

## Redmine - Defect #32915

### Internal Server Error occurred when exporting gantt chart to png on Windows

2020-01-29 05:11 - Anonymous

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Gantt	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Affected version:</b>	4.1.0
<b>Resolution:</b>			
<b>Description</b>			
<p>Redmine 4.0.4 using RMagick was fine. However, Redmine 4.1.0 using MiniMagick does not work.</p> <p>production.log of Redmine 4.1.0 is this:</p> <pre>Started GET "/redmine/projects/some_project/issues/gantt.png?..." for (IP) at (date) Processing by GanttController#show as PNG   Parameters: {...}   Current user: admin (id=1) Completed 500 Internal Server Error in 290ms (ActiveRecord: 63.0ms)  Errno::E2BIG (Arg list too long - convert):  lib/redmine/helpers/gantt.rb:381:in `to_image' app/controllers/gantt_controller.rb:44:in `block (2 levels) in show' app/controllers/gantt_controller.rb:42:in `show'</pre> <p>On Windows, the ImageMagick's "convert" command length is too long, and then error occurred. I want to shorten the command length or split the command if Redmine keeps using Minimagick. And there should be some exception code as below:</p> <pre>--- a\lib\redmine\helpers\gantt.rb +++ b\lib\redmine\helpers\gantt.rb @@ -488,4 +488,15 @@      end      img.to_blob +  rescue +    MiniMagick::Tool::Convert.new do  gc  +      # create error message image +      gc.size('%dx%d' % [200, 50]) +      gc.xc('white') +      gc.stroke('transparent').fill('red') +      # and draw error message +      gc.draw('text %d,%d %s' % [10, 20, Redmine::Utils::Shell.shell_quote('Gantt size too bi g.')]]) +      gc &lt;&lt; img.path +    end +    img.to_blob   ensure     img.destroy! if img</pre> <p>For some reason, I have to keep using Redmine on Windows...</p>			

#### History

##### #1 - 2020-01-31 05:17 - Anonymous

To work on Windows, divided the MiniMagick "convert" block.

This patch can display up to about 200 issues.

However, the subjects() and lines() functions create "convert" commands internally, so these could't be divided.

```
--- a/lib/redmine/helpers/gantt.rb
+++ b/lib/redmine/helpers/gantt.rb
@@ -386,4 +386,11 @@
      gc.stroke('transparent')
      subjects(:image => gc, :top => (headers_height + 20), :indent => 4, :format => :image)
+      gc << img.path
+      gc.call
+    end
+    # Make blocks smaller for windows
+    MiniMagick::Tool::Convert.new do |gc|
+      gc << img.path
+      gc.font(font_path) if font_path.present?
+      # Months headers
+      month_f = @date_from
@@ -472,4 +479,11 @@
      0, 0, subject_width + g_width, g_height + headers_height - 1
    ])
+    gc << img.path
+    gc.call
+  end
+  # Make blocks smaller for windows
+  MiniMagick::Tool::Convert.new do |gc|
+    gc << img.path
+    gc.font(font_path) if font_path.present?
+    # content
+    top = headers_height + 20
```

## #2 - 2020-02-07 06:37 - Anonymous

Changed to split MiniMagick's "convert" command on Windows only.

To do so, changed the "convert" object (gc) to an instance variable (@gc), and added an @minimagick\_command\_max instance variable for the command splitting.

```
@@ -90,4 +90,6 @@
      @max_rows = Setting.gantt_items_limit.blank? ? nil : Setting.gantt_items_limit.to_i
    end
+    @gc = nil
+    @minimagick_command_max = Redmine::Platform.mswin? ? 1000 : -1
  end

@@ -379,11 +381,11 @@
      font_path = Redmine::Configuration['minimagick_font_path'].presence || Redmine::Configuration['rmagic
k_font_path'].presence
      img = MiniMagick::Image.create("#{format}", false)
      MiniMagick::Tool::Convert.new do |gc|
-        gc.size('%dx%d' % [subject_width + g_width + 1, height])
-        gc.xc('white')
-        gc.font(font_path) if font_path.present?
+        @gc = MiniMagick::Tool::Convert.new
+        @gc.size('%dx%d' % [subject_width + g_width + 1, height])
+        @gc.xc('white')
+        @gc.font(font_path) if font_path.present?
+        # Subjects
-        gc.stroke('transparent')
-        subjects(:image => gc, :top => (headers_height + 20), :indent => 4, :format => :image)
+        @gc.stroke('transparent')
+        subjects(:image => @gc, :top => (headers_height + 20), :indent => 4, :format => :image, :img_path =
> img.path, :font_path => font_path)
+        # Months headers
+        month_f = @date_from
@@ -391,14 +393,14 @@
      @months.times do
        width = ((month_f >> 1) - month_f) * zoom
-        gc.fill('white')
-        gc.stroke('grey')
-        gc.strokewidth(1)
-        gc.draw('rectangle %d,%d %d,%d' % [
+        @gc.fill('white')
+        @gc.stroke('grey')
+        @gc.strokewidth(1)
+        @gc.draw('rectangle %d,%d %d,%d' % [
          left, 0, left + width, height
```

```

    ])
-   gc.fill('black')
-   gc.stroke('transparent')
-   gc.strokewidth(1)
-   gc.draw('text %d,%d %s' % [
+   @gc.fill('black')
+   @gc.stroke('transparent')
+   @gc.strokewidth(1)
+   @gc.draw('text %d,%d %s' % [
        left.round + 8, 14, Redmine::Utils::Shell.shell_quote("#{month_f.year}-#{month_f.month}")
    ])
@@ -417,8 +419,8 @@
    week_f = @date_from + (7 - @date_from.cwday + 1)
    width = (7 - @date_from.cwday + 1) * zoom
-   gc.fill('white')
-   gc.stroke('grey')
-   gc.strokewidth(1)
-   gc.draw('rectangle %d,%d %d,%d' % [
+   @gc.fill('white')
+   @gc.stroke('grey')
+   @gc.strokewidth(1)
+   @gc.draw('rectangle %d,%d %d,%d' % [
        left, header_height, left + width, 2 * header_height + g_height - 1
    ])
@@ -427,14 +429,14 @@
    while week_f <= date_to
        width = (week_f + 6 <= date_to) ? 7 * zoom : (date_to - week_f + 1) * zoom
-   gc.fill('white')
-   gc.stroke('grey')
-   gc.strokewidth(1)
-   gc.draw('rectangle %d,%d %d,%d' % [
+   @gc.fill('white')
+   @gc.stroke('grey')
+   @gc.strokewidth(1)
+   @gc.draw('rectangle %d,%d %d,%d' % [
        left.round, header_height, left.round + width, 2 * header_height + g_height - 1
    ])
-   gc.fill('black')
-   gc.stroke('transparent')
-   gc.strokewidth(1)
-   gc.draw('text %d,%d %s' % [
+   @gc.fill('black')
+   @gc.stroke('transparent')
+   @gc.strokewidth(1)
+   @gc.draw('text %d,%d %s' % [
        left.round + 2, header_height + 14, Redmine::Utils::Shell.shell_quote(week_f.cweek.to_s)
    ])
@@ -450,10 +452,18 @@
    (date_to - @date_from + 1).to_i.times do
        width = zoom
-   gc.fill(non_working_week_days.include?(wday) ? '#eee' : 'white')
-   gc.stroke('#ddd')
-   gc.strokewidth(1)
-   gc.draw('rectangle %d,%d %d,%d' % [
+   @gc.fill(non_working_week_days.include?(wday) ? '#eee' : 'white')
+   @gc.stroke('#ddd')
+   @gc.strokewidth(1)
+   @gc.draw('rectangle %d,%d %d,%d' % [
        left, 2 * header_height, left + width, 2 * header_height + g_height - 1
    ])
+   # Make blocks smaller for windows
+   if (@minimagick_command_max > 0) && (@gc.command.length > @minimagick_command_max)
+       @gc << img.path
+       @gc.call
+       @gc = MiniMagick::Tool::Convert.new
+       @gc << img.path
+       @gc.font(font_path) if font_path.present?
+   end
    left = left + width
    wday = wday + 1
@@ -462,29 +472,45 @@
    end
    # border
-   gc.fill('transparent')
-   gc.stroke('grey')

```

```

-         gc.strokewidth(1)
-         gc.draw('rectangle %d,%d %d,%d' % [
+         @gc.fill('transparent')
+         @gc.stroke('grey')
+         @gc.strokewidth(1)
+         @gc.draw('rectangle %d,%d %d,%d' % [
            0, 0, subject_width + g_width, headers_height
        ])
-         gc.stroke('black')
-         gc.draw('rectangle %d,%d %d,%d' % [
+         @gc.stroke('black')
+         @gc.draw('rectangle %d,%d %d,%d' % [
            0, 0, subject_width + g_width, g_height + headers_height - 1
        ])
        # content
        top = headers_height + 20
-         gc.stroke('transparent')
-         lines(:image => gc, :top => top, :zoom => zoom,
-             :subject_width => subject_width, :format => :image)
+         @gc.stroke('transparent')
+         lines(:image => @gc, :top => top, :zoom => zoom,
+             :subject_width => subject_width, :format => :image, :img_path => img.path, :font_path => font
+         _path)
        # today red line
        if User.current.today >= @date_from and User.current.today <= date_to
-         gc.stroke('red')
+         @gc.stroke('red')
            x = (User.current.today - @date_from + 1) * zoom + subject_width
-         gc.draw('line %g,%g %g,%g' % [
+         @gc.draw('line %g,%g %g,%g' % [
            x, headers_height, x, headers_height + g_height - 1
        ])
        end
-         gc << img.path
-         end
+         @gc << img.path
+         @gc.call
+         img.to_blob
+         rescue
+         @gc = MiniMagick::Tool::Convert.new
+         # create error message image
+         @gc.size('%dx%d' % [600, 120])
+         @gc.xc('white')
+         @gc.stroke('transparent').fill('red')
+         @gc.pointsize(20)
+         # and draw error message
+         err_message = <<"EOS"
+Image generation failed.
+There may be too much issues.
+Please decrease the number of issues.
+EOS
+         @gc.draw('text %d,%d %s' % [10, 40, Redmine::Utils::Shell.shell_quote(err_message)])
+         @gc << img.path
+         img.to_blob
+         ensure
@@ -803,10 +829,18 @@

def image_subject(params, subject, options={})
-     params[:image].fill('black')
-     params[:image].stroke('transparent')
-     params[:image].strokewidth(1)
-     params[:image].draw('text %d,%d %s' % [
+     @gc.fill('black')
+     @gc.stroke('transparent')
+     @gc.strokewidth(1)
+     @gc.draw('text %d,%d %s' % [
        params[:indent], params[:top] + 2, Redmine::Utils::Shell.shell_quote(subject)
    ])
+     # Make blocks smaller for windows
+     if (@minimagick_command_max > 0) && (@gc.command.length > @minimagick_command_max)
+         @gc << params[:img_path]
+         @gc.call
+         @gc = MiniMagick::Tool::Convert.new
+         @gc << params[:img_path]
+         @gc.font(params[:font_path]) if params[:font_path].present?

```

```

+         end
+     end

@@ -992,6 +1026,6 @@
    # Renders the task bar, with progress and late
    if coords[:bar_start] && coords[:bar_end]
-       params[:image].fill('#aaa')
-       params[:image].draw('rectangle %d,%d %d,%d' % [
+       @gc.fill('#aaa')
+       @gc.draw('rectangle %d,%d %d,%d' % [
            params[:subject_width] + coords[:bar_start],
            params[:top],
@@ -1000,6 +1034,6 @@
        ])
        if coords[:bar_late_end]
-       params[:image].fill('#f66')
-       params[:image].draw('rectangle %d,%d %d,%d' % [
+       @gc.fill('#f66')
+       @gc.draw('rectangle %d,%d %d,%d' % [
            params[:subject_width] + coords[:bar_start],
            params[:top],
@@ -1009,6 +1043,6 @@
        end
        if coords[:bar_progress_end]
-       params[:image].fill('#00c600')
-       params[:image].draw('rectangle %d,%d %d,%d' % [
+       @gc.fill('#00c600')
+       @gc.draw('rectangle %d,%d %d,%d' % [
            params[:subject_width] + coords[:bar_start],
            params[:top],
@@ -1023,6 +1057,6 @@
        x = params[:subject_width] + coords[:start]
        y = params[:top] - height / 2
-       params[:image].fill('blue')
-       params[:image].draw('polygon %d,%d %d,%d %d,%d %d,%d' % [
+       @gc.fill('blue')
+       @gc.draw('polygon %d,%d %d,%d %d,%d %d,%d' % [
            x - 4, y,
            x, y - 4,
@@ -1034,6 +1068,6 @@
        x = params[:subject_width] + coords[:end] + params[:zoom]
        y = params[:top] - height / 2
-       params[:image].fill('blue')
-       params[:image].draw('polygon %d,%d %d,%d %d,%d %d,%d' % [
+       @gc.fill('blue')
+       @gc.draw('polygon %d,%d %d,%d %d,%d %d,%d' % [
            x - 4, y,
            x, y - 4,
@@ -1045,9 +1079,17 @@
        # Renders the label on the right
        if label
-       params[:image].fill('black')
-       params[:image].draw('text %d,%d %s' % [
+       @gc.fill('black')
+       @gc.draw('text %d,%d %s' % [
            params[:subject_width] + (coords[:bar_end] || 0) + 5, params[:top] + 1, Redmine::Utils::Shell.shell_quote(label)
        ])
    end
+    # Make blocks smaller for windows
+    if (@minimagick_command_max > 0) && (@gc.command.length > @minimagick_command_max)
+        @gc << params[:img_path]
+        @gc.call
+        @gc = MiniMagick::Tool::Convert.new
+        @gc << params[:img_path]
+        @gc.font(params[:font_path]) if params[:font_path].present?
+    end
+end
+end
+end

```

### #3 - 2020-02-07 08:39 - Go MAEDA

- Subject changed from Internal Server Error occurred when exporting gantt chart to png to Internal Server Error occurred when exporting gantt chart to png on Windows

#4 - 2021-09-17 05:30 - 0000

taca tadocolo wrote:

Changed to split MiniMagick's "convert" command on Windows only.

To do so, changed the "convert" object (gc) to an instance variable (@gc), and added an @minimagick\_command\_max instance variable for the command splitting.

[...]

hello! taka tadocolo

"@gc.size('%dx%d' % [subject\_width + g\_width + 1, height])" build error was occurred.

how to fix it!