reVu

> Warning
> Please, note that these documentation pages are frozen for Intellij IDEA Contest.
> You will find updated reVu site on http://idea-revu.googlecode.com

Introduction

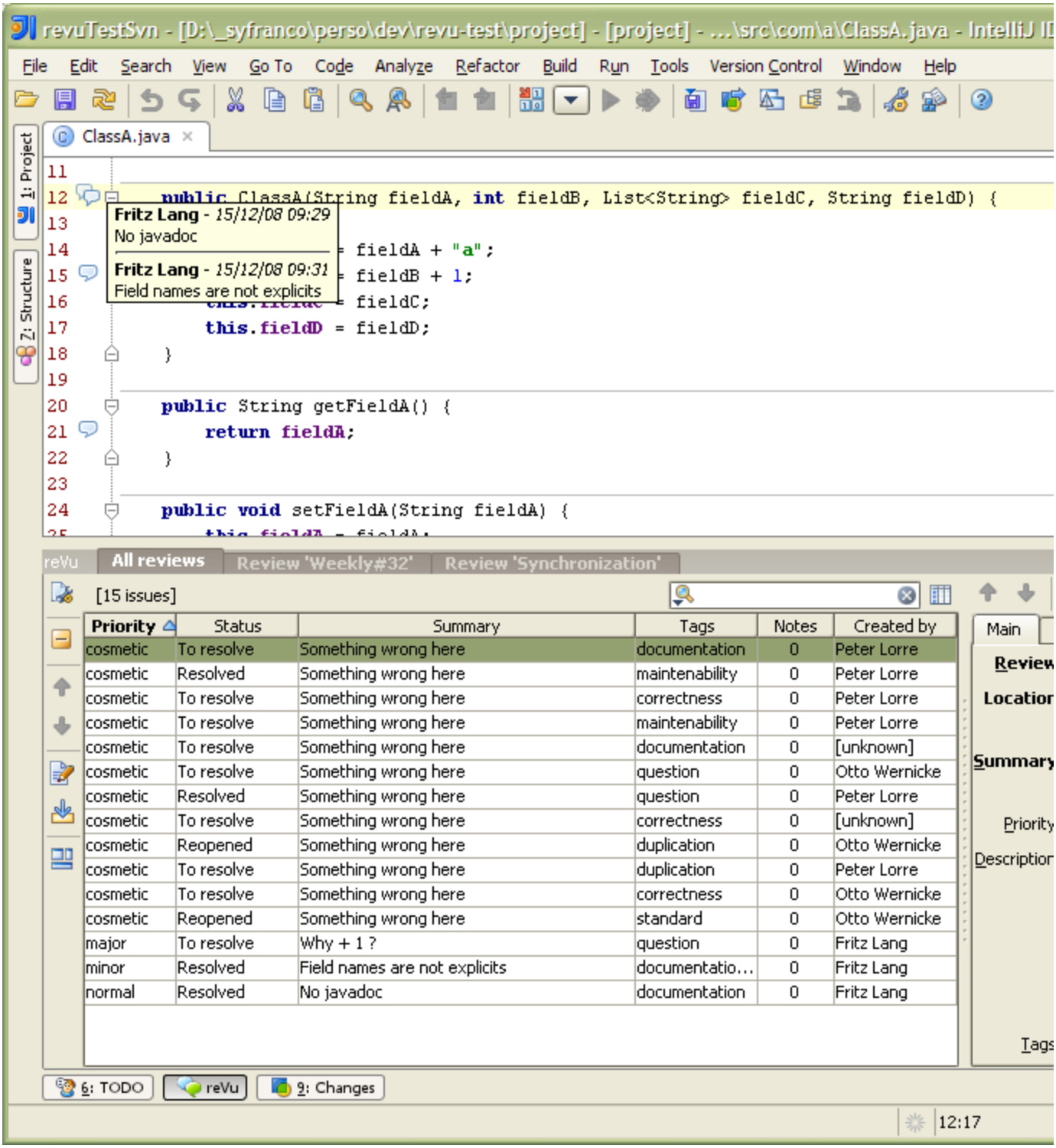
What is it ?

reVu is a plugin for Intellij IDEA which helps to perform team code reviews or simply annotate your code. Intellij IDEA is great for automatic code inspections, but not at the point to make team code reviews useless

There's much literature about peer code review process, I'll just mention main advantages for me:

- each team member knows better application code
- code becomes more homogeneous
- code is usually better first because reviewers may find potential issues or improvements, but also because authors pay more attention when they know their code will be reviewed

Some tools already exist, but they are either commercial (see Crucible from Atlassian) or are not tied to IDEA (GoogleCode provides a review process, Eclipse and Netbeans have their code review plugin, ...) whereas IDE integration is a key feature.



Key features

- very simple
  - no server required. Review definitions are stored in XML files managed through your Version Control System.
- flexible. You are able to customize:
  - users
  - priorities
  - custom tags used to classify tags freely
- keep original source line
  - if reviewed code is changed or deleted, reVu will keep issue and will display it as desynchronized
- VCS (Version Control System) integration

    - issues are attached to VCS version (if code is under VCS of course)
- templates management
    - review definitions may be built from review templates which you can create according to your needs
- private / shared reviews
    - each user may create its own reviews and decide to share them to team

Installation

IntellijIDEA provides a plugin manager which allows user to download plugins from IDE. Plugin manager is available from Settings dialog. No further step is required.

Review process

- A review may be created by anyone. Creator becomes an administrator for this review.
- Administrator configures review: name, goals, priorities, tags, users, ...
- When adding users to review, administrator configures their roles for this review : administrator, reviewer or author.
- Reviewers inspect code and add issues
- Authors fix code and resolve issues
- Reviewers may check resolved issues and close them
- Administrator may deactivate or remove review at any time

Creating review

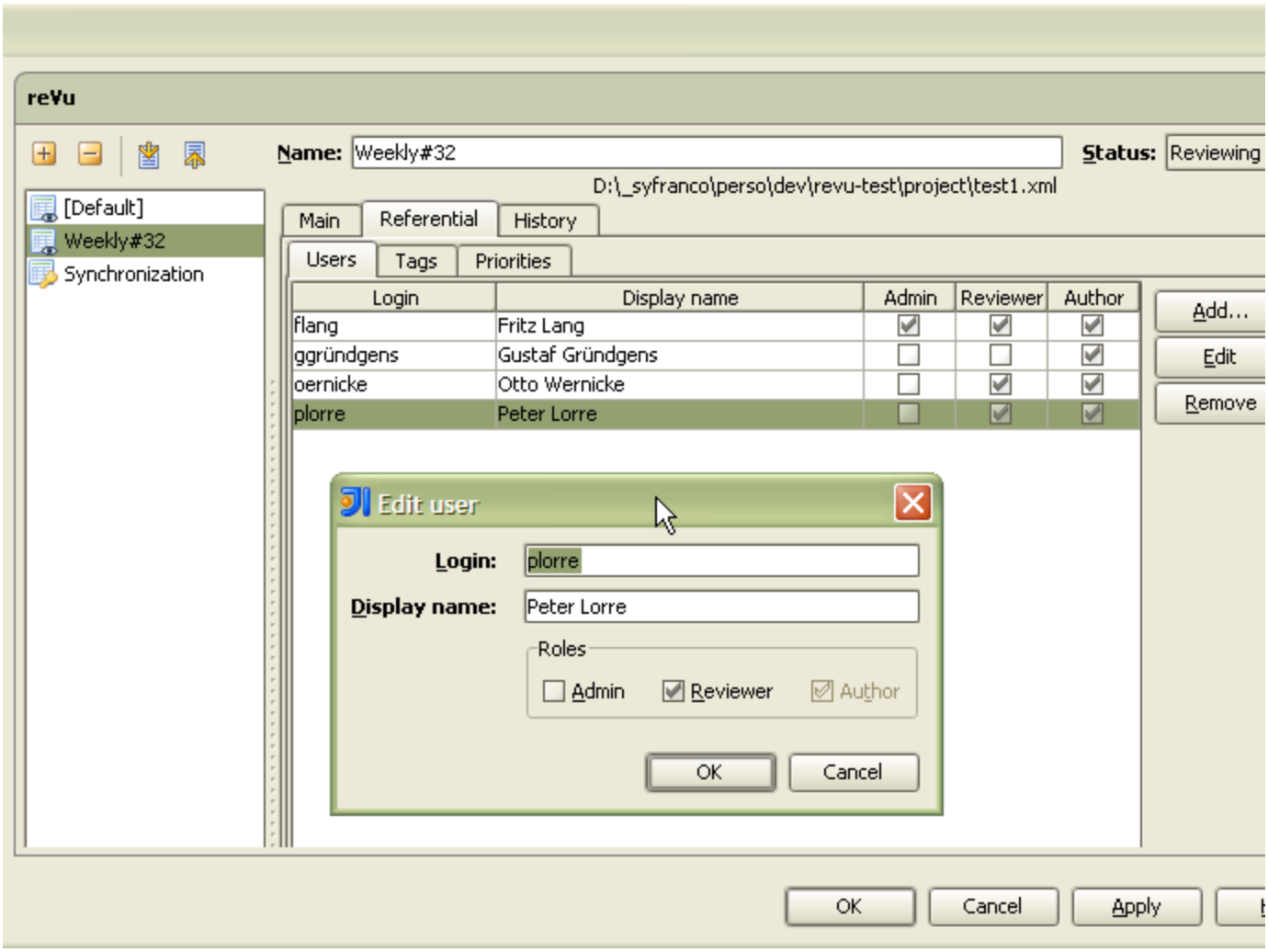
You create reviews from reVu project settings (which may be accessed directly from reVu tool window). A review is identified by its unique name. This name is used to create XML file which stores review definition.

When you create a new review, you have to choose if this review will be shared with others or will be local for you. A shared review is referenced from project settings file, whereas a local review is referenced from workspace settings file. But you will be able to change this setting using Share checkbox at any time.

A review definition is composed of:

- a goal
- users
- priorities
- classification tags

Users, priorities and tags are defined in a referential. You'll probably use similar referentials between your reviews. In this case, you should create a review template with your custom referential and copy or extend it for your you reviews. Extend a template means that changes performed upon template will be replicated for reviews extending it, whereas copy makes the review independant from template.

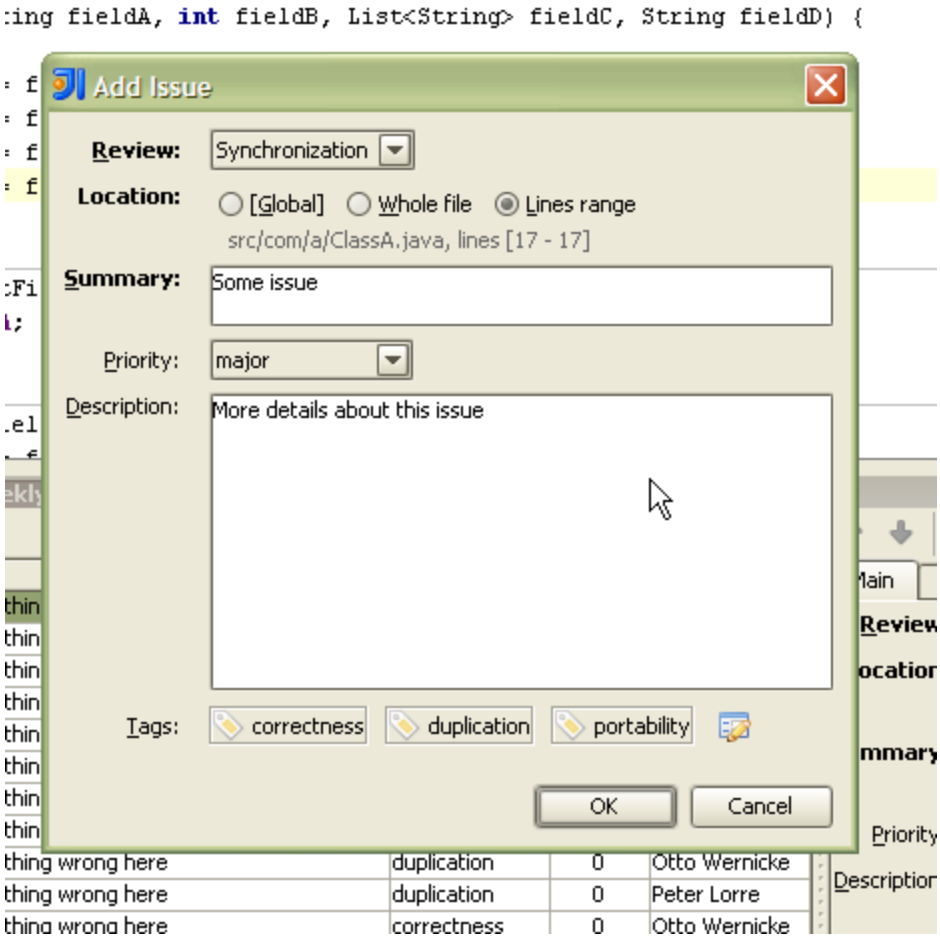


Status combo defines current status of review. Administrator is in charge of changing this status according to current phase of your review:

- Draft: review is being prepared
- Reviewing: code is being inspected by reviewers
- Fixing: issues are being fixed by authors
- Closed: review is finished
- [Template]: special status for reviews only used as template

Adding issues

Just point on reviewed code and invoke Add Issue action (by default: Alt I shortcut and displayed in gutter popup menu). Issue is attached to selected ccde: if code is changed, issue is marked as desynchronized but is still available (and you'll still be able to preview initial code at any time).



Write a summary for this item. Optionally, you may add information such as priority, or tags.

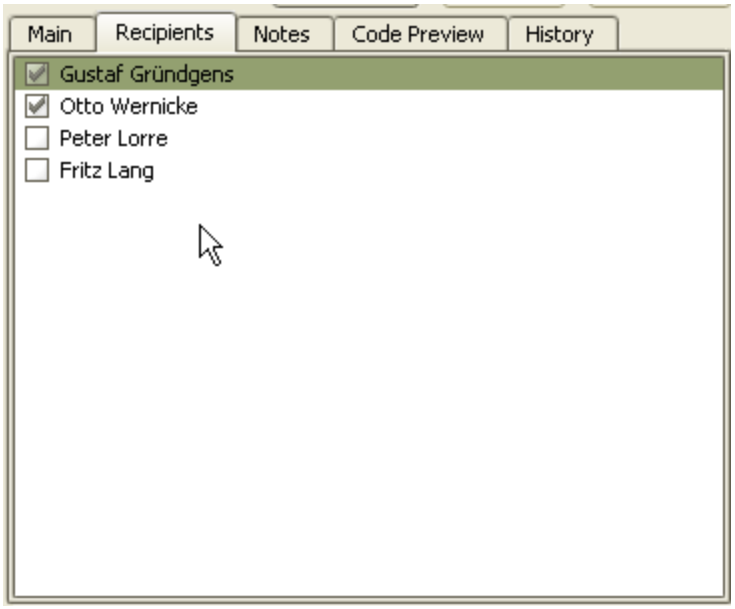
Tags allow you to categorize issues according your needs. Usually, you'll use tags to set type of issue: coding standard, optimization, i18n, doc, ... But you can use them for additional classification: related business layer, resolution complexity, ...

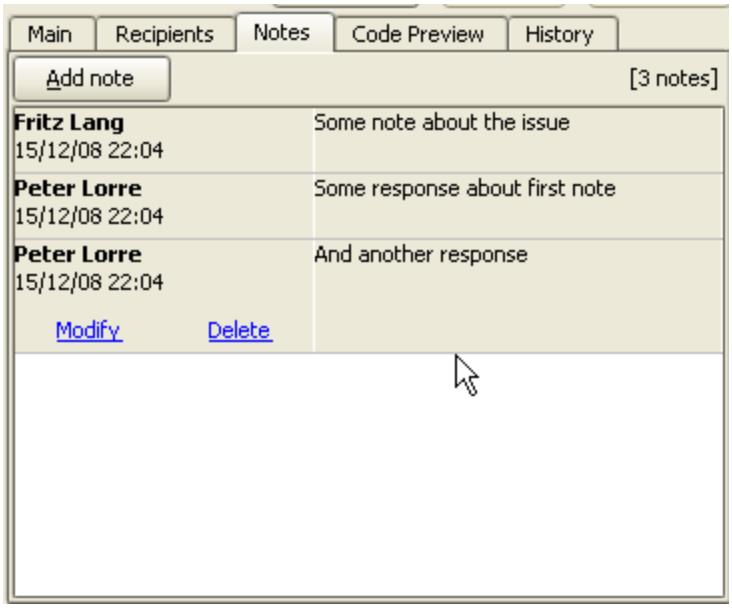
Managing existing issues

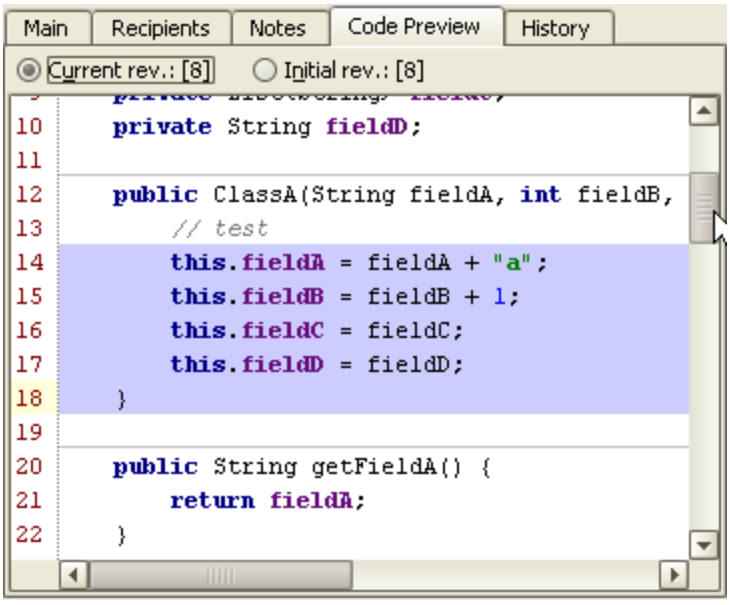
The browsing tool window allows to navigate and update existing issues. There's one tab for each active review (i.e. reviewing or fixing) plus a tab gathering all issues.

Depending on your role, you are able to update issue information and change issue status : resolved, closed, reopened.

You can also configure issue recipients, i.e. author(s) which should fix issues, adding free notes and preview reviewed code. When no recipient is specified, it means all authors defined for review are concerned.


Main
Recipients
Notes
Code Preview
History
Gustaf Gründgens
Otto Wernicke
Peter Lorre
Fritz Lang


Main
Recipients
Notes
Code Preview
History
Add note
[3 notes]
Fritz Lang
15/12/08 22:04
Some note about the issue
Peter Lorre
15/12/08 22:04
Some response about first note
Peter Lorre
15/12/08 22:04
And another response
Modify
Delete

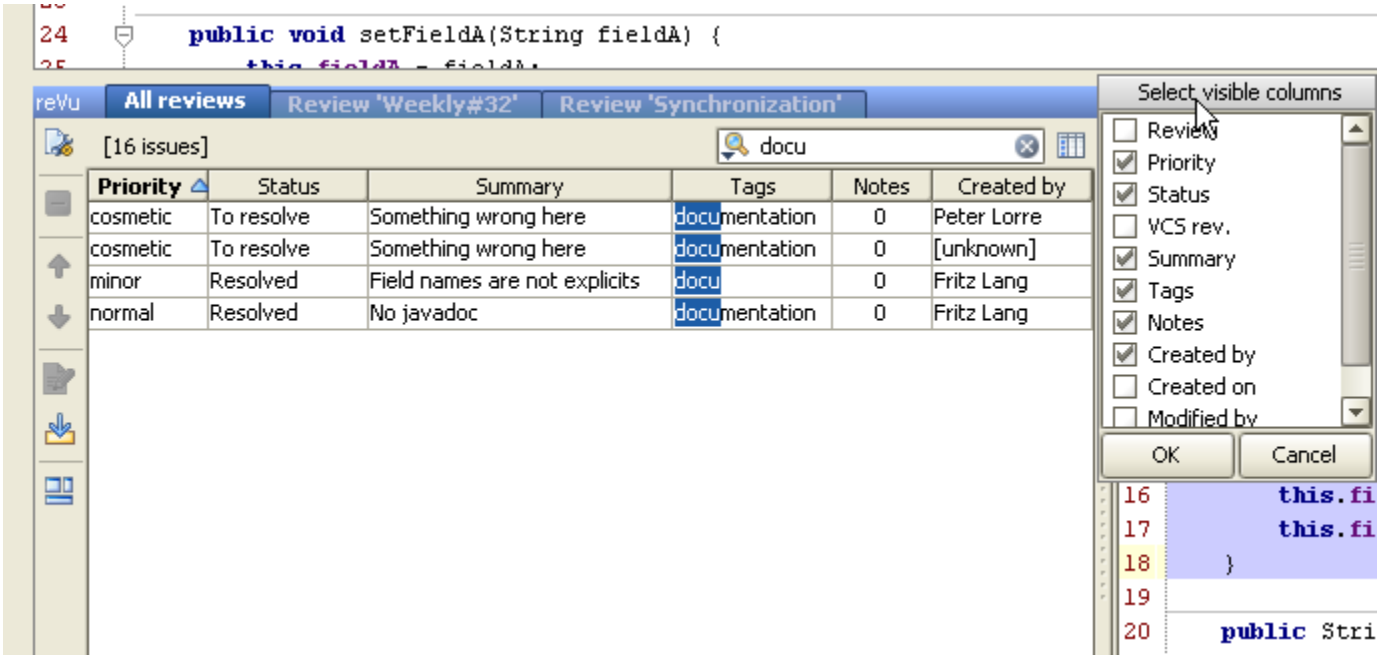

Main
Recipients
Notes
Code Preview
History
Current rev.: [8]
Initial rev.: [8]
10 private String fieldD;
11
12 public ClassA(String fieldA, int fieldB,
13 // test
14 this.fieldA = fieldA + "a";
15 this.fieldB = fieldB + 1;
16 this.fieldC = fieldC;
17 this.fieldD = fieldD;
18 }
19
20 public String getFieldA() {
21 return fieldA;
22 }

Issue table

Issue table is searchable through 2 ways:

- using IDEA speed search: just type some text on table, table will scroll to first issue whose summary match your text
- using search field above table: this filter applies to all visible columns. Only rows with at least one cell matchs your search will be visible and searched text will be highlighted.

Issue table is also sortable and you can choose visible columns using column selector button next to search field.



Scope

reVu adds also a scope for each active review. Theses scopes contain files which have at least one issue.

In future release, there could be scopes containing all files to review.

VCS integration

Issues are attached to the VCS revision of reviewed file. This allows later to retrieve original code when it has been changed between reviewing phase and fixing phase. The Preview tab in issue form shows the current preview and the original preview (when available).

NB: one key feature which is planned for future releases is to provide a mechanism for reviewers to work on diff between a reference VCS revision and a current revision so they may concentrate on changes.

Storage

Review definitions are stored in XML files. These files should be saved in your project tree for 2 reasons:

- Others should be able to access them
- Review files should be added to VCS

If you store shared reviews outside from project, others will have to access them using same file paths (e.g. a shared directory).

Warning
When a review is shared and several users work at the same on this review, usually its XML file must be merged. Most of time, there won't be any conflict, but sometimes you'll have to perform merge. It's the price to pay for having a such simple mechanism A custom merge resolution mechanism might be added in future releases.

If you find this issue critical, you should consider using a server based tool such as Crucible from Atlassian, see http://www.atlassian.com/software/crucible. I don't know it, but Atlassian usually makes great products.

Security

Each review has its own user list (but remember you should create a review template with your default users). User roles are specific to reviews: a user may be defined as reviewer in a review and author in another one.

Users are identified by a login and shown in UI through a display name. User must define their login using reVu application settings so they can be identified in reviews. Otherwise, an alert is shown and user can't use reVu.

> Password management
> reVu does not manage passwords. Since user information are stored in XML files, each one could alter easily these passwords, even if they are crypted. Or it would required some protection which would degrade usability

Future plan

Some features I'd like to implement in future releases :

- a tree navigation pane
- assign files to review from VCS revision or date and making view of changes easier
- multi-location on issues
- custom UI for review file merges
- CSV export
- create issues from intentions
- more detailed history on review or issues
- ...

Found a bug, want new feature ?

Please use Issue tracker.

More information is available on reVu site: http://idea-revu.googlecode.com

Icons come from famous FAMFAMFAM Silk icon library: http://www.famfamfam.com/lab/icons/silk/