Redmine - Feature #5893

Filter issues by notes

2010-07-16 12:26 - Peter Höpfl

Status:	Closed	**Start date:**	2010-07-16
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Filters	**Estimated time:**	0.00 hour
Target version:	5.0.0		
Resolution:			

Description

We run a big Redmine installation with more than 20000 issues. What we are missing is the ability to apply a filter on description and notes.

For Redmine version 0.9.4 I made a patch that allows us to filter for these fields and now I would like to contribute this patch to all Redmine users.
Unfortunately I have no svn access to provide an official patch file. Also I don't know if my code (first Ruby code) is ok.

Here is what i did:

```
File app/models/query.rb:

Line 125ff:
Old:
    QueryColumn.new(:subject, :sortable => "#{Issue.table_name}.subject"),

New:
    QueryColumn.new(:subject, :sortable => "#{Issue.table_name}.subject"),
    QueryColumn.new(:description, :sortable =>

Line 176ff:
Old:
                            "subject" => { :type => :text, :order => 8 },

New:
                            "subject" => { :type => :text, :order => 8 },
                            "description" => { :type => :text, :order => 16 },
                            "notes" => { :type => :text, :order => 17 },

Line 400ff:
Old:
      elsif field == 'watcher_id'
        db_table = Watcher.table_name
        db_field = 'user_id'
        sql << "#{Issue.table_name}.id #{ operator == '=' ? 'IN' : 'NOT IN' } (SELECT #{db_table}.
watchable_id FROM #{db_table} WHERE #{db_table}.watchable_type='Issue' A$
        sql << sql_for_field(field, '=', v, db_table, db_field) + ')'
      else

New:
      elsif field == 'watcher_id'
        db_table = Watcher.table_name
        db_field = 'user_id'
        sql << "#{Issue.table_name}.id #{ operator == '=' ? 'IN' : 'NOT IN' } (SELECT #{db_table}.
watchable_id FROM #{db_table} WHERE #{db_table}.watchable_type='Issue' A$
        sql << sql_for_field(field, '=', v, db_table, db_field) + ')'
      elsif field == 'notes'
        db_table = Journal.table_name
        db_field = 'notes'
        sql << '((' + sql_for_field(field, operator, v, db_table, db_field) + ') OR (' + sql_for_f
ield(field, operator, v, Issue.table_name, 'description') + '))'
      else
```

```
Line 423ff:
Old:
  def issue_count
    Issue.count(:include => [:status, :project], :conditions => statement)

New:
  def issue_count
    r = false
    filters.each_key do |field|
      next unless field == "notes"
      r = true
    end
    if r
      Issue.count(:include => [:status, :project, :journals], :conditions => statement)
    else
      Issue.count(:include => [:status, :project], :conditions => statement)
    end

Line 439ff:
Old:
  def issue_count_by_group
    r = nil
    if grouped?
      begin
        # Rails will raise an (unexpected) RecordNotFound if there's only a nil group value
        r = Issue.count(:group => group_by_statement, :include => [:status, :project], :conditions
 => statement)

New:
  def issue_count_by_group
    r = nil
    if grouped?
      begin
        x = false
        filters.each_key do |field|
          next unless field == "notes"
          x = true
        end
        if x
          # Rails will raise an (unexpected) RecordNotFound if there's only a nil group value
          r = Issue.count(:group => group_by_statement, :include => [:status, :project, :journals]
, :conditions => statement)
        else
          # Rails will raise an (unexpected) RecordNotFound if there's only a nil group value
          r = Issue.count(:group => group_by_statement, :include => [:status, :project], :conditio
ns => statement)
        end

Line 470ff:
Old:
  def issues(options={})
    order_option = [group_by_sort_order, options[:order]].reject {|s| s.blank?}.join(',')
    order_option = nil if order_option.blank?

    Issue.find :all, :include => ([:status, :project] + (options[:include] || [])).uniq,
                     :conditions => Query.merge_conditions(statement, options[:conditions]),
                     :order => order_option,
                     :limit  => options[:limit],
                     :offset => options[:offset]

New:
  def issues(options={})
    order_option = [group_by_sort_order, options[:order]].reject {|s| s.blank?}.join(',')
    order_option = nil if order_option.blank?

    r = false
```

```
    filters.each_key do |field|
      next unless field == "notes"
      r = true
    end
    if r
    Issue.find :all, :include => ([:status, :project, :journals] + (options[:include] || [])).uniq,
                      :conditions => Query.merge_conditions(statement, options[:conditions]),
                    :order => order_option,
                    :limit  => options[:limit],
                    :offset => options[:offset]
    else
    Issue.find :all, :include => ([:status, :project] + (options[:include] || [])).uniq,
                      :conditions => Query.merge_conditions(statement, options[:conditions]),
                    :order => order_option,
                    :limit  => options[:limit],
                    :offset => options[:offset]
    end
```

It would be great if you would integrate this code into Redmine.

Thanks,

Peter

Related issues:		
Related to Redmine - Feature #1159: Allow issue description to be searchable ...	**Closed**	2008-05-01
Related to Redmine - Feature #14468: Search for text in description and comme...	**Closed**	
Related to Redmine - Feature #680: free text ticket filter	**Closed**	2008-02-17

Associated revisions

Revision 20955 - 2021-04-21 09:01 - Go MAEDA

Filter issues by notes (#5893).

Patch by Yuichi HARADA.

History

#1 - 2010-07-18 02:56 - Mischa The Evil

- Tracker changed from Feature to Patch

- Category set to Issues

Fixed and improved:

- Tracker
- Description
- Category

#2 - 2011-03-30 14:54 - Simon RES

- File HOW_TO.txt added

Here is a how-to add a public/private field for journals.

#3 - 2011-11-23 19:48 - Charles Monteiro

We need to filter on description as well

#4 - 2013-01-02 14:46 - Terence Mill

related to #1159

#5 - 2013-08-06 00:29 - Mischa The Evil

- Related to Feature #14468: Search for text in description and comments in issue search and restrict projects. added

#6 - 2016-09-13 07:07 - Toshi MARUYAMA

- Related to Feature #680: free text ticket filter added

#7 - 2019-05-24 12:24 - Tomasz O

This patch enables applying filter on notes in Redmine version 3.4.4:

```
 diff -c app/models/issue_query.rb.orig app/models/issue_query.rb
*** app/models/issue_query.rb.orig    2018-01-08 13:38:16.000000000 -0600
--- app/models/issue_query.rb    2019-05-24 04:41:38.709524653 -0500
***************
*** 136,141 ****
--- 136,143 ----
      add_available_filter "estimated_hours", :type => :float
      add_available_filter "done_ratio", :type => :integer

+     add_available_filter "notes", :type => :text
+
      if User.current.allowed_to?(:set_issues_private, nil, :global => true) ||
        User.current.allowed_to?(:set_own_issues_private, nil, :global => true)
        add_available_filter "is_private",
***************
*** 359,364 ****
--- 361,376 ----
      raise StatementInvalid.new(e.message)
    end

+   def sql_for_notes_field(field, operator, value)
+     neg = (operator == '!' ? 'NOT' : '')
+     subquery = "SELECT 1 FROM #{Journal.table_name}" +
+       " WHERE #{Journal.table_name}.journalized_type='Issue' AND #{Journal.table_name}.journalized_id=#{Issue.table_name}.id" +
+       " AND (#{sql_for_field field, operator , value, Journal.table_name, 'notes'})" +
+       " AND (#{Journal.visible_notes_condition(User.current, :skip_pre_condition => true)})"
+
+     "#{neg} EXISTS (#{subquery})"
+   end
+
    def sql_for_updated_by_field(field, operator, value)
      neg = (operator == '!' ? 'NOT' : '')
      subquery = "SELECT 1 FROM #{Journal.table_name}" +
```

Maybe someone else will find them usefull.

#8 - 2019-12-02 10:39 - Alexander Achenbach

- File filter-by-notes.patch added

As a follow-up to [#5893#note-7,](#) here's a revised patch of the notes query against Redmine 4.0.5.
It fixes logic problems in operator handling that were in the original patch.

#9 - 2019-12-15 00:58 - Go MAEDA

- Category changed from Issues to Issues filter

#10 - 2021-03-09 01:44 - Yuichi HARADA

- File filter-by-notes-with-tests.patch added

Alexander Achenbach wrote:

> As a follow-up to [#5893#note-7,](#) here's a revised patch of the notes query against Redmine 4.0.5.
> It fixes logic problems in operator handling that were in the original patch.

Thank you for creating the patch.
I adjusted filter-by-notes.patch to work with Redmine-trunk ([r20766](#)) and added tests.

#11 - 2021-03-09 10:31 - Go MAEDA

- File filter-by-notes-with-tests-v2.patch added

- Target version set to Candidate for next major release

This is a feature I have wanted for years. Thank you for updating the patch and adding the test.

I have changed the patch to move the "notes" position in the filters dropdown to below the "description".

#12 - 2021-03-11 06:08 - Go MAEDA

- Target version changed from Candidate for next major release to 4.2.0

Setting the target version to 4.2.0.

#13 - 2021-03-13 03:14 - Go MAEDA

- File start-with-does-not-work.png added

- Target version changed from 4.2.0 to Candidate for next major release

The patch needs to be fixed.

"starts with" and "ends with" operators for notes don't work as expected. They behave like "any" operator. This means that even if you give a nonexistent string to those operators, the issues list displays issues that have any notes.

   start-with-does-not-work.png
My environment:

```
Environment:
  Redmine version          4.1.1.devel
  Ruby version             2.7.2-p137 (2020-10-01) [x86_64-darwin19]
  Rails version            5.2.4.4
  Environment              development
  Database adapter         SQLite
  Mailer queue             ActiveJob::QueueAdapters::InlineAdapter
  Mailer delivery          smtp
```

#14 - 2021-03-15 05:43 - Yuichi HARADA

- File filter-by-notes-with-tests-v3.patch added

Go MAEDA wrote:

> "starts with" and "ends with" operators for notes don't work as expected. They behave like "any" operator. This means that even if you give a nonexistent string to those operators, the issues list displays issues that have any notes.

I've confirmed. I fixed the patch as follows.

```
diff --git a/app/models/issue_query.rb b/app/models/issue_query.rb
index deb06dc82..4848fcb2d 100644
--- a/app/models/issue_query.rb
+++ b/app/models/issue_query.rb
@@ -451,9 +451,9 @@ class IssueQuery < Query
   def sql_for_notes_field(field, operator, value)
     subquery = "SELECT 1 FROM #{Journal.table_name}" +
       " WHERE #{Journal.table_name}.journalized_type='Issue' AND #{Journal.table_name}.journalized_id=#{Issue
.table_name}.id" +
-      " AND (#{sql_for_field field, operator.ends_with?("~") ? "~" : "*", value, Journal.table_name, 'notes'}
)" +
+      " AND (#{sql_for_field field, operator.sub(/^!/, ''), value, Journal.table_name, 'notes'})" +
       " AND (#{Journal.visible_notes_condition(User.current, :skip_pre_condition => true)})"
-    "#{operator.starts_with?("!") ? "NOT EXISTS" : "EXISTS"} (#{subquery})"
+    "#{/^!/.match?(operator) ? "NOT EXISTS" : "EXISTS"} (#{subquery})"
   end

   def sql_for_updated_by_field(field, operator, value)
```

#15 - 2021-04-05 11:17 - Go MAEDA

- Target version changed from Candidate for next major release to 5.0.0

Setting the target version to 5.0.0.

#16 - 2021-04-05 11:58 - Go MAEDA

- File statement-invalid-log.txt added

- Target version changed from 5.0.0 to Candidate for next major release

While I was trying the patch, I came across an SQL error (sorry, I don't know how to reproduce).

```
Query::StatementInvalid: SQLite3::SQLException: no such column: issues.notes
```

#17 - 2021-04-15 04:16 - Go MAEDA

- Target version changed from Candidate for next major release to 5.0.0

Go MAEDA wrote:

> While I was trying the patch, I came across an SQL error (sorry, I don't know how to reproduce).
>
> [...]

I have been investigating this issue for a while, and the only time I could reproduce the error was when I removed the sql_for_notes_field method.

The error reported in #5893#note-16 is probably due to my tampering the source code while checking the patch.

Setting the target version to 5.0.0 again.

#18 - 2021-04-21 09:01 - Go MAEDA

- Subject changed from Filter Options: Allow to filter for description and notes to Filter issues by notes

- Status changed from New to Closed

- Assignee set to Go MAEDA

Committed the patch. Thank you for your contribution.

#19 - 2021-05-31 07:00 - Go MAEDA

- Tracker changed from Patch to Feature

#20 - 2022-03-27 04:56 - Go MAEDA

- Category changed from Issues filter to Filters

Files

HOW_TO.txt	5.04 KB	2011-03-30	Simon RES
filter-by-notes.patch	1.32 KB	2019-12-02	Alexander Achenbach
filter-by-notes-with-tests.patch	3.39 KB	2021-03-09	Yuichi HARADA
filter-by-notes-with-tests-v2.patch	3.42 KB	2021-03-09	Go MAEDA
start-with-does-not-work.png	68.8 KB	2021-03-13	Go MAEDA
filter-by-notes-with-tests-v3.patch	3.4 KB	2021-03-15	Yuichi HARADA
statement-invalid-log.txt	3.78 KB	2021-04-05	Go MAEDA