

Redmine - Feature #39111

Enable Asset Pipeline Integration using Propshaft

2023-09-24 17:48 - Takashi Kato

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Marius BĂLTEANU	% Done:	0%
Category:	Code cleanup/refactoring	Estimated time:	0.00 hour
Target version:	6.0.0		
Resolution:	Fixed		
Description			
<p>This patch enables the Asset Pipeline in Redmine, assuming you have upgraded to Rails 7(#36320).</p> <p>Until Rails 6.1, Sprockets was a prerequisite for using the Asset Pipeline. However, starting from Rails 7, Propshaft has been introduced as an alternative option.</p> <p>While Propshaft offers fewer features compared to Sprockets, it boasts a smaller codebase and greater clarity in its operations (eliminating the need for a JavaScript runtime). Additionally, manifest files are no longer necessary during development.</p> <p>Key benefits of using Propshaft include:</p> <ul style="list-style-type: none">• The ability to append file digests to file names instead of Rails3 style cache-busting approach involving query strings at the end of assets (expected to enhance cache hit rates).• Support for tasks like copying plugin assets to the public directory each time Redmine starts.• Compatibility with Turbo, which is almost mandatory in Rails 7 and later. (Note that Rails 7 and later deprecate rails-ujs, the current Redmine dependency.) <p>To address potential challenges associated with introducing Propshaft, we have extended its functionality in the following ways:</p> <ul style="list-style-type: none">• We've made it possible for Propshaft to handle Redmine plugin and theme assets without requiring modification(If we use propshaft in default, we will need to modify the links to other assets in stylesheets.).• We've ensured that introducing Asset Pipeline does not necessitate rewriting existing stylesheets, thus avoiding major incompatibilities with Redmine themes.• In a production environment, rather than manually running the rails asset:precompile command before starting Rails, we now automatically check for asset updates and run precompilation as needed. (You can disable this automatic execution by setting config.assets.redmine_detect_update to false in config/environments/production.rb).			
Related issues:			
Related to Redmine - Patch #40137: Jstoolbar help files should import images ...		Closed	
Related to Redmine - Defect #40204: `rake redmine:plugins` fails with the err...		Closed	
Related to Redmine - Patch #40221: Update wiki content related to how to crea...		New	
Related to Redmine - Patch #40559: Fix incorrect icon image paths for Wiki he...		Closed	

Associated revisions

Revision 22626 - 2024-01-25 06:38 - Marius BĂLTEANU

Add Propshaft library to enable the asset pipeline without modifying existing assets (#39111).

Patch by Takashi Kato (@tohosaku).

Revision 22627 - 2024-01-25 06:40 - Marius BĂLTEANU

Remove code that is no longer needed as a result of asset pipeline implementation (#39111).

Patch by Takashi Kato (@tohosaku).

Revision 22628 - 2024-01-25 06:41 - Marius BĂLTEANU

Fix tests after asset pipeline implementation (#39111).

Patch by Takashi Kato (@tohosaku).

Revision 22629 - 2024-01-25 06:44 - Marius BĂLTEANU

Remove hardcoded version of rails-ujs and include the one provided by the asset pipeline (#39111).

Revision 22630 - 2024-01-25 06:45 - Marius BĂLTEANU

Removes redmine:plugins:assets task which is no longer needed after asset pipeline implementation (#39111).

Revision 22631 - 2024-01-25 08:32 - Marius BĂLTEANU

Fixes javascript file name by removing ujs reference (#39111).

Revision 22638 - 2024-01-26 03:20 - Marius BĂLTEANU

Adds debug code to investigate test fail on CI (#39111).

Revision 22639 - 2024-01-26 03:39 - Marius BĂLTEANU

Fixes multiple RuboCop offenses (#39111).

Revision 22640 - 2024-01-26 03:42 - Marius BĂLTEANU

Removes debug code added in r22638 (#39111).

Revision 22641 - 2024-01-26 03:52 - Marius BĂLTEANU

Fix failing test by ignoring the directories that start with a dot (#39111).

Revision 22642 - 2024-01-26 09:02 - Go MAEDA

Fix RuboCop offense Performance/RedundantEqualityComparisonBlock (#39111).

Revision 22643 - 2024-01-26 09:19 - Go MAEDA

Fix RuboCop offense Style/ReturnNilInPredicateMethodDefinition (#39111).

Revision 22644 - 2024-01-26 09:29 - Go MAEDA

Fix RuboCop offense Style/MapToHash (#39111).

Revision 22645 - 2024-01-27 02:41 - Go MAEDA

Fix RuboCop offense Style/BlockDelimiters, Layout/MultilineBlockLayout, and Layout/BlockEndNewline (#39111).

Revision 22646 - 2024-01-27 02:44 - Go MAEDA

Fix RuboCop offense Layout/EmptyLineAfterMagicComment (#39111).

Revision 22649 - 2024-01-27 04:55 - Marius BĂLTEANU

Use images generated by the asset pipeline in context_menu CSS files (#39111).

Revision 22657 - 2024-01-29 02:03 - Marius BĂLTEANU

Ensure url in context_menu css files is absolute (#39111).

Revision 22658 - 2024-01-29 02:05 - Marius BĂLTEANU

Use absolute URL (propshaft default format) for images in jstoolbar.css (#39111).

Revision 22659 - 2024-01-29 02:11 - Marius BĂLTEANU

Use absolute URL (propshaft default format) for images in rtl and scm css files (#39111).

Revision 22660 - 2024-01-29 02:16 - Marius BĂLTEANU

Use absolute URL (propshaft default format) for images in application.css (#39111).

Revision 22661 - 2024-01-29 02:21 - Marius BĂLTEANU

Import application.css in themes using absolute url (#39111).

Revision 22662 - 2024-01-29 02:28 - Marius BĂLTEANU

Use absolute URL (propshaft default format) for images in application.css from classic theme (#39111).

Revision 22690 - 2024-02-07 20:53 - Marius BĂLTEANU

Removes redmine:plugins:assets from redmine:plugins task (#40204).

Revision 22691 - 2024-02-07 20:55 - Marius BĂLTEANU

Removes configuration flag example related to redmine:plugins:assets (#40204, #39111).

Revision 22692 - 2024-02-07 21:41 - Marius BĂLTEANU

Includes favicon in assets pipeline (#39111).

Revision 22693 - 2024-02-07 21:42 - Marius BĂLTEANU

Replaces public/plugin_assets with public/assets in installation docs (#39111).

Revision 22694 - 2024-02-07 21:44 - Marius BĂLTEANU

Move public assets (images, javascripts, stylesheets, themes) to app/assets (#39111).

Revision 22695 - 2024-02-07 21:48 - Marius BĂLTEANU

Reverts r22694 (#39111).

Revision 22696 - 2024-02-07 21:55 - Marius BĂLTEANU

Move public assets (images, javascripts, stylesheets, themes) to app/assets (#39111).

Revision 22704 - 2024-02-13 22:55 - Marius BĂLTEANU

Fix failing tests related to favicon after r22692 (#39111).

Revision 22705 - 2024-02-13 23:14 - Marius BĂLTEANU

Fix failing tests related to favicon in themes after r22692 (#39111).

Revision 22706 - 2024-02-13 23:54 - Marius BĂLTEANU

Fix one more failing test related to favicon after r22692 (#39111).

Revision 22708 - 2024-02-14 00:11 - Marius BĂLTEANU

Custom themes are now loaded from "/themes" directory from project root. The themes provided by Redmine (alternate and classic) remains in "app/assets/themes" (#39111).

Revision 22711 - 2024-02-14 00:23 - Marius BĂLTEANU

Updates UPGRADING doc to include the new themes directory location from [redmine_root] (#39111).

Revision 22712 - 2024-02-14 00:26 - Marius BĂLTEANU

Update .gitignore with the new location for themes directory (#39111).

Revision 22716 - 2024-02-18 20:39 - Marius BĂLTEANU

Add svn:ignore property on new themes directory (#39111).

Revision 22717 - 2024-02-19 21:24 - Marius BĂLTEANU

- Moves all wiki syntax files from public/help to app/views/help/wiki_syntax.
- Adds route to serve wiki syntax help files.
- Moves wiki syntax CSS files to asset pipeline.

(#39111, #40137).

Revision 22718 - 2024-02-19 21:28 - Marius BĂLTEANU

Reverts r22717 (#39111, #40137).

Revision 22719 - 2024-02-19 21:41 - Marius BĂLTEANU

- Introduces HelpController and a new route help/wiki_syntax/(type) to serve wiki syntax help files.
- Moves all wiki syntax help files to app/views/help and CSS files to asset pipeline.

(#39111, #40137).

Revision 22720 - 2024-02-19 21:51 - Marius BĂLTEANU

Reorganize all the wiki syntax help files per text formatter (markdown, common_mark and textile) (#39111, #40137).

Revision 22721 - 2024-02-19 21:54 - Marius BĂLTEANU

- Remove languages from wiki syntax help files that are identical with en language.
- Fall back to the english help page if there is none for the current language.

(#39111, #40137).

Revision 22722 - 2024-02-19 21:58 - Marius BĂLTEANU

Updates JS toolbar help button to use the new route (#39111, #40137).

Revision 22723 - 2024-02-19 22:00 - Marius BĂLTEANU

Adds erb extension to all wiki syntax help files (#39111, #40137).

Revision 22724 - 2024-02-19 22:02 - Marius BĂLTEANU

Cleanup wiki syntax files by moving inline style for img border to CSS (#39111, #40137).

Revision 22725 - 2024-02-19 22:03 - Marius BĂLTEANU

Includes CSS files from the asset pipeline (#39111, #40137).

Revision 22726 - 2024-02-19 22:09 - Marius BĂLTEANU

Replaces static img tag from wiki syntax help files with image_tag (#39111, #40137).

Revision 22727 - 2024-02-19 22:28 - Marius BĂLTEANU

Replaces static link to detailed version with the new route (#39111, #40137).

Revision 22728 - 2024-02-19 22:37 - Marius BĂLTEANU

Cleanup empty directories (#39111, #40137).

Revision 22729 - 2024-02-21 07:36 - Marius BĂLTEANU

Deletes Redmine::WikiFormatting::CommonMark::HelperTest tests that are no longer required after wiki syntax help files refactor (#39111, #40137).

Revision 22730 - 2024-02-25 21:12 - Marius BĂLTEANU

Add tests for /help/wiki_syntax route (#39111, #40137).

Revision 22731 - 2024-02-25 21:53 - Marius BĂLTEANU

Add tests for HelpController (#39111, #40137).

Revision 22732 - 2024-02-25 22:41 - Marius BĂLTEANU

Adds route for code highlighting languages page (#39111, #40137).

Revision 22733 - 2024-02-25 22:42 - Marius BĂLTEANU

Use assert_select to assert text from h1 element of wiki syntax pages (#39111, #40137).

Revision 22757 - 2024-03-03 23:24 - Marius BĂLTEANU

Fix rubocop warning (#39111, #40137).

Revision 22771 - 2024-03-08 08:13 - Go MAEDA

Fix RuboCop offense Layout/SpaceInsideStringInterpolation (#39111, #40137).

Revision 22782 - 2024-04-16 20:56 - Marius BĂLTEANU

Fix incorrect icon image paths on Markdown help page (#40559, #40137, #39111).

Patch by Katsuya HIDAKA (hidakatsuya).

Revision 22783 - 2024-04-16 20:57 - Marius BĂLTEANU

Fix incorrect icon image paths on Textile help page (#40559, #40137, #39111).

Patch by Katsuya HIDAKA (hidakatsuya).

Revision 22784 - 2024-04-16 20:58 - Marius BĂLTEANU

Fix incorrect icon image path for DatePicker button (#40559, #39111).

Patch by Katsuya HIDAKA (hidakatsuya).

Revision 22785 - 2024-04-16 21:23 - Marius BĂLTEANU

Add short description for flag (#39111).

History

#1 - 2023-10-24 07:44 - Dmitry Lisichkin

Hello. Sorry for long response.

Am I right that your idea is about using propshaft without other tools (without compilation)?

In my work I start from another side - compilation with old good sprockets as digest generator.

#2 - 2023-10-24 08:05 - Dmitry Lisichkin

And after all I can say that your way can be much simpler because of using tools like webpack or other requires many changes in existed js files of redmine.

For example application.js is just a big helper that in proper way should be splitted to many small libraries.

Another problem is a huge usage of js.erb files that directly call functions from application.js and other js files.

#3 - 2023-10-24 08:20 - Dmitry Lisichkin

One other problem that we have on our installation - many existed themes and plugins assumed that redmine assets available on root of public directory (not in assets subdirectory), I see a lot of hardcoded pathes in js and css that use absolute and relative paths to the root of redmine hosts. As result I still have old mechanic with copying assets to root of public directory for backward compability.

Good alternative is usage of old behaviour for plugin_name/assets/* and use new behaviour for some other directories like plugin_name/app/assets/*.

#4 - 2023-12-22 05:10 - Marius BĂLTEANU

- Target version set to 6.0.0

Let's discuss this for Redmine 6.0.0.

#5 - 2024-01-01 14:56 - Marius BĂLTEANU

[Takashi Kato](#), this is a very nice improvement, thank you for your hard working on this.

I made some quick tests now and I didn't find any major problem, I was able to use the current trunk version of Redmine with all three patches applied and two plugins:

- additional: <https://github.com/alphanodes/additionals>
- redmine_issue_dynamic_edit: https://github.com/llogeek/redmine_issue_dynamic_edit

Also, I've quickly created a fourth patch (attached) that removes the manually hardcoded rails-ujs library from jquery-3.6.1-ui-1.13.2-ujs-7.1.2.js and uses the one provided by the actionview gem, it's working very well.

Some notes:

1. Task bin/rails redmine:plugins:assets no longer works:

```
bin/rails aborted!  
NoMethodError: undefined method `mirror_assets' for Redmine::PluginLoader:Class
```

```
Redmine::PluginLoader.mirror_assets(name)  
^^^^^^^^^^^^^^^^^^^^  
/work/lib/tasks/redmine.rake:164:in `block (3 levels) in <top (required)>'
```

Should we adapt it to generate the assets for a specific plugin or completely drop it.

2. For Redmine core assets, as next improvement after integrating these patches, I would like to move our existing assets to app/assets for the sake of consistency with Rails default structure and to not provide in the public folder two folders, the "source" assets and the actual assets (generated with Asset Pipeline). I'm aware that more changes are required, but I do not see a real problem to include all of them in [6.0.0](#).

3. [Dmitry Lisichkin](#) has a point in [#note-3](#), but at the same time, I think the complexity to keep both behaviours for plugins do not worth it. I'm even in favour to move the plugin assets in "app/assets" to force plugins to use the correct way and also to stick to the Rails structure. The changes to adapt a plugin to this breaking change are minimal.

#6 - 2024-01-01 14:56 - Marius BĂLTEANU

- File 0004-Use-bundled-rails-ujs.patch added

#7 - 2024-01-02 05:48 - Takashi Kato

- File 0001v2-Add-propshaft-to-enable-asset-pipeline.patch added

[Marius BĂLTEANU](#)

Thank you for your review and positive feedback.

I agree with matching the behavior of the plugin to Railsway as much as possible.

I've updated the 0001-patch to fix the bug that prevents reloading when some assets are modified in development mode(the name of instance variable was incorrect).

[Dmitry Lisichkin](#)

You're correct to point out that if an asset is hard-coded in JavaScript under the 'public' directory, that section of the JavaScript requires modification when applying this patch. However, such path definitions can also cause issues like incorrect cache retrieval during version upgrades. Modifying the image retrieval method in line with the Asset Pipeline introduction could address this.

"js.erb" files in the app/views directory are returned by the controller and not handled with the asset pipeline.

While Ruby on Rails has recently seen the role of "js.erb" replaced by Hotwire's Turbo Streams, that's distinct from the changes under discussion here.

#8 - 2024-01-25 06:48 - Marius BĂLTEANU

- Assignee set to Marius BĂLTEANU

I've committed the first 4 patches in order to enable the asset pipeline. I'm aware that it's a big and *potentially* breaking change so I wanted to integrate this as soon as possible in order to have enough time to get feedback from everyone involved.

I'm going to make more changes in the following days, my goal is to match as much as possible the rails recommended directory structure with minimal changes for plugins developers.

#9 - 2024-01-25 08:52 - Marius BĂLTEANU

[Takashi Kato](#), do you have any clue why the following test fails on Redmine CI, but not on local environment?

Failure:

```
Redmine::AssetPathTest#test_asset_path_size [test/unit/lib/redmine/asset_path_test.rb:31]:
```

```
Expected: 2
```

```
Actual: 3
```

```
bin/rails test test/unit/lib/redmine/asset_path_test.rb:30
```

#10 - 2024-01-25 23:04 - Dmitry Lisichkin

Marius BĂLTEANU wrote in [#note-9](#):

[Takashi Kato](#), do you have any clue why the following test fails on Redmine CI, but not on local environment?

[...]

Can it be .svn directory?

#11 - 2024-01-26 03:22 - Marius BĂLTEANU

Dmitry Lisichkin wrote in [#note-10](#):

Marius BĂLTEANU wrote in [#note-9](#):

[Takashi Kato](#), do you have any clue why the following test fails on Redmine CI, but not on local environment?

[...]

Can it be .svn directory?

It could be, I've added some test code to see what asset_path contains, now we need to wait for the CI to run.

#12 - 2024-01-26 03:54 - Marius BĂLTEANU

Fix the failing test in [r22641](#).

[Dmitry Lisichkin](#), thanks for your advise!

#13 - 2024-01-26 23:35 - Takashi Kato

[Marius BĂLTEANU](#)

Thank you for the detailed review and merge!!

[Dmitry Lisichkin](#)

I didn't realize it! Thanks!

#14 - 2024-01-27 04:55 - Marius BĂLTEANU

- Subject changed from Asset Pipeline Integration in Redmine for Rails 7 without modifying existing themes to Enable Asset Pipeline Integration

#15 - 2024-01-29 02:42 - Marius BĂLTEANU

- Related to Patch #40137: Jstoolbar help files should import images from the asset pipeline added

#16 - 2024-01-29 02:47 - Marius BĂLTEANU

- Subject changed from Enable Asset Pipeline Integration to Enable Asset Pipeline Integration using Propshaft

I changed all CSS files from public directory to use absolute url to images which is Propshaft expected format.

Takashi Kato, you're implementation Redmine::Asset.new is very useful and I think we should keep it for at least 1 major version in order to give plugins developer time to update their files, but with a deprecation message in the logs. What do you think?

Also, I've added a new issue that we need to fix [#40137](#).

#17 - 2024-02-07 14:26 - Marius BĂLTEANU

- Related to Defect #40204: `rake redmine:plugins` fails with the error "Don't know how to build task 'redmine:plugins:assets'" added

#18 - 2024-02-07 22:04 - Marius BĂLTEANU

I made the following changes:

- included favicon in assets pipeline and switched to favicon_link_tag (https://apidock.com/rails/ActionView/Helpers/AssetTagHelper/favicon_link_tag)
- updated installation docs
- moved all public assets (javascript, stylesheets, images and themes) from public to app/assets

I'm wondering if it's a good idea to split our assets from the one owned by outside entities, such as jQuery, jQuery UI or Tribute and move all these to vendor/javascripts and vendor/stylesheets.

#19 - 2024-02-09 05:36 - Mizuki ISHIKAWA

Marius BĂLTEANU wrote in [#note-18](#):

I made the following changes:

- included favicon in assets pipeline and switched to favicon_link_tag (https://apidock.com/rails/ActionView/Helpers/AssetTagHelper/favicon_link_tag)
- updated installation docs
- moved all public assets (javascript, stylesheets, images and themes) from public to app/assets

I'm wondering if it's a good idea to split our assets from the one owned by outside entities, such as jQuery, jQuery UI or Tribute and move all these to vendor/javascripts and vendor/stylesheets.

I've been using a non-standard theme(e.g., https://github.com/farend/redmine_theme_farend_basic , https://github.com/farend/redmine_theme_farend_bleuclair) with Redmine, and after the changes in [r22696](#), I needed to do the following to keep the theme working:

1. Move the theme from public/themes to app/assets/themes.
2. Change the path where the theme imports Redmine's application.css. (Refer to the commit: https://github.com/farend/redmine_theme_farend_bleuclair/commit/c06330c064d19b02ea3bb6ca7e9a5c37d6c33519)

However, making the change in step 2 causes the theme to break in earlier versions of Redmine, so I'm looking for a way to make the change compatible with both versions.

Additionally, when I added the standard theme to app/assets/themes, it became a target for git staging. I would appreciate a change in .gitignore to prevent this.

```
diff --git a/.gitignore b/.gitignore
index f25b049a5..f38d49c39 100644
--- a/.gitignore
+++ b/.gitignore
@@ -27,10 +27,10 @@
 /public/assets/*
 /public/dispatch.*
 /public/plugin_assets/*
-/public/themes/*
-!/public/themes/alternate
-!/public/themes/classic
-!/public/themes/README
+/app/assets/themes/*
+!/app/assets/themes/alternate
+!/app/assets/themes/classic
+!/app/assets/themes/README
 /tmp/*
 /tmp/cache/*
 /tmp/pdf/*
```

#20 - 2024-02-10 11:14 - Marius BĂLTEANU

Mizuki ISHIKAWA wrote in [#note-19](#):

Marius BĂLTEANU wrote in [#note-18](#):

[...]

I've been using a non-standard theme(e.g., https://github.com/farend/redmine_theme_farend_basic , https://github.com/farend/redmine_theme_farend_bleuclair) with Redmine, and after the changes in [r22696](#), I needed to do the following to keep the theme working:

1. Move the theme from public/themes to app/assets/themes.
2. Change the path where the theme imports Redmine's application.css. (Refer to the commit: https://github.com/farend/redmine_theme_farend_bleuclair/commit/c06330c064d19b02ea3bb6ca7e9a5c37d6c33519)

Yes, both changes are required. The first change could be avoided if we keep the themes folder in public/themes, but I find very confusing to have the sources and the assets in the same public directory.

However, making the change in step 2 causes the theme to break in earlier versions of Redmine, so I'm looking for a way to make the change compatible with both versions.

Indeed, I don't see right now a way to keep them compatible, but I will investigate more in the following days.

Additionally, when I added the standard theme to app/assets/themes, it became a target for git staging. I would appreciate a change in .gitignore to prevent this.

[...]

I will update the ignore files for git, mercurial and svn, thanks!

Regarding the directory for non-standard themes, what do you think if we move them to themes in the root directory as we have for plugins? The structure will be the following:

```
app/assets/themes
+- alternate/
+- classic/

themes/
+- <theme name>/
  |
  +- favicon/
  |   +- <favicon file> (e.g. favicon.ico, favicon.png)
  |
  +- javascripts/
  |   +- theme.js
  |
  +- stylesheets/
  |   +- application.css
```

In this way, when you install Redmine and you want to customize it, you work only with themes and plugins directories, you don't need to browse the

internal structure from app.

#21 - 2024-02-10 11:31 - Marius BĂLTEANU

- Related to Patch #40221: Update wiki content related to how to create a custom theme added

#22 - 2024-02-16 01:17 - Mizuki ISHIKAWA

Marius BĂLTEANU wrote in [#note-20](#):

Regarding the directory for non-standard themes, what do you think if we move them to themes in the root directory as we have for plugins? The structure will be the following:
[...]

In this way, when you install Redmine and you want to customize it, you work only with themes and plugins directories, you don't need to browse the internal structure from app.

I agree. Having a similar structure to the plugins directory ensures consistency and should make it easier to understand for both new users and existing users.

#23 - 2024-02-26 00:10 - Marius BĂLTEANU

- Category set to Code cleanup/refactoring
- Status changed from New to Resolved
- Resolution set to Fixed

Now that [#40137](#) is finished, I think we can consider this feature done from developing point of view. I will keep this ticket as Resolved until we prepare a detailed changelog and a wiki page.

Please open new issue if you find any problem!

#24 - 2024-04-16 21:00 - Marius BĂLTEANU

- Related to Patch #40559: Fix incorrect icon image paths for Wiki help pages added

#25 - 2024-04-16 21:25 - Marius BĂLTEANU

- Status changed from Resolved to Closed

I think we can consider this feature done, please open new tickets in case of issues.

Thanks again to everyone who contributed to this change.

Files

0001-Add-propshaft-to-enable-asset-pipeline.patch	22.4 KB	2023-09-24	Takashi Kato
0002-Remove-codes-that-is-no-longer-needed-as-a-result-of.patch	5.13 KB	2023-09-24	Takashi Kato
0003-Fix-tests.patch	9.9 KB	2023-09-24	Takashi Kato
0004-Use-bundled-rails-ujs.patch	274 KB	2024-01-01	Marius BĂLTEANU
0001v2-Add-propshaft-to-enable-asset-pipeline.patch	22.3 KB	2024-01-02	Takashi Kato