumtext uses value size + 3 bytes. largetext uses value size + 4 bytes. See also: https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html#data-types-storage-reqs-strings

```
diff --git a/db/migrate/20170613064930_add_fulltext_to_attachments.rb b/db/migrate/20170613064930_add_fulltext_to_attachments.rb
index 393dedd5f..b9f42ebe7 100644
--- a/db/migrate/20170613064930_add_fulltext_to_attachments.rb
+++ b/db/migrate/20170613064930_add_fulltext_to_attachments.rb
@@ -1,5 +1,5 @@
 class AddFulltextToAttachments < ActiveRecord::Migration[4.2]
   def change
-    add_column :attachments, :fulltext, :text, :limit => 4.megabytes # room for at least 1 million characters / approx. 80 pages of english text
+    add_column :attachments, :fulltext, :text, :limit => 2 ** 32 - 1
   end
 end
```

- Remove needless commits. I think that we can squash the 5 patches.

Files

0001-moves-shellout-method-to-Utils-Shell.patch	1.99 KB	2017-06-21	Jens Krämer
0003-store-fulltext-in-the-attachment-model-and-make-it-s.patch	6.38 KB	2017-06-21	Jens Krämer
0004-turns-attachment-search-on-by-default.patch	998 Bytes	2017-06-21	Jens Krämer
0005-simplify-wording-for-search-form-options-de-en.patch	1.71 KB	2017-06-21	Jens Krämer
0002-implements-fulltext-extraction-for-attachments.patch	207 KB	2017-06-21	Jens Krämer
0006-moves-text-extraction-to-an-ActiveJob-Job.patch	3.46 KB	2017-06-23	Jens Krämer
0001-implements-fulltext-extraction-for-attachments.patch	4.98 KB	2018-10-26	Jens Krämer
0002-store-fulltext-in-the-attachment-model-and-make-it-s.patch	6.38 KB	2018-10-26	Jens Krämer
0003-turns-attachment-search-on-by-default.patch	1003 Bytes	2018-10-26	Jens Krämer
0004-simplify-wording-for-search-form-options-de-en.patch	1.75 KB	2018-10-26	Jens Krämer
0005-moves-text-extraction-to-an-ActiveJob-Job.patch	3.43 KB	2018-10-26	Jens Krämer