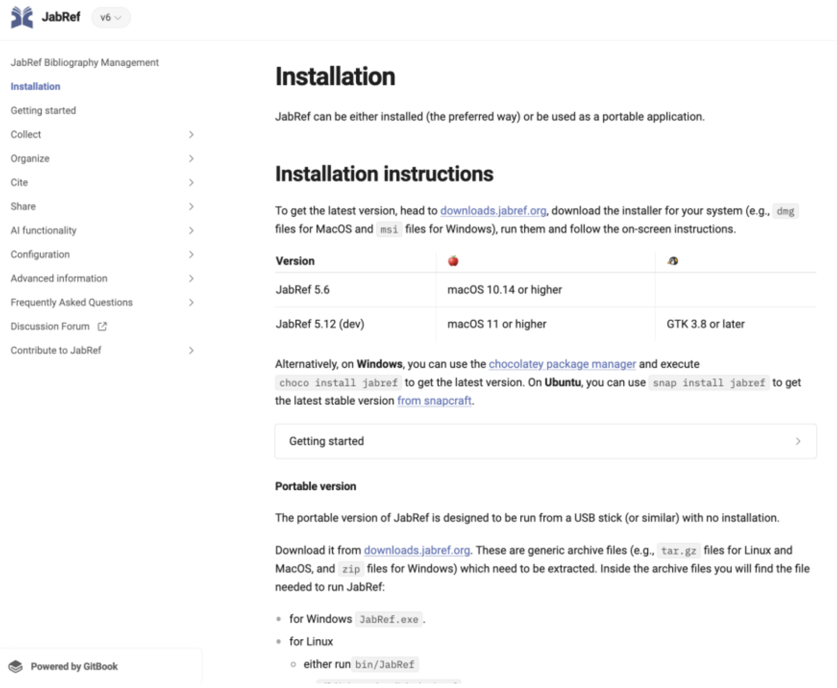
Activity 1 - OSS Project Selection

Selecting a Repository

After discussing with my tutor, we decided to work on the JabRef repository (https://github.com/JabRef/jabref). The selection process involved evaluating multiple repositories to find one that met the following criteria:

1. **Good Documentation and Setup Instructions**: The JabRef repository provided clear and well-documented setup/installation instructions, which made it easier to get started.



Installation

JabRef can be either installed (the preferred way) or be used as a portable application.

Installation instructions

To get the latest version, head to downloads.jabref.org, download the installer for your system (e.g., `dmg` files for MacOS and `msi` files for Windows), run them and follow the on-screen instructions.

Version	🍎	🐧
JabRef 5.6	macOS 10.14 or higher	
JabRef 5.12 (dev)	macOS 11 or higher	GTK 3.8 or later

Alternatively, on **Windows**, you can use the chocolatey package manager and execute `choco install jabref` to get the latest version. On **Ubuntu**, you can use `snap install jabref` to get the latest stable version from snapcraft.

Getting started

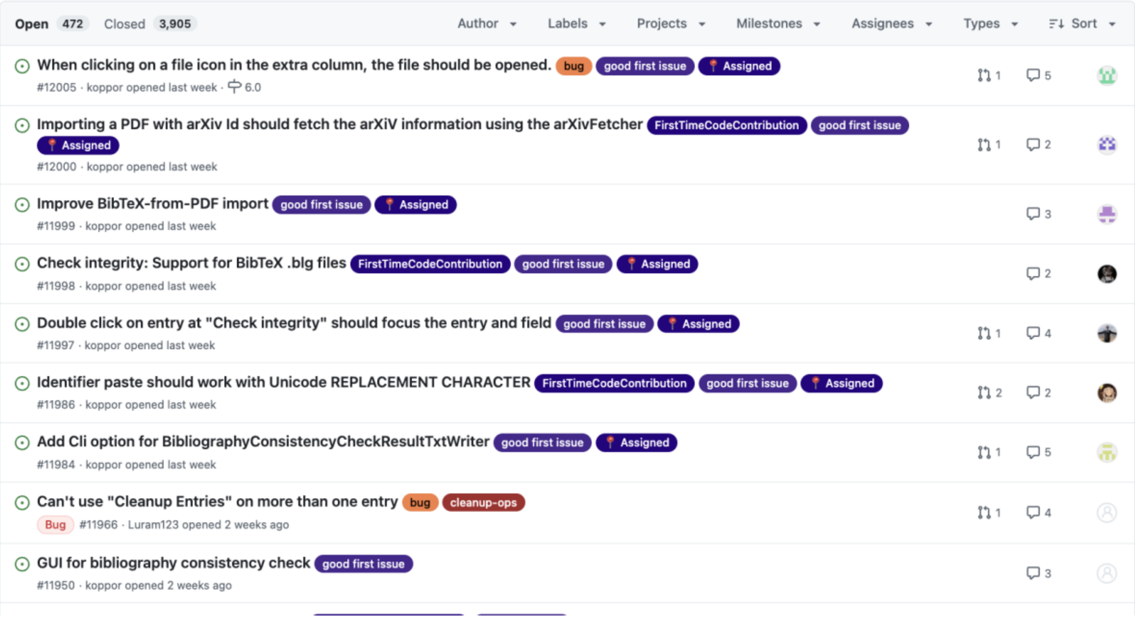
Portable version

The portable version of JabRef is designed to be run from a USB stick (or similar) with no installation.

Download it from downloads.jabref.org. These are generic archive files (e.g., `tar.gz` files for Linux and MacOS, and `zip` files for Windows) which need to be extracted. Inside the archive files you will find the file needed to run JabRef:

- for Windows `JabRef.exe`.
- for Linux
  - either run `bin/JabRef`

2. **Active Repository**: We ensured that the repository was active, with recent commits and issues, to avoid challenges with inactive maintainers. This was confirmed by checking the commit history and the activity on open issues. An active repository meant I would have a better chance of receiving timely feedback from maintainers.



Open 472 Closed 3,905

When clicking on a file icon in the extra column, the file should be opened. bug good first issue Assigned
#12005 · koppor opened last week · 6.0 — 1 — 5

Importing a PDF with arXiv Id should fetch the arXiV information using the arXivFetcher FirstTimeCodeContribution good first issue Assigned
#12000 · koppor opened last week — 1 — 2

Improve BibTeX-from-PDF import good first issue Assigned
#11999 · koppor opened last week — 3

Check integrity: Support for BibTeX .blg files FirstTimeCodeContribution good first issue Assigned
#11998 · koppor opened last week — 2

Double click on entry at "Check integrity" should focus the entry and field good first issue Assigned
#11997 · koppor opened last week — 1 — 4

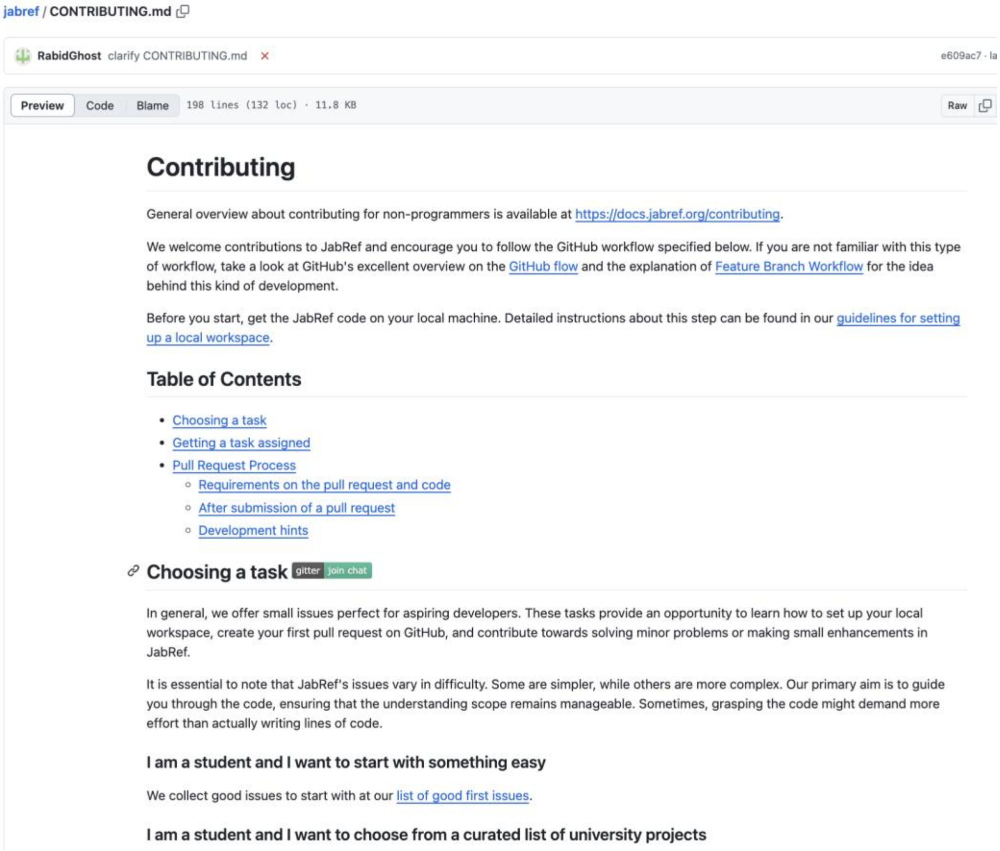
Identifier paste should work with Unicode REPLACEMENT CHARACTER FirstTimeCodeContribution good first issue Assigned
#11986 · koppor opened last week — 2 — 2

Add Cli option for BibliographyConsistencyCheckResultTxtWriter good first issue Assigned
#11984 · koppor opened last week — 1 — 5

Can't use "Cleanup Entries" on more than one entry bug cleanup-ops
Bug #11966 · Luram123 opened 2 weeks ago — 1 — 4

GUI for bibliography consistency check good first issue
#11950 · koppor opened 2 weeks ago — 3

3. **Understandable Codebase**: I reviewed the repository and found it to be straightforward and understandable, with a clear structure that made it easy to navigate. The documentation was comprehensive, and the issues were well-defined, which allowed me to effectively contribute.



jabref / CONTRIBUTING.md

RabidGhost clarify CONTRIBUTING.md ✕ e609ac7 · la

198 lines (132 loc) · 11.8 KB

Contributing

General overview about contributing for non-programmers is available at https://docs.jabref.org/contributing.

We welcome contributions to JabRef and encourage you to follow the GitHub workflow specified below. If you are not familiar with this type of workflow, take a look at GitHub's excellent overview on the GitHub flow and the explanation of Feature Branch Workflow for the idea behind this kind of development.

Before you start, get the JabRef code on your local machine. Detailed instructions about this step can be found in our guidelines for setting up a local workspace.

Table of Contents

- Choosing a task
- Getting a task assigned
- Pull Request Process
  - Requirements on the pull request and code
  - After submission of a pull request
  - Development hints

Choosing a task

In general, we offer small issues perfect for aspiring developers. These tasks provide an opportunity to learn how to set up your local workspace, create your first pull request on GitHub, and contribute towards solving minor problems or making small enhancements in JabRef.

It is essential to note that JabRef's issues vary in difficulty. Some are simpler, while others are more complex. Our primary aim is to guide you through the code, ensuring that the understanding scope remains manageable. Sometimes, grasping the code might demand more effort than actually writing lines of code.

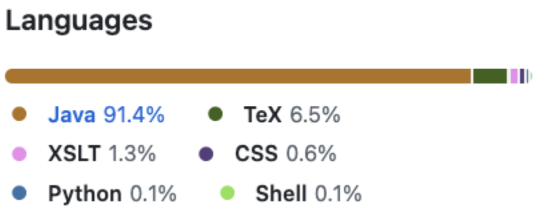
I am a student and I want to start with something easy

We collect good issues to start with at our list of good first issues.

I am a student and I want to choose from a curated list of university projects

4. **Language Familiarity**: The repository's content was in a language I was comfortable

with—Markdown for documentation updates—ensuring I didn't have to learn a new language while working on the project.



How Do We Choose Issues #10557

- **Difficulty**: This issue presents a moderate level of technical challenge. It requires understanding JabRef's data storage structure, directory management, and Windows file path handling. Additionally, knowledge of CSV file processing and user preference implementation is necessary, but these concepts are well-documented and approachable.
- **Size**: The scope of this task is medium-sized, encompassing several key components. The work involves creating a new directory structure, implementing file auto-detection and conversion functionality, modifying the preference dialog, and updating journal abbreviation-related components. Each component is well-defined and modular in nature.
- **Time**: I estimate this issue will take approximately 8-10 hours to complete. This includes time for understanding the existing codebase, implementing core functionality, testing, and review processes. The timeline accounts for potential challenges in integration and ensuring backward compatibility.
- **Workload**: The workload is focused and manageable, consisting primarily of implementing directory management logic, file handling features, and user interface updates. Additional tasks include collaboration with team members for code review, ensuring adherence to project standards, and maintaining documentation. The clear requirements and modular nature of the changes make this workload structured and achievable.

How Do We Choose Issues #468

- **Difficulty**: The issue was quite easy. It involved understanding the structure of the documentation and locating the corresponding changes in the software.

- **Size**: The issue was relatively small, involving minor updates to two documentation files. It did not require large-scale changes to the codebase.

- **Time**: I anticipated needing around 1 hour to fully understand the context, make the necessary changes, and verify their accuracy. This included time for collaboration and review.

- **Workload**: The workload was manageable, mainly involving updating text and images in the documentation. The potential workload included collaborating with my group members for reviews and ensuring that the changes met the repository's standards.

How Do We Choose Issues #549: Eric Liehne

- **Difficulty**: This issue was not too hard for me. It involved understanding the structure of the software and how functions were handled, and a decent knowledge of Java’s functions and classes.

- **Size**: The issue was somewhat significant in size. It involved updating multiple functions throughout the repository, as well as adding new tests to ensure that I was writing something that worked.

- **Time**: I expected this issue to take around 6 hours to implement, including understanding the context, making necessary changes, and confirming my work through implementing tests.

- **Workload**: The workload was manageable, entirely focused on updating functions and adding tests to check the validity of my work. The potential workload included collaborating with my group members for reviews and ensuring that the changes met the repository’s standards.

How Do We Choose Issues #11316

- **Difficulty:** The issue was moderately complex. It involved implementing a new feature—automatic renaming of linked files when entry data changes—which required understanding and integrating with existing file handling and pattern parsing mechanisms in the application.
- **Size:** The issue was of medium size. It was more than a simple bug fix but less than a large feature addition. It required adding new functionality, modifying several classes, and ensuring consistency across the application.
- **Time:** I anticipated needing around 15 hours to complete the issue, including

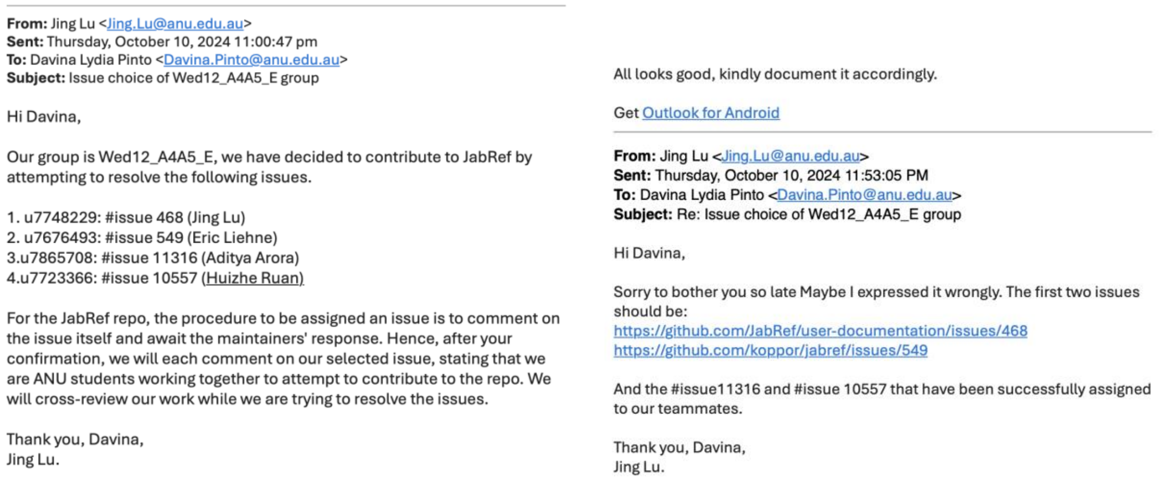
understanding the codebase, implementation, and testing.

- **Workload:** The workload was manageable and included designing the checkbox to update user preferences, implementing the automatic renaming feature, integrating it with existing code to utilize current file naming patterns, refactoring to eliminate code duplication and testing the new functionality.

Activity 2 - Working on the OSS Project

Checking with our tutor

We sent an email to the tutor to confirm whether the issue we selected meets the requirements. The picture is a screenshot of the agreement.



From: Jing Lu <Jing.Lu@anu.edu.au>
Sent: Thursday, October 10, 2024 11:00:47 pm
To: Davina Lydia Pinto <Davina.Pinto@anu.edu.au>
Subject: Issue choice of Wed12_A4A5_E group

Hi Davina,

Our group is Wed12_A4A5_E, we have decided to contribute to JabRef by attempting to resolve the following issues.

1. u7748229: #issue 468 (Jing Lu)
2. u7676493: #issue 549 (Eric Liehne)
3.u7865708: #issue 11316 (Aditya Arora)
4.u7723366: #issue 10557 (Huizhe Ruan)

For the JabRef repo, the procedure to be assigned an issue is to comment on the issue itself and await the maintainers' response. Hence, after your confirmation, we will each comment on our selected issue, stating that we are ANU students working together to attempt to contribute to the repo. We will cross-review our work while we are trying to resolve the issues.

Thank you, Davina,
Jing Lu.

All looks good, kindly document it accordingly.

Get Outlook for Android

From: Jing Lu <Jing.Lu@anu.edu.au>
Sent: Thursday, October 10, 2024 11:53:05 PM
To: Davina Lydia Pinto <Davina.Pinto@anu.edu.au>
Subject: Re: Issue choice of Wed12_A4A5_E group

Hi Davina,

Sorry to bother you so late Maybe I expressed it wrongly. The first two issues should be:
https://github.com/JabRef/user-documentation/issues/468
https://github.com/koppor/jabref/issues/549

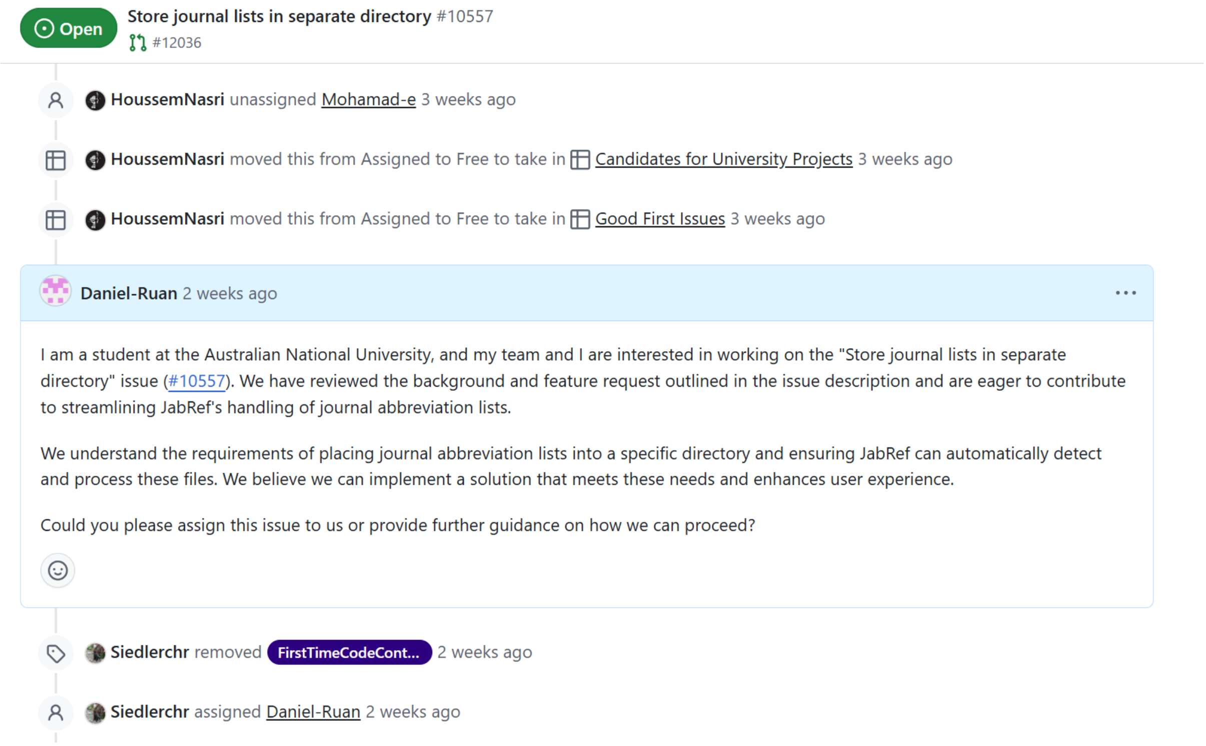
And the #issue11316 and #issue 10557 that have been successfully assigned to our teammates.

Thank you, Davina,
Jing Lu.

Issue #10557:

https://github.com/JabRef/jabref/issues/10557

After selecting issue 10557, I immediately sent a request to the maintainer in the comment section and got assigned.



Open Store journal lists in separate directory #10557
#12036

HoussemNasri unassigned Mohamad-e 3 weeks ago

HoussemNasri moved this from Assigned to Free to take in Candidates for University Projects 3 weeks ago

HoussemNasri moved this from Assigned to Free to take in Good First Issues 3 weeks ago

Daniel-Ruan 2 weeks ago

I am a student at the Australian National University, and my team and I are interested in working on the "Store journal lists in separate directory" issue (#10557). We have reviewed the background and feature request outlined in the issue description and are eager to contribute to streamlining JabRef's handling of journal abbreviation lists.

We understand the requirements of placing journal abbreviation lists into a specific directory and ensuring JabRef can automatically detect and process these files. We believe we can implement a solution that meets these needs and enhances user experience.

Could you please assign this issue to us or provide further guidance on how we can proceed?

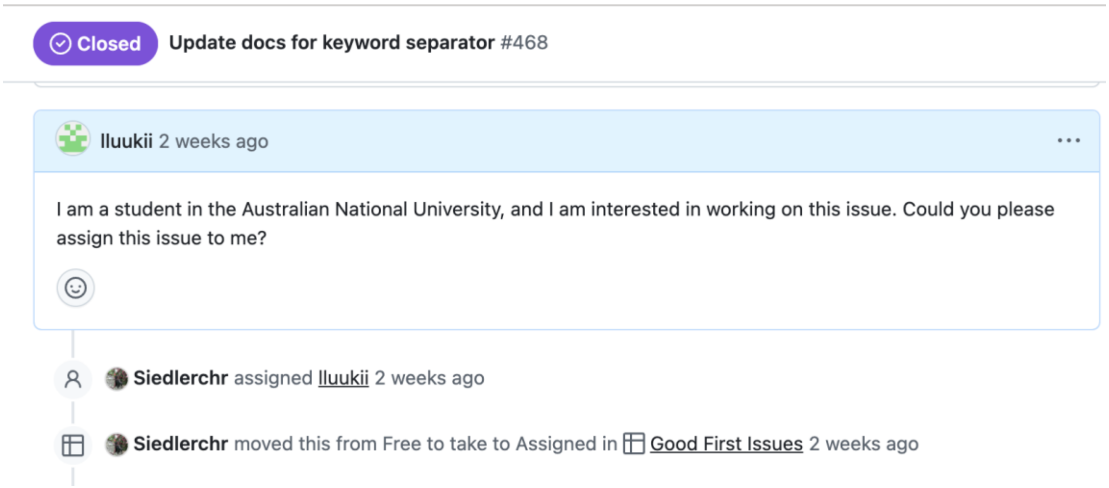
Siedlerchr removed FirstTimeCodeCont... 2 weeks ago

Siedlerchr assigned Daniel-Ruan 2 weeks ago

Issue #468:

https://github.com/JabRef/user-documentation/issues/468

After selecting issue 468, I immediately sent a request to the maintainer in the comment section and got assigned.



Closed Update docs for keyword separator #468

lluukii 2 weeks ago

I am a student in the Australian National University, and I am interested in working on this issue. Could you please assign this issue to me?

Siedlerchr assigned lluukii 2 weeks ago

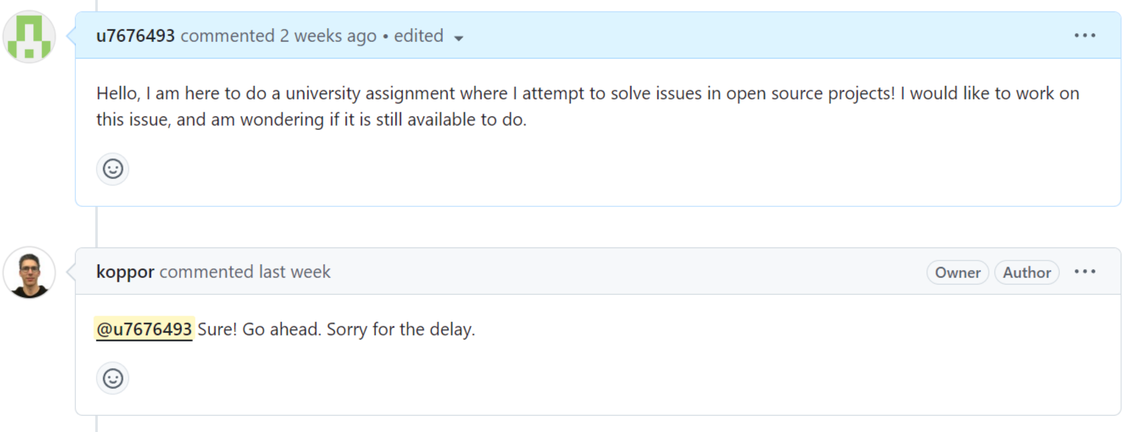
Siedlerchr moved this from Free to take to Assigned in Good First Issues 2 weeks ago

(https://github.com/JabRef/user-documentation/issues/468#issuecomment-2402233741)

Issue #549: Eric Liehne

https://github.com/koppor/jabref/issues/549

After selecting Issue 549, I sent a request to the maintainer in the comment section and my assignment was approved.



u7676493 commented 2 weeks ago • edited

Hello, I am here to do a university assignment where I attempt to solve issues in open source projects! I would like to work on this issue, and am wondering if it is still available to do.

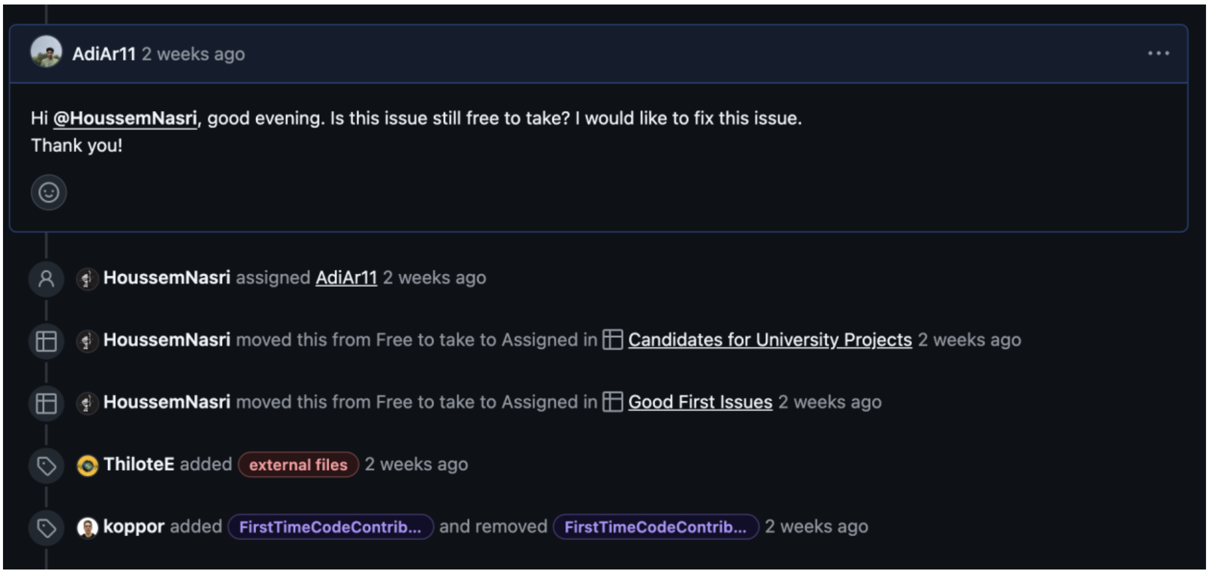
koppor commented last week

@u7676493 Sure! Go ahead. Sorry for the delay.

Issues #11316:

https://github.com/JabRef/jabref/issues/11316

After shortlisting this issue, I asked if the issue was still available to take and mentioned that I would like to work on this issue. This issue was previously assigned to other people, but they did not have any updates, so they were removed from assignee and the issue became free to take. I was later assigned to this issue.



AdiAr11 2 weeks ago

Hi @HoussemNasri, good evening. Is this issue still free to take? I would like to fix this issue.
Thank you!

HoussemNasri assigned AdiAr11 2 weeks ago

HoussemNasri moved this from Free to take to Assigned in Candidates for University Projects 2 weeks ago

HoussemNasri moved this from Free to take to Assigned in Good First Issues 2 weeks ago

ThiloteE added external files 2 weeks ago

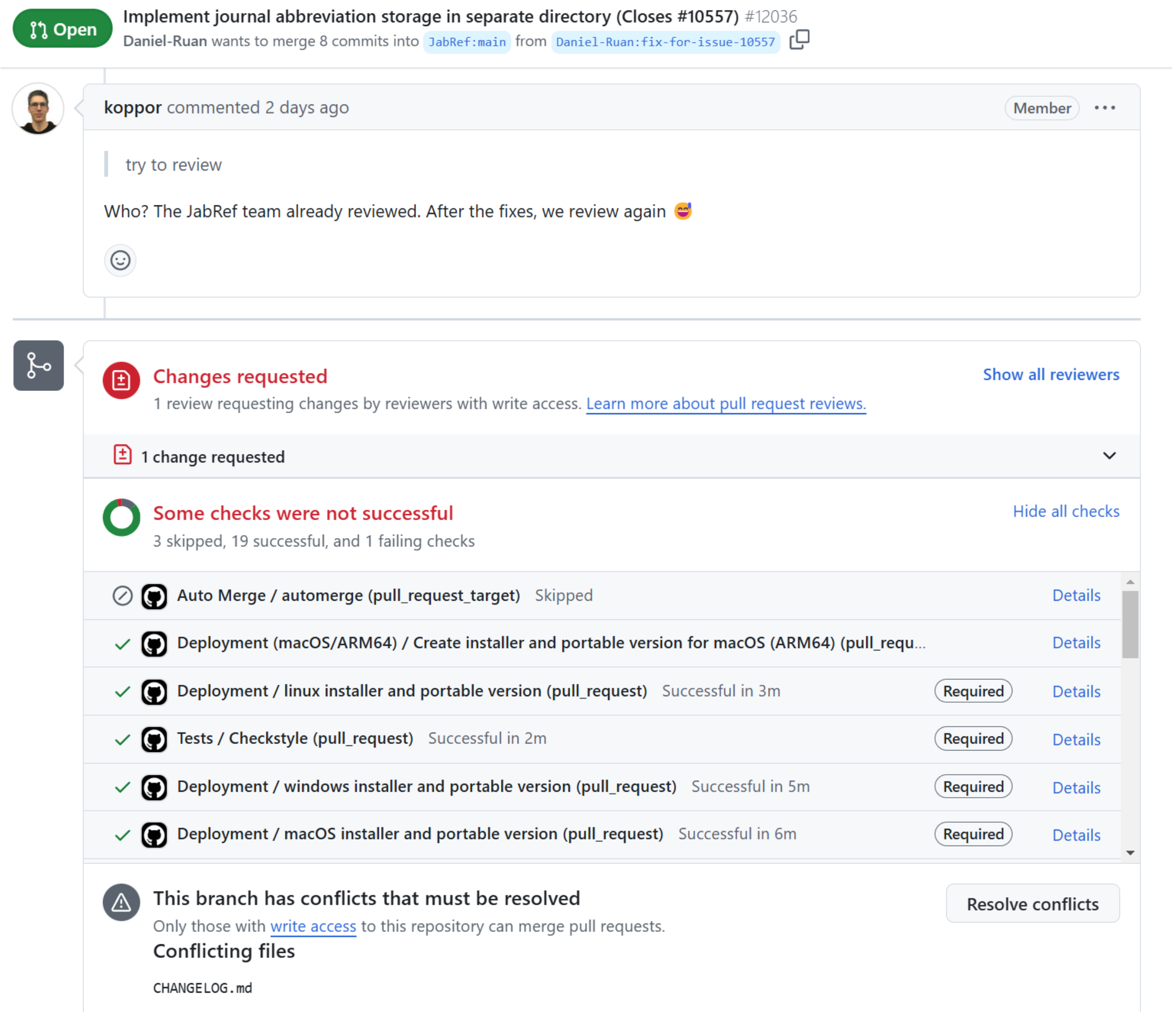
koppor added FirstTimeCodeContrib... and removed FirstTimeCodeContrib... 2 weeks ago

Activity 3 - Opening and Submit a PR to the project

Issue #10557:

Submitted the PR as required; however, it was not successfully merged because some unit tests did not pass.

https://github.com/JabRef/jabref/pull/12036



Open Implement journal abbreviation storage in separate directory (Closes #10557) #12036

Daniel-Ruan wants to merge 8 commits into `JabRef:main` from `Daniel-Ruan:fix-for-issue-10557`

koppor commented 2 days ago Member

> try to review

Who? The JabRef team already reviewed. After the fixes, we review again 😅

Changes requested Show all reviewers

1 review requesting changes by reviewers with write access. Learn more about pull request reviews.

1 change requested

Some checks were not successful Hide all checks

3 skipped, 19 successful, and 1 failing checks

⊘	Auto Merge / automerge (pull_request_target) Skipped		Details
✓	Deployment (macOS/ARM64) / Create installer and portable version for macOS (ARM64) (pull_requ...		Details
✓	Deployment / linux installer and portable version (pull_request) Successful in 3m	Required	Details
✓	Tests / Checkstyle (pull_request) Successful in 2m	Required	Details
✓	Deployment / windows installer and portable version (pull_request) Successful in 5m	Required	Details
✓	Deployment / macOS installer and portable version (pull_request) Successful in 6m	Required	Details

This branch has conflicts that must be resolved Resolve conflicts

Only those with write access to this repository can merge pull requests.

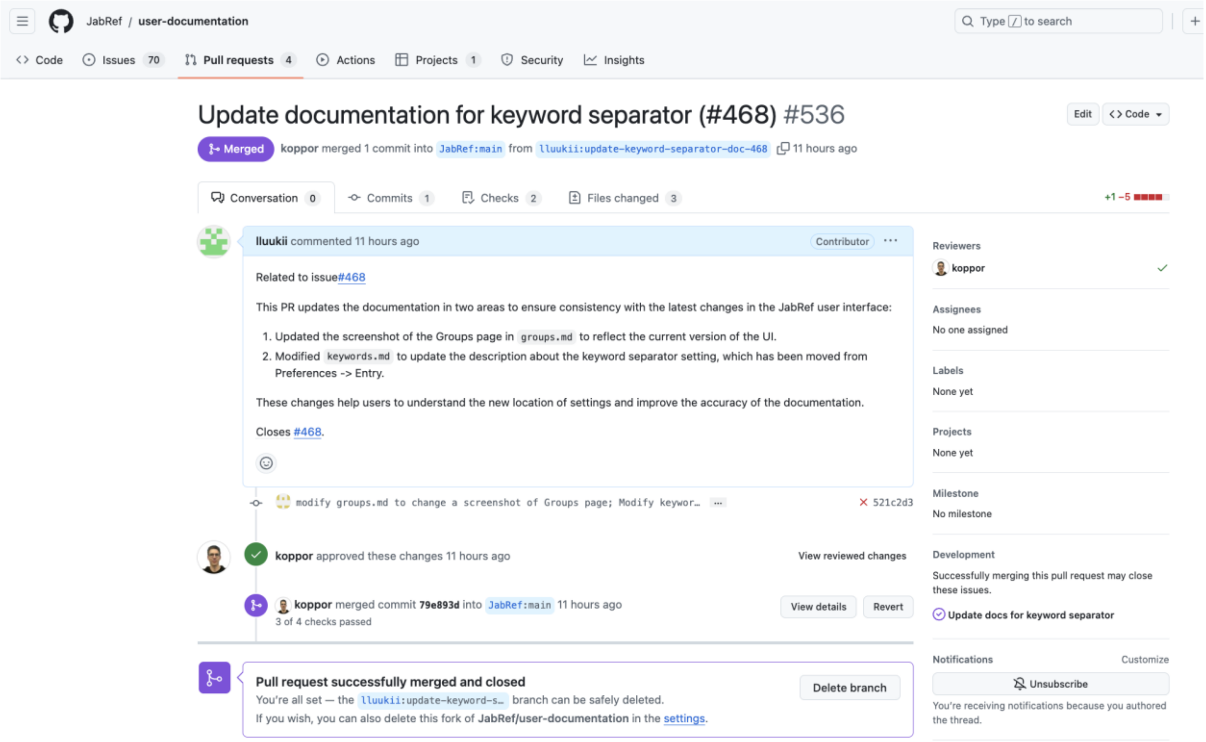
Conflicting files

`CHANGELOG.md`

Issue #468:

Submit PR as required, and it is successfully accepted.

(https://github.com/JabRef/user-documentation/pull/536)



Update documentation for keyword separator (#468) #536

koppor merged 1 commit into `JabRef:main` from `lluukii:update-keyword-separator-doc-468` 11 hours ago

lluukii commented 11 hours ago

Related to issue#468

This PR updates the documentation in two areas to ensure consistency with the latest changes in the JabRef user interface:

1. Updated the screenshot of the Groups page in `groups.md` to reflect the current version of the UI.
2. Modified `keywords.md` to update the description about the keyword separator setting, which has been moved from Preferences -> Entry.

These changes help users to understand the new location of settings and improve the accuracy of the documentation.

Closes #468.

`modify groups.md to change a screenshot of Groups page; Modify keywor…` ✗ 521c2d3

koppor approved these changes 11 hours ago

koppor merged commit **79e893d** into `JabRef:main` 11 hours ago
3 of 4 checks passed

Pull request successfully merged and closed
You're all set — the `lluukii:update-keyword-s…` branch can be safely deleted.
If you wish, you can also delete this fork of **JabRef/user-documentation** in the settings.

Reviewers
koppor ✓

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
Update docs for keyword separator

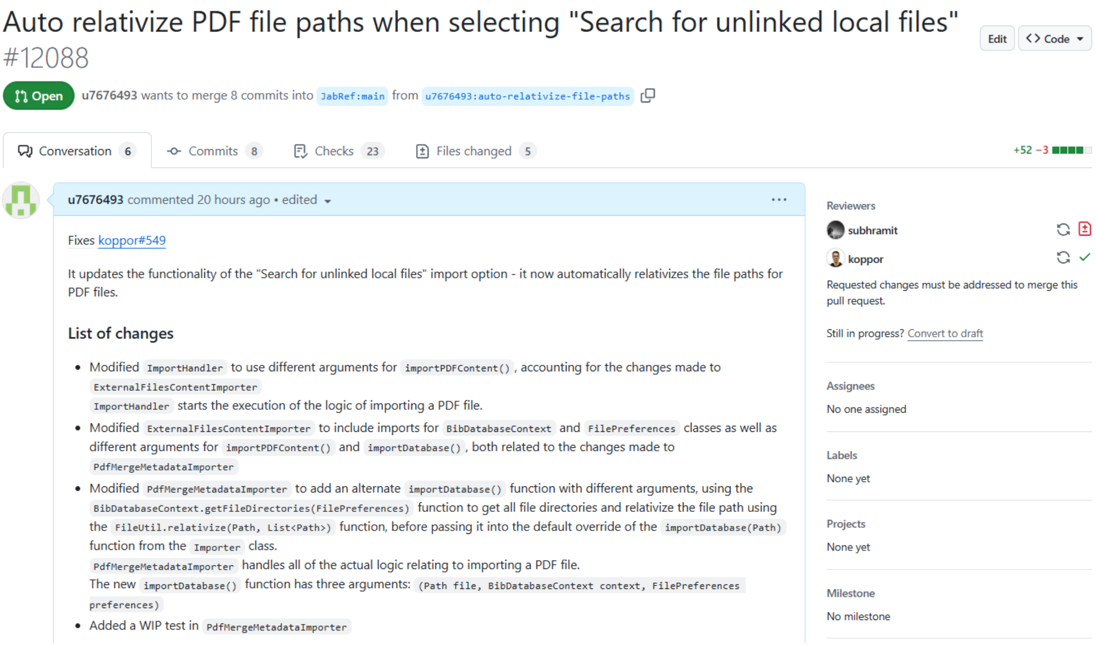
Notifications

You're receiving notifications because you authored the thread.

Issue #549: Eric Liehne

https://github.com/JabRef/jabref/pull/12088

I submitted the PR as requested, and as of 27/10, it is partially approved.



Auto relativize PDF file paths when selecting "Search for unlinked local files" #12088

u7676493 wants to merge 8 commits into `JabRef:main` from `u7676493:auto-relativize-file-paths`

u7676493 commented 20 hours ago • edited

Fixes koppor#549

It updates the functionality of the "Search for unlinked local files" import option - it now automatically relativizes the file paths for PDF files.

List of changes

- Modified `ImportHandler` to use different arguments for `importPDFContent()`, accounting for the changes made to `ExternalFilesContentImporter`
  `ImportHandler` starts the execution of the logic of importing a PDF file.
- Modified `ExternalFilesContentImporter` to include imports for `BibDatabaseContext` and `FilePreferences` classes as well as different arguments for `importPDFContent()` and `importDatabase()`, both related to the changes made to `PdfMergeMetadataImporter`
- Modified `PdfMergeMetadataImporter` to add an alternate `importDatabase()` function with different arguments, using the `BibDatabaseContext.getFileDirectories(FilePreferences)` function to get all file directories and relativize the file path using the `FileUtil.relativize(Path, List<Path>)` function, before passing it into the default override of the `importDatabase(Path)` function from the `Importer` class.
  `PdfMergeMetadataImporter` handles all of the actual logic relating to importing a PDF file.
  The new `importDatabase()` function has three arguments: `(Path file, BibDatabaseContext context, FilePreferences preferences)`
- Added a WIP test in `PdfMergeMetadataImporter`

Reviewers
subhramit
koppor ✓
Requested changes must be addressed to merge this pull request.

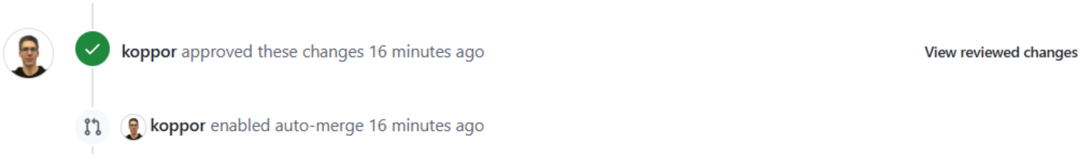
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

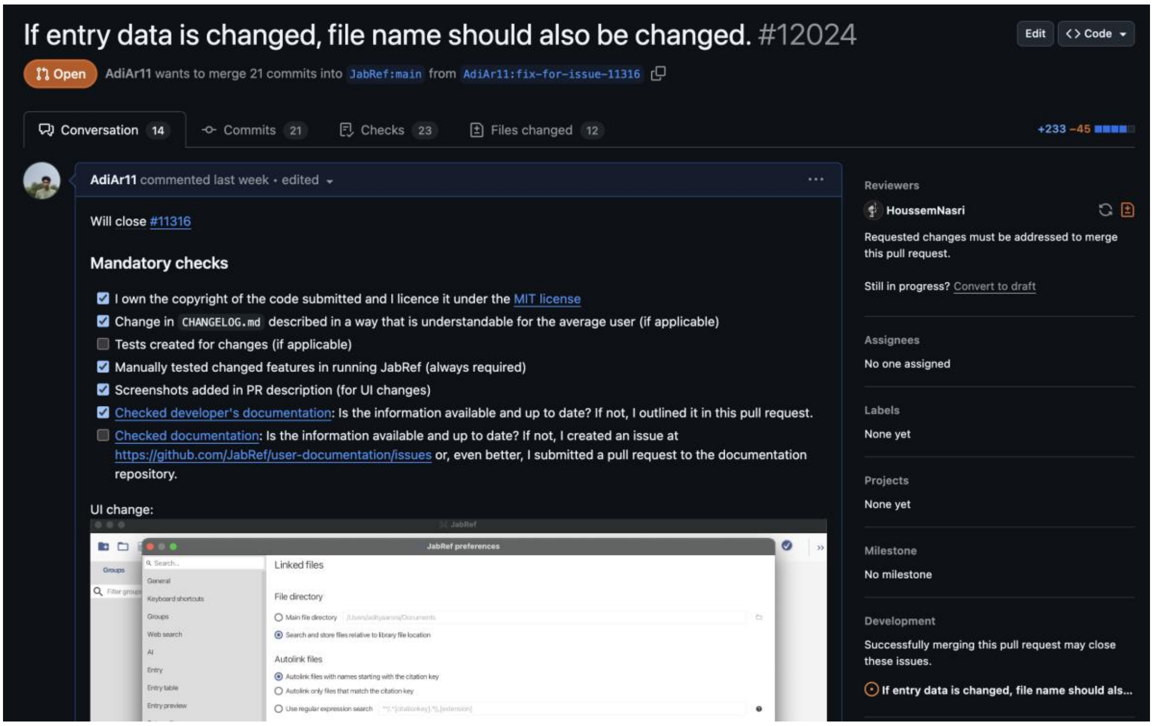


koppor approved these changes 16 minutes ago

koppor enabled auto-merge 16 minutes ago

Issues #11316:

https://github.com/JabRef/jabref/pull/12024



If entry data is changed, file name should also be changed. #12024

Open AdiAr11 wants to merge 21 commits into JabRef:main from AdiAr11:fix-for-issue-11316

AdiAr11 commented last week • edited

Will close #11316

Mandatory checks

- ☑ I own the copyright of the code submitted and I licence it under the MIT license
- ☑ Change in `CHANGELOG.md` described in a way that is understandable for the average user (if applicable)
- □ Tests created for changes (if applicable)
- ☑ Manually tested changed features in running JabRef (always required)
- ☑ Screenshots added in PR description (for UI changes)
- ☑ Checked developer's documentation: Is the information available and up to date? If not, I outlined it in this pull request.
- □ Checked documentation: Is the information available and up to date? If not, I created an issue at https://github.com/JabRef/user-documentation/issues or, even better, I submitted a pull request to the documentation repository.

UI change:

Linked files

File directory

○ Main file directory /Users/adityaarora/Documents

● Search and store files relative to library file location

Autolink files

● Autolink files with names starting with the citation key

○ Autolink only files that match the citation key

○ Use regular expression search **/.*[citationkey].*\\.[extension]

Reviewers

HoussemNasri

Requested changes must be addressed to merge this pull request.

Still in progress? Convert to draft

Assignees

No one assigned

Labels

None yet

Projects

None yet

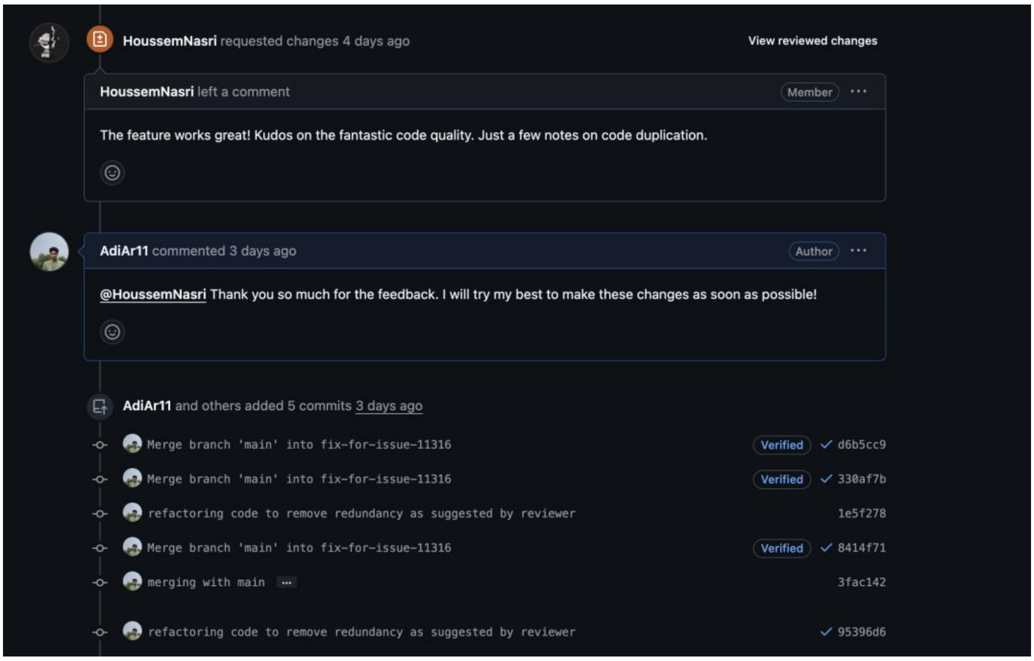
Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

If entry data is changed, file name should als...



HoussemNasri requested changes 4 days ago

HoussemNasri left a comment Member

The feature works great! Kudos on the fantastic code quality. Just a few notes on code duplication.

AdiAr11 commented 3 days ago Author

@HoussemNasri Thank you so much for the feedback. I will try my best to make these changes as soon as possible!

AdiAr11 and others added 5 commits 3 days ago

```
Merge branch 'main' into fix-for-issue-11316                    Verified ✓ d6b5cc9
Merge branch 'main' into fix-for-issue-11316                    Verified ✓ 330af7b
refactoring code to remove redundancy as suggested by reviewer            1e5f278
Merge branch 'main' into fix-for-issue-11316                    Verified ✓ 8414f71
merging with main ...                                                     3fac142

refactoring code to remove redundancy as suggested by reviewer          ✓ 95396d6
```

The PR is still open. I was asked to make some changes, which I did. Waiting to hear back from the reviewer. It should get merged soon once the reviewer reviews it.

Activity 4 - Documenting the Process

Documenting Issue #10557: Huizhe Ruan

Issue Analysis and Implementation

The issue "#10557: Store journal lists in separate directory" addresses the management of journal abbreviation lists in JabRef. Currently, users must manually select directories for each journal abbreviation file, making the process cumbersome. The issue proposes centralizing these lists in a single, configurable directory with automatic file detection and processing capabilities.

Personal Identification:

ANU Student ID: u7723366

GitHub Username: Daniel-Ruan

GitHub Profile: https://github.com/Daniel-Ruan

Project Guidelines and Standards:

- **Code Conventions:**

  Followed JabRef's MVC architecture pattern

  Maintained consistent code formatting and naming conventions

  Referenced existing implementations for similar functionality

- **Contribution Process:**

  Branch naming follows project convention ('fix-for-issue-10557')

  PR documentation includes clear description and implementation details

  Actively engaged with maintainers' feedback

- **Documentation:**

  Added inline code comments following project standards

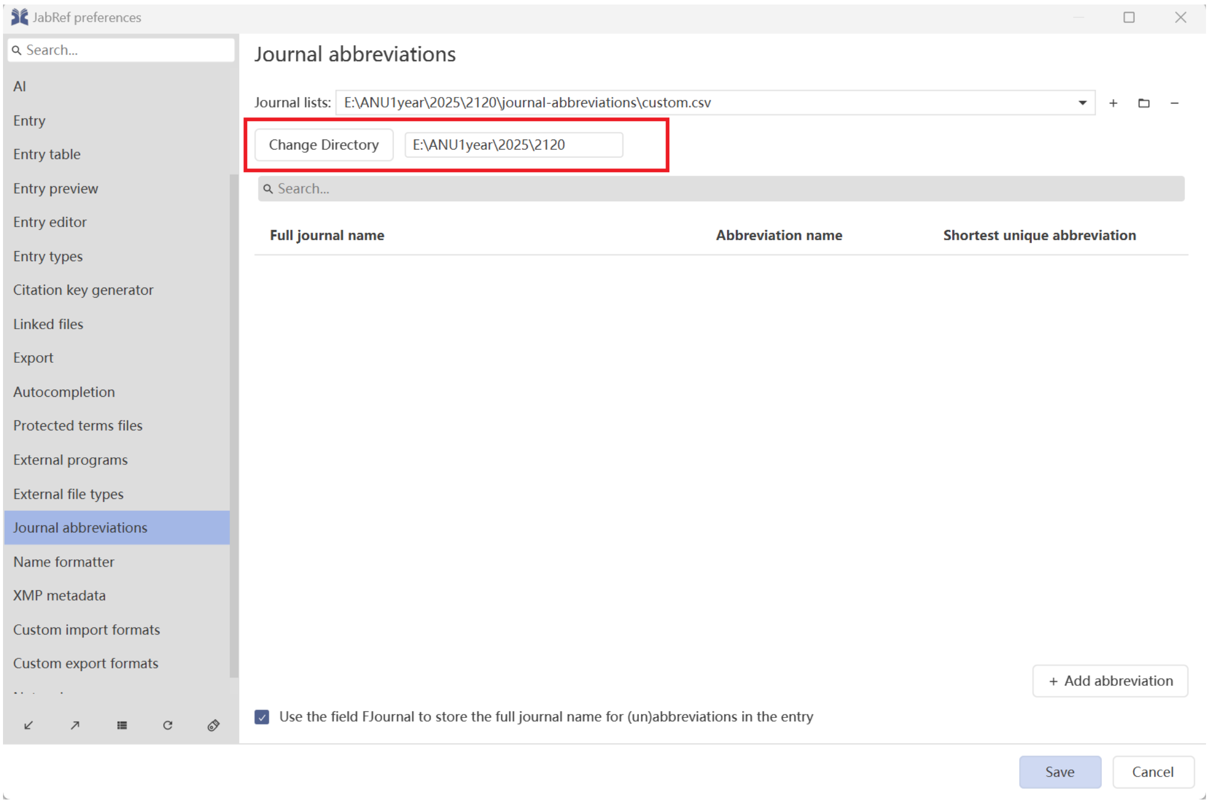
  Updated relevant documentation for new features

  Included screenshots demonstrating UI changes

Implementation Details:

- **User Interface Changes**

1. Added a directory selection button and text display in the preferences dialog
2. Implemented following MVC (Model-View-Controller) pattern to maintain code consistency
3. New UI elements allow users to view and modify the journal lists directory path



Journal abbreviations

Journal lists: E:\ANU1year\2025\2120\journal-abbreviations\custom.csv

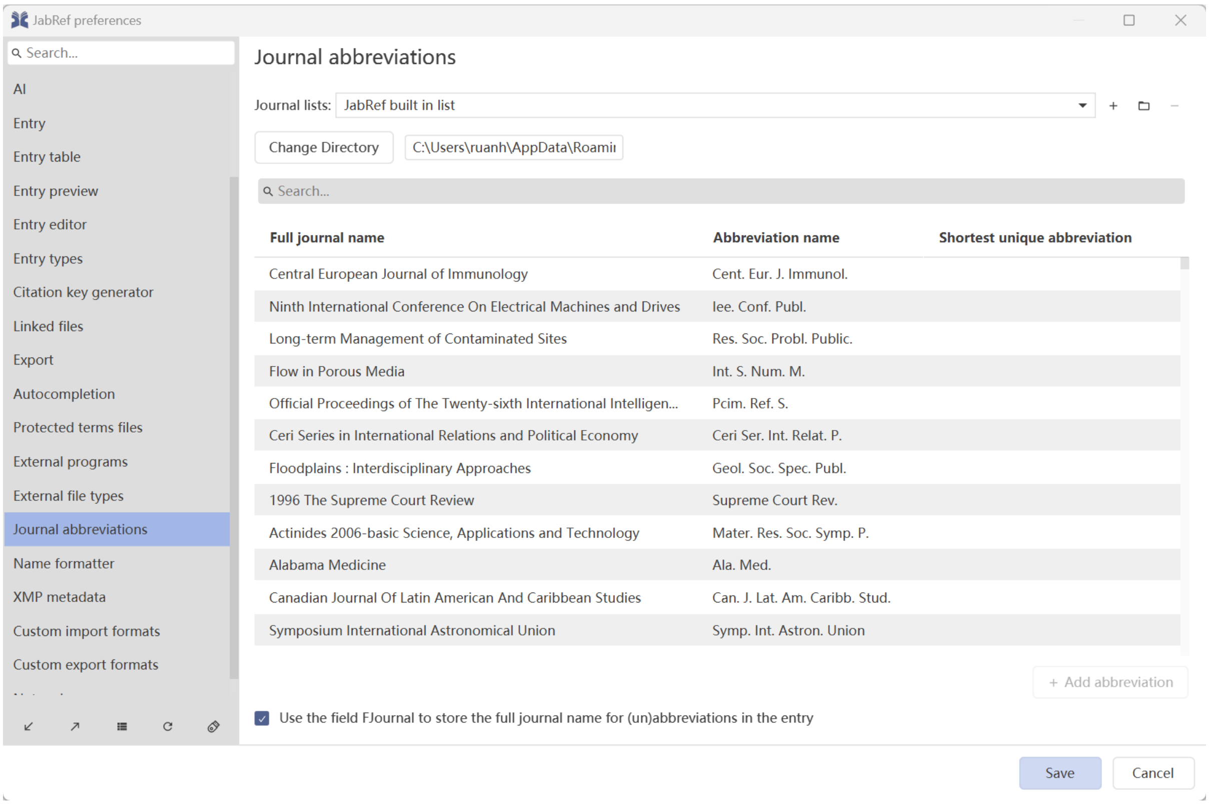
Change Directory E:\ANU1year\2025\2120

Full journal name	Abbreviation name	Shortest unique abbreviation

☑ Use the field FJournal to store the full journal name for (un)abbreviations in the entry

As shown in the screenshot, I have added content in the red area which includes:

1. A new UI component for directory selection and management:
2. Added a directory selection button
3. Added a text field displaying the current directory path
4. Implemented these changes following JabRef's established MVC pattern



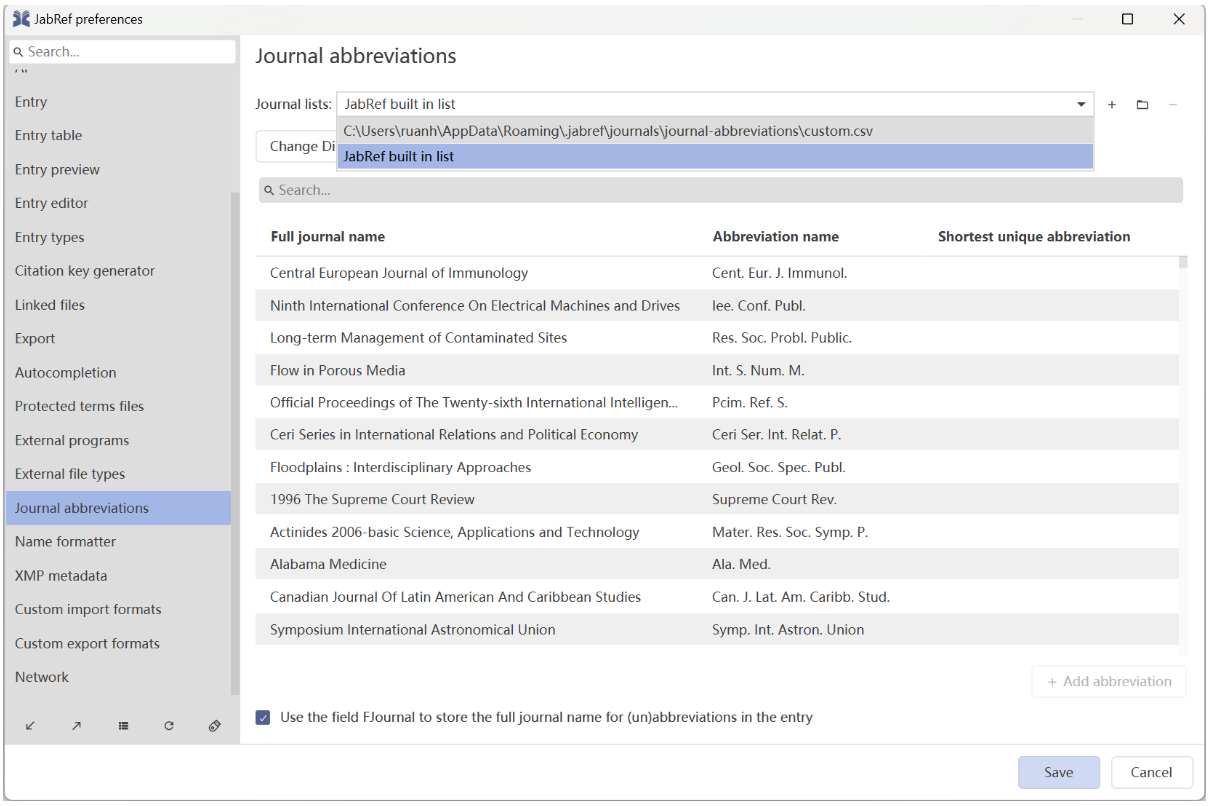
Journal abbreviations

Journal lists: JabRef built in list

C:\Users\ruanh\AppData\Roamii

Full journal name	Abbreviation name	Shortest unique abbreviation
Central European Journal of Immunology	Cent. Eur. J. Immunol.	
Ninth International Conference On Electrical Machines and Drives	Iee. Conf. Publ.	
Long-term Management of Contaminated Sites	Res. Soc. Probl. Public.	
Flow in Porous Media	Int. S. Num. M.	
Official Proceedings of The Twenty-sixth International Intelligen...	Pcim. Ref. S.	
Ceri Series in International Relations and Political Economy	Ceri Ser. Int. Relat. P.	
Floodplains : Interdisciplinary Approaches	Geol. Soc. Spec. Publ.	
1996 The Supreme Court Review	Supreme Court Rev.	
Actinides 2006-basic Science, Applications and Technology	Mater. Res. Soc. Symp. P.	
Alabama Medicine	Ala. Med.	
Canadian Journal Of Latin American And Caribbean Studies	Can. J. Lat. Am. Caribb. Stud.	
Symposium International Astronomical Union	Symp. Int. Astron. Union	

☑ Use the field FJournal to store the full journal name for (un)abbreviations in the entry



Journal abbreviations

Journal lists: JabRef built in list

C:\Users\ruanh\AppData\Roaming\jabref\journals\journal-abbreviations\custom.csv

JabRef built in list

Full journal name	Abbreviation name	Shortest unique abbreviation
Central European Journal of Immunology	Cent. Eur. J. Immunol.	
Ninth International Conference On Electrical Machines and Drives	Iee. Conf. Publ.	
Long-term Management of Contaminated Sites	Res. Soc. Probl. Public.	
Flow in Porous Media	Int. S. Num. M.	
Official Proceedings of The Twenty-sixth International Intelligen...	Pcim. Ref. S.	
Ceri Series in International Relations and Political Economy	Ceri Ser. Int. Relat. P.	
Floodplains : Interdisciplinary Approaches	Geol. Soc. Spec. Publ.	
1996 The Supreme Court Review	Supreme Court Rev.	
Actinides 2006-basic Science, Applications and Technology	Mater. Res. Soc. Symp. P.	
Alabama Medicine	Ala. Med.	
Canadian Journal Of Latin American And Caribbean Studies	Can. J. Lat. Am. Caribb. Stud.	
Symposium International Astronomical Union	Symp. Int. Astron. Union	

☑ Use the field FJournal to store the full journal name for (un)abbreviations in the entry

New functionality for journal abbreviation management:

1. Users can now select a custom directory for storing journal lists

2. The application automatically creates a custom.csv file if none exists

3. The selected directory path is saved in user preferences
4. All journal abbreviation lists from this directory are automatically loaded

- **Logic Implementation**

1. Created getDefaultAbbreviationDir() method to manage default directory paths
2. Implemented automatic CSV file creation functionality
3. Added logic to detect and process journal lists in the specified directory

Codebase Integration:

- **Key files modified include**:

1. JournalAbbreviationPreferences.java: Added directory management functionality
2. UI components for directory selection and display
3. Logic layer modifications for file handling and preference storage

Development Process:

1. Forked and cloned the JabRef repository
2. Created branch 'fix-for-issue-10557' following project conventions
3. Implemented UI changes following MVC pattern
4. Added backend logic for directory management
5. Submitted PR with detailed documentation

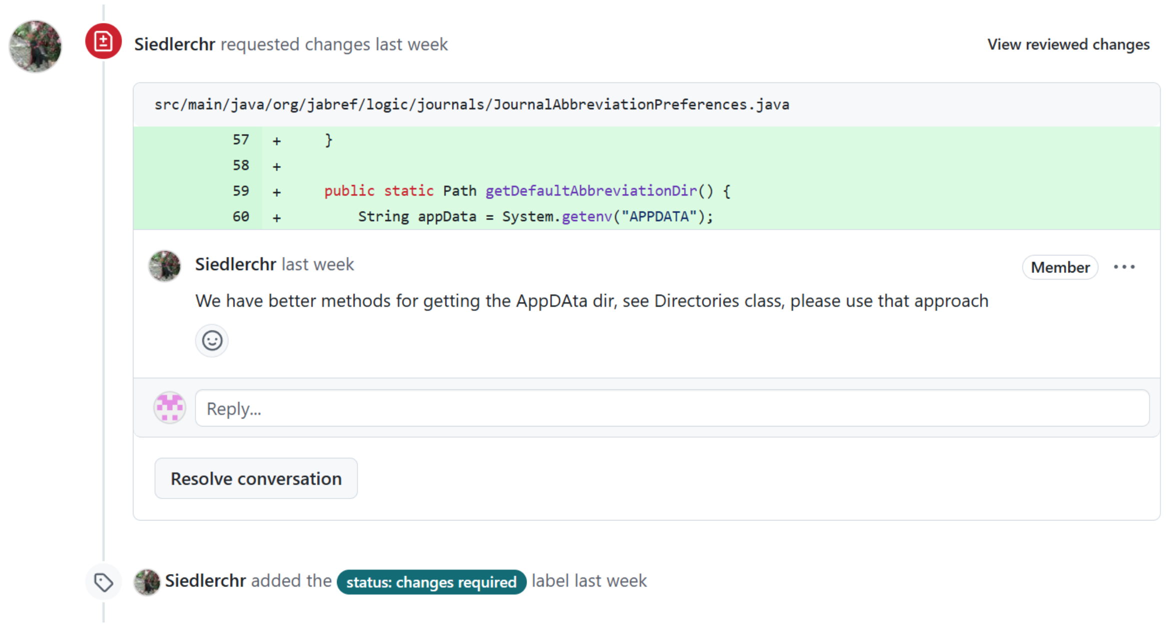
Areas for Improvement:

Current unit tests failures related to getDefaultAbbreviationDir() implementation

Code URL:

https://github.com/Daniel-Ruan/jabref/blob/fix-for-issue-10557/src/main/java/org/jabref/logic/journals/JournalAbbreviationPreferences.java#L59-L65

Need to utilize existing directory code as suggested by maintainers



Siedlerchr requested changes last week

src/main/java/org/jabref/logic/journals/JournalAbbreviationPreferences.java

```
+     }
+
+     public static Path getDefaultAbbreviationDir() {
+         String appData = System.getenv("APPDATA");
```

Siedlerchr last week

We have better methods for getting the AppDAta dir, see Directories class, please use that approach

Siedlerchr added the status: changes required label last week

As noted by the JabRef maintainers, they suggested using the existing directory code for creating the default directory. I am currently studying and researching this approach to better understand the existing codebase implementation. While my current implementation creates a functional solution, it doesn't fully align with the project's established patterns for directory management.

I remain hopeful and enthusiastic about eventually resolving these issues and successfully merging my contribution into the JabRef project.

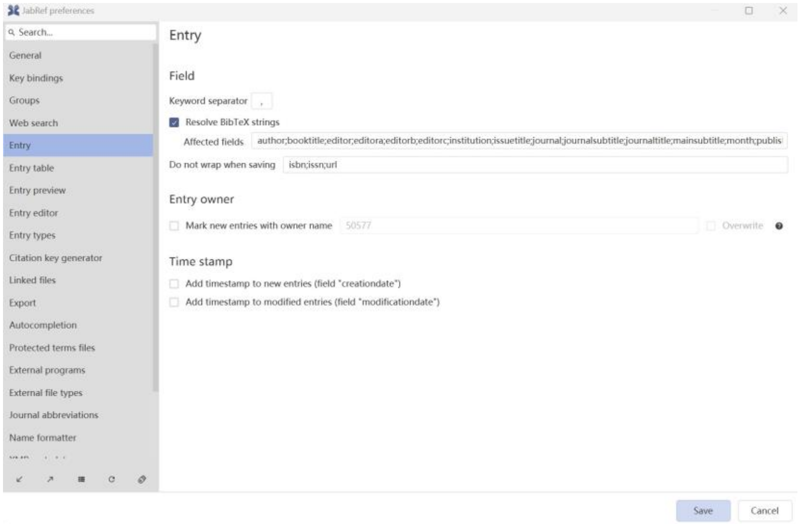
Documenting Issue #468: Jing Lu

What the issue is:

The issue I worked on, Issue #468 involved updating the user documentation for JabRef's "Keyword Separator" setting. The issue at hand involved outdated documentation for the "keyword separator" setting. It required updating both textual descriptions and screenshots. The need for this update was initially identified through Issue #177, which introduced changes

to the keyword separator feature, including limiting it to a single character and relocating the setting from Preferences -> Group to Preferences -> Entry.

I studied the existing documentation files (groups.md and keywords.md) and navigated through the software to verify the correct settings location. By understanding how these files are structured and how they fit within the overall documentation, I was able to plan the necessary changes.



Entry

Field

Keyword separator ,

☑ Resolve BibTeX strings

Affected fields author;booktitle;editor;editora;editorb;editorc;institution;issuetitle;journal;journalsubtitle;journaltitle;mainsubtitle;month;publis

Do not wrap when saving isbn;issn;url

Entry owner

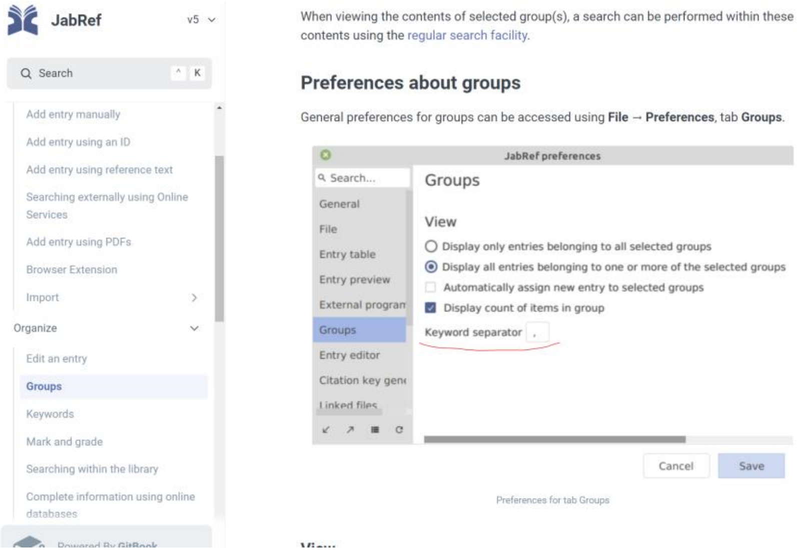
□ Mark new entries with owner name 50577 □ Overwrite

Time stamp

□ Add timestamp to new entries (field "creationdate")

□ Add timestamp to modified entries (field "modificationdate")

Also, the screenshots for the page under the groups section were incorrect.



When viewing the contents of selected group(s), a search can be performed within these contents using the regular search facility.

Preferences about groups

General preferences for groups can be accessed using **File → Preferences**, tab **Groups**.

Groups

View

○ Display only entries belonging to all selected groups

● Display all entries belonging to one or more of the selected groups

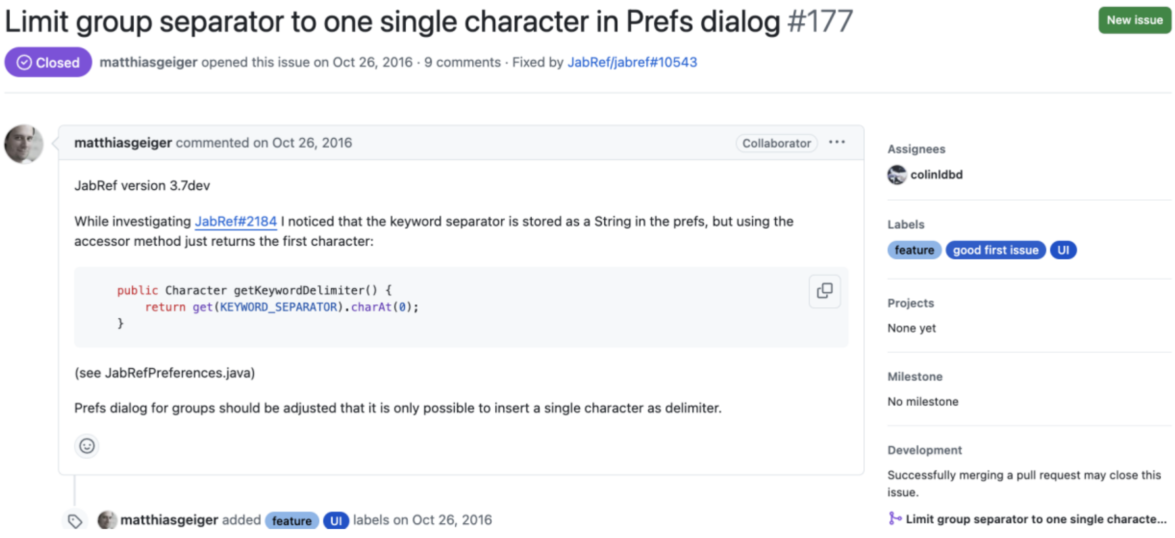
□ Automatically assign new entry to selected groups

☑ Display count of items in group

Keyword separator ,

Preferences for tab Groups

The issue was triggered by an inconsistency between the current software interface and the existing documentation. This issue was discovered because of the resolution of #issue177: https://github.com/koppor/jabref/issues/177.



Limit group separator to one single character in Prefs dialog #177

matthiasgeiger opened this issue on Oct 26, 2016 · 9 comments · Fixed by JabRef/jabref#10543

matthiasgeiger commented on Oct 26, 2016

JabRef version 3.7dev

While investigating JabRef#2184 I noticed that the keyword separator is stored as a String in the prefs, but using the accessor method just returns the first character:

```
    public Character getKeywordDelimiter() {
        return get(KEYWORD_SEPARATOR).charAt(0);
    }
```

(see JabRefPreferences.java)

Prefs dialog for groups should be adjusted that it is only possible to insert a single character as delimiter.

matthiasgeiger added feature UI labels on Oct 26, 2016

Assignees

colinldbd

Labels

feature good first issue UI

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Limit group separator to one single characte...

My original plan to resolve the issue, and the specifics of my implementation:

To address this issue, I planned to:

- Run the Jabref project, open preferences, view the current keyword location, and check the existing information for groups.
- Update the Text Descriptions: Update the relevant text in the documentation to accurately reflect the current location of the "keyword separator" setting.
- Update Screenshots: Replace outdated screenshots with new ones showing the current UI to avoid further confusion.

The specifics of my implementation included editing two files: groups.md and keywords.md, where both the descriptions and screenshots needed updates.

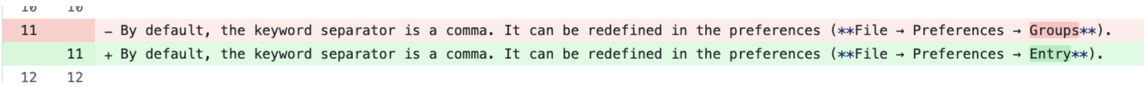
What relevant parts of the codebase are involved in the issue?

- Repository URL: JabRef User Documentation
- Issue URL: Issue #468
- Related Issue: Issue #177

The files involved were:

- groups.md: This file contained outdated screenshots related to the keyword separator setting.
- keywords.md: This file had the incorrect textual description of where the keyword

separator setting was located.



```
10  10
11      - By default, the keyword separator is a comma. It can be redefined in the preferences (**File → Preferences → Groups**).
    11  + By default, the keyword separator is a comma. It can be redefined in the preferences (**File → Preferences → Entry**).
12  12
```

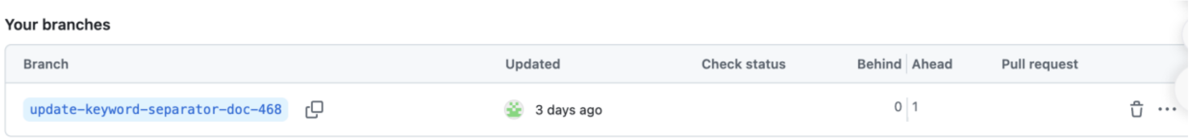
- En/.gitbook/assets: The screenshot needs to be replaced.

These files were part of the JabRef user documentation, which aims to assist users in configuring and using the software effectively.

What was involved in opening and submitting the PR?

The process of opening and submitting the Pull Request (PR) involved:

- Creating a branch: First, I created a new branch called update-keyword-separator-doc-#468 in my forked repository to implement the changes.



Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
update-keyword-separator-doc-468	3 days ago		0	1	

- Making the changes: After editing the file and testing it for accuracy, I staged the changes, committed them, and pushed them to the forked repository.

- Checking with others: After making the changes, I checked my changes with the group members in the week12 tutorial and asked the tutor for feedback after the presentation.

- PR submission: I then opened a PR to merge these changes into the main repository. The PR was titled "Update keyword separator documentation (#468)" and contained a detailed description of the changes made, why the changes were necessary, and how the changes would benefit users.

Can you explain all the relevant parts within the repository that relate to, rely on, are impacted by, or anything else by the issue you are working on? And What would solving this mean to the project?

Fixing this issue meant:

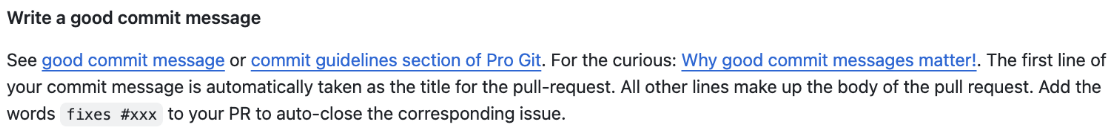
- Improved User Experience: Users would be able to follow the instructions accurately without confusion.

- Up-to-date Documentation: The documentation now accurately reflects the current state of the software, ensuring users have reliable and helpful resources.

Does your solution fit neatly into the rest of the project? Does it follow the conventions of the project?

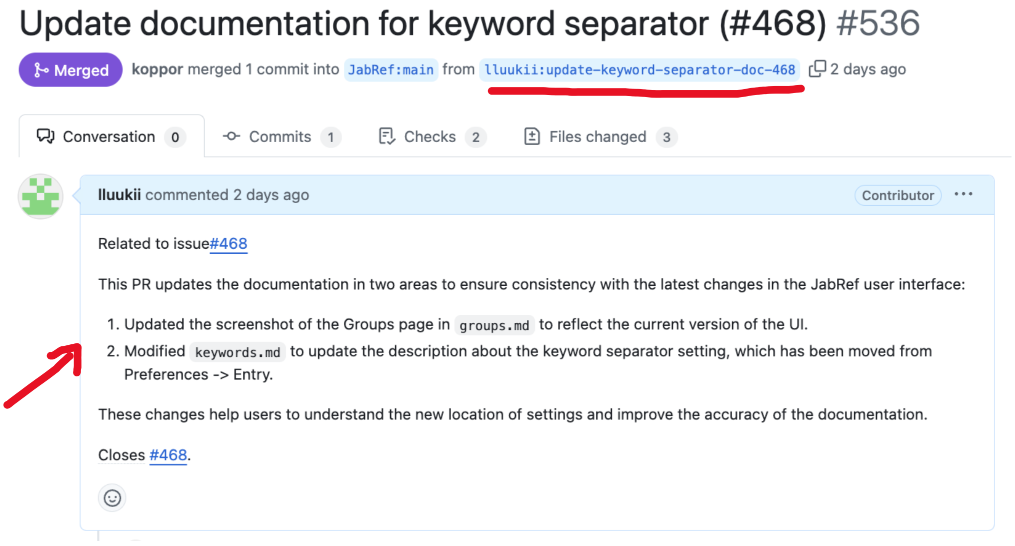
My solution followed all project conventions:

- Consistent Formatting: The updated documentation maintained the same formatting and structure as other parts of the user manual.
- Branch Naming and Commit Messages: I used descriptive branch names and commit messages that adhered to the project's contributing guidelines.



Write a good commit message

See good commit message or commit guidelines section of Pro Git. For the curious: Why good commit messages matter!. The first line of your commit message is automatically taken as the title for the pull-request. All other lines make up the body of the pull request. Add the words `fixes #xxx` to your PR to auto-close the corresponding issue.



Update documentation for keyword separator (#468) #536

Merged koppor merged 1 commit into `JabRef:main` from `lluukii:update-keyword-separator-doc-468` 2 days ago

lluukii commented 2 days ago

Related to issue#468

This PR updates the documentation in two areas to ensure consistency with the latest changes in the JabRef user interface:

1. Updated the screenshot of the Groups page in `groups.md` to reflect the current version of the UI.
2. Modified `keywords.md` to update the description about the keyword separator setting, which has been moved from Preferences -> Entry.

These changes help users to understand the new location of settings and improve the accuracy of the documentation.

Closes #468.

(https://github.com/JabRef/user-documentation/pull/536#issue-2610684039)

Is it easy to understand what you have done/propose to do?

The changes I made were easy to understand:

- The PR provided a detailed explanation of what was changed, along with updated screenshots for visual reference.
- The edits were straightforward: updating screenshots and a few textual descriptions.

Are there edge cases / performance problems / other things that you (or they - the individual who opened the issue) have not considered/addressed?

There were a few things I considered:

- Accuracy of UI Descriptions: I verified that the new screenshots matched the current UI to avoid misleading users.
- Potential Updates: I also considered that the UI could change again in the future, so I made the descriptions general enough to accommodate minor adjustments without becoming immediately outdated.

Was there anything you could have improved on?

One area I could improve on would be to automate the process of validating the documentation against software updates. This would reduce the risk of the documentation becoming outdated after future software changes. Implementing a script to verify that settings descriptions match the actual UI could help maintain documentation accuracy over time.

Another area for improvement was the review process. In the process of solving this issue, the hastiest part was the review. I did not consider requesting the review of my teammates through the function on GitHub, but instead submitted the PR immediately after seeking their consent offline. Using GitHub's built-in review request feature would have formalized the process, ensured a documented review trail and made it easier for maintainers and other contributors to see who reviewed the changes and what feedback was provided.

Contributions to the issue and GitHub usernames:

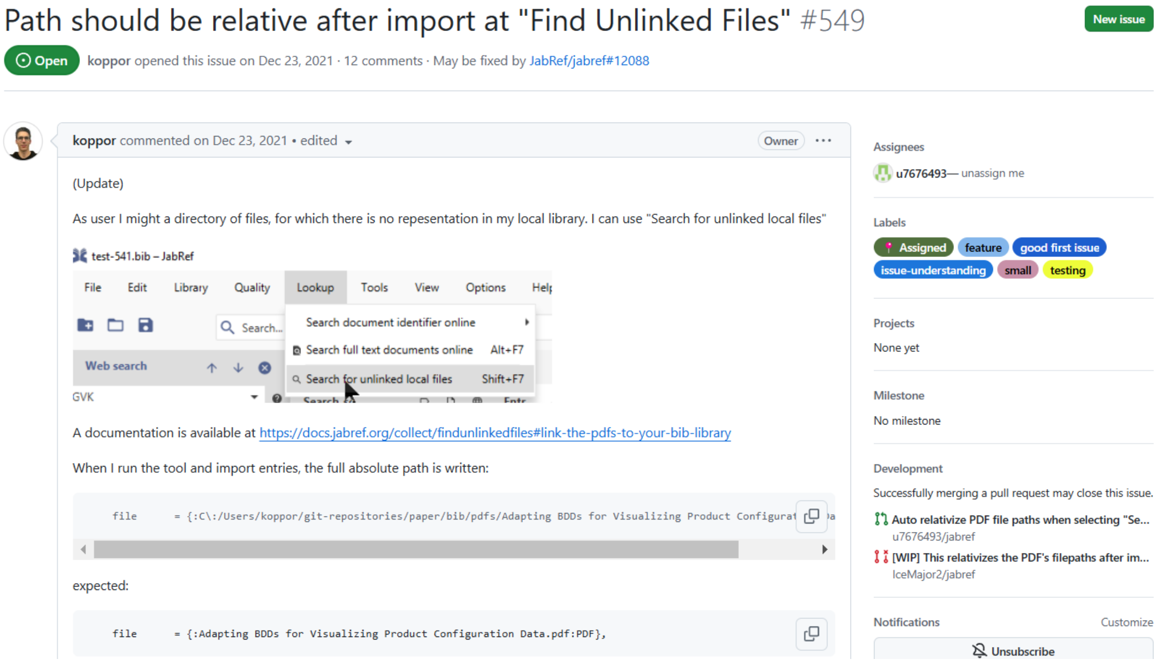
- **My Contributions**: I implemented the changes by updating both the text and images to match the current JabRef interface. I also documented the process thoroughly and ensured that the PR followed the project's conventions.
- **Collaborator Contributions**: I worked with koppor(@koppor-GitHub), who reviewed my proposed changes.
- **GitHub usernames:** lluukii(https://github.com/lluukii) and Jing Lu

(https://github.com/lluuki). Both of them are mine.

Documenting Issue #549: Eric Liehne

What is the issue?

https://github.com/koppor/jabref/issues/549



Path should be relative after import at "Find Unlinked Files" #549

koppor opened this issue on Dec 23, 2021 · 12 comments · May be fixed by JabRef/jabref#12088

koppor commented on Dec 23, 2021 • edited

(Update)

As user I might a directory of files, for which there is no repesentation in my local library. I can use "Search for unlinked local files"

test-541.bib – JabRef
File
Edit
Library
Quality
Lookup
Tools
View
Options
Help
Search...
Search document identifier online
Search full text documents online Alt+F7
Search for unlinked local files Shift+F7
Web search
GVK

A documentation is available at https://docs.jabref.org/collect/findunlinkedfiles#link-the-pdfs-to-your-bib-library

When I run the tool and import entries, the full absolute path is written:

```
    file     = {:C\:/Users/koppor/git-repositories/paper/bib/pdfs/Adapting BDDs for Visualizing Product Configurat
```

expected:

```
    file     = {:Adapting BDDs for Visualizing Product Configuration Data.pdf:PDF},
```

Assignees

u7676493— unassign me

Labels

Assigned feature good first issue issue-understanding small testing

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Auto relativize PDF file paths when selecting "Se...
u7676493/jabref

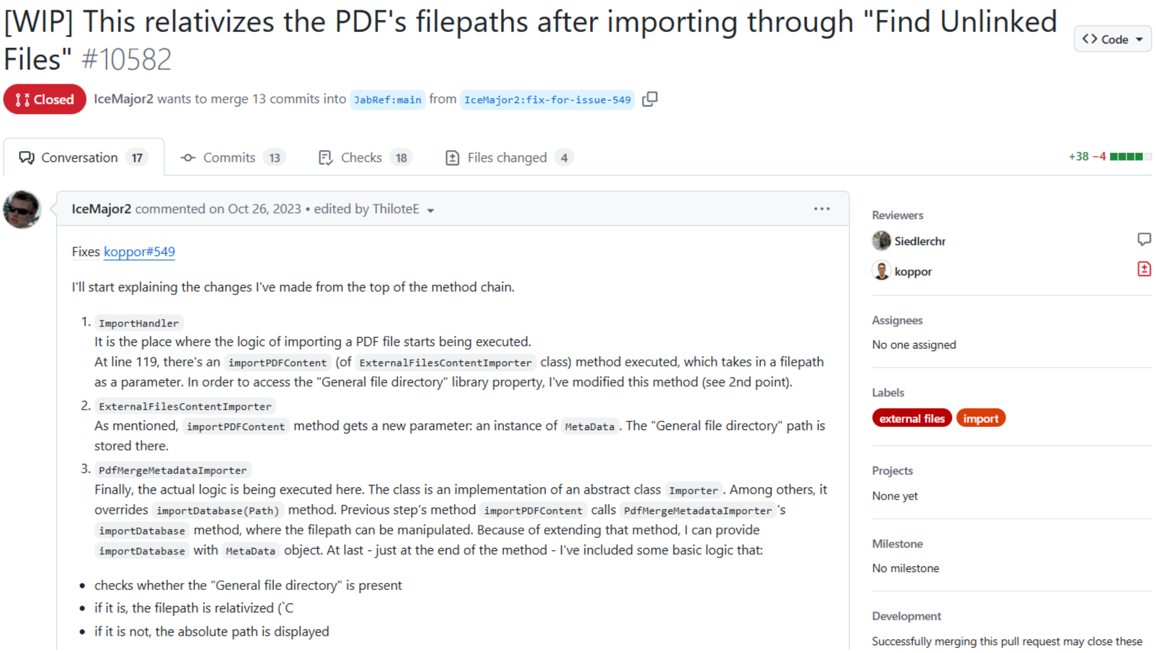
[WIP] This relativizes the PDF's filepaths after im...
IceMajor2/jabref

This issue involves importing files using the “Search for unlinked local files” option as outlined in the following documentation: https://docs.jabref.org/collect/findunlinkedfiles#link-the-pdfs-to-your-bib-library - attempting to import a file will not relativise the file path.

This happens because the file path is not changed by any of the functions that are used to import the file.

How I planned to solve the issue

I planned to mostly follow the previous PR (https://github.com/JabRef/jabref/pull/10582) that attempted to solve this issue (created by IceMajor2), whilst also looking at the feedback and reviews for that PR to see how I could improve on it.



[WIP] This relativizes the PDF's filepaths after importing through "Find Unlinked Files" #10582

IceMajor2 wants to merge 13 commits into `JabRef:main` from `IceMajor2:fix-for-issue-549`

IceMajor2 commented on Oct 26, 2023 • edited by ThiloteE

Fixes koppor#549

I'll start explaining the changes I've made from the top of the method chain.

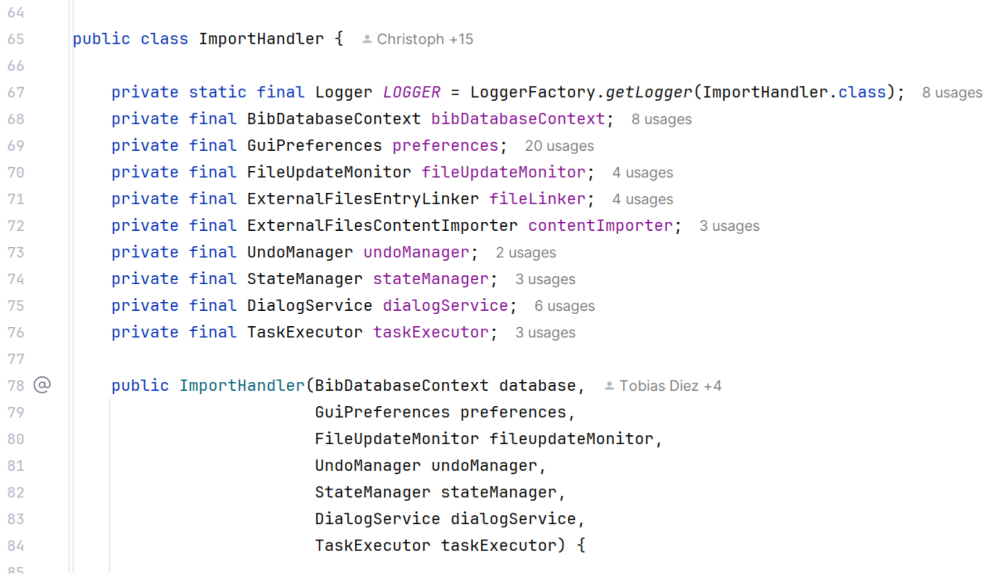
1. `ImportHandler`
   It is the place where the logic of importing a PDF file starts being executed.
   At line 119, there's an `importPDFContent` (of `ExternalFilesContentImporter` class) method executed, which takes in a filepath as a parameter. In order to access the "General file directory" library property, I've modified this method (see 2nd point).
2. `ExternalFilesContentImporter`
   As mentioned, `importPDFContent` method gets a new parameter: an instance of `MetaData`. The "General file directory" path is stored there.
3. `PdfMergeMetadataImporter`
   Finally, the actual logic is being executed here. The class is an implementation of an abstract class `Importer`. Among others, it overrides `importDatabase(Path)` method. Previous step's method `importPDFContent` calls `PdfMergeMetadataImporter`'s `importDatabase` method, where the filepath can be manipulated. Because of extending that method, I can provide `importDatabase` with `MetaData` object. At last - just at the end of the method - I've included some basic logic that:

- checks whether the "General file directory" is present
- if it is, the filepath is relativized (`C
- if it is not, the absolute path is displayed

Relevant parts of the codebase

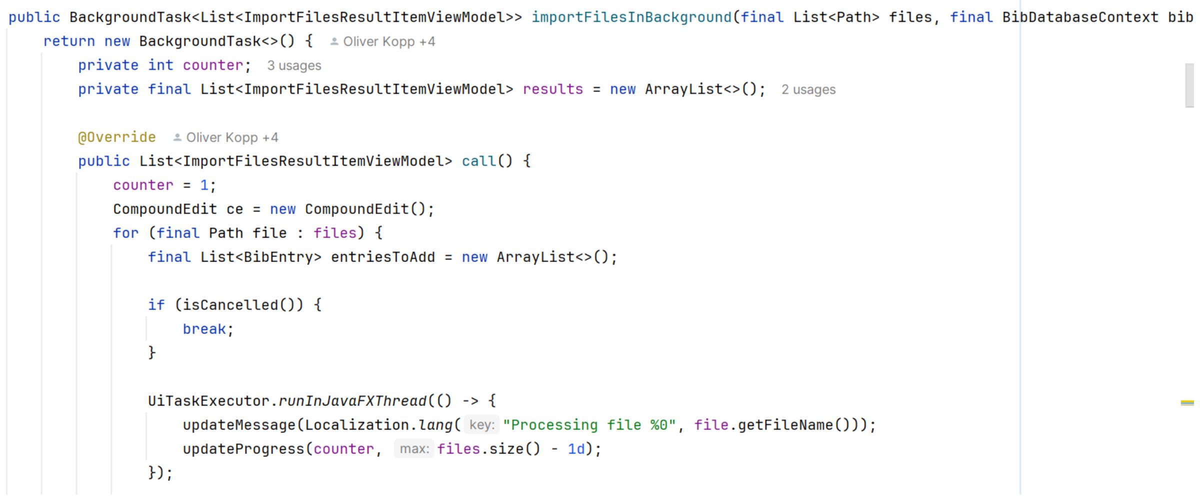
To improve my understanding of the codebase, I read through the changes in the previous PR for this issue, as well as comments made on and reviews of these changes.

Through this, I was able to determine that the `PdfMergeMetadataImporter`, `ExternalFilesContentImporter`, and `ImportHandler` classes would have to be modified to start the solution to this issue. I was able to determine this thanks to reading the previous PR.



```
public class ImportHandler {  Christoph +15

    private static final Logger LOGGER = LoggerFactory.getLogger(ImportHandler.class);  8 usages
    private final BibDatabaseContext bibDatabaseContext;  8 usages
    private final GuiPreferences preferences;  20 usages
    private final FileUpdateMonitor fileUpdateMonitor;  4 usages
    private final ExternalFilesEntryLinker fileLinker;  4 usages
    private final ExternalFilesContentImporter contentImporter;  3 usages
    private final UndoManager undoManager;  2 usages
    private final StateManager stateManager;  3 usages
    private final DialogService dialogService;  6 usages
    private final TaskExecutor taskExecutor;  3 usages

    public ImportHandler(BibDatabaseContext database,  Tobias Diez +4
                         GuiPreferences preferences,
                         FileUpdateMonitor fileupdateMonitor,
                         UndoManager undoManager,
                         StateManager stateManager,
                         DialogService dialogService,
                         TaskExecutor taskExecutor) {
```

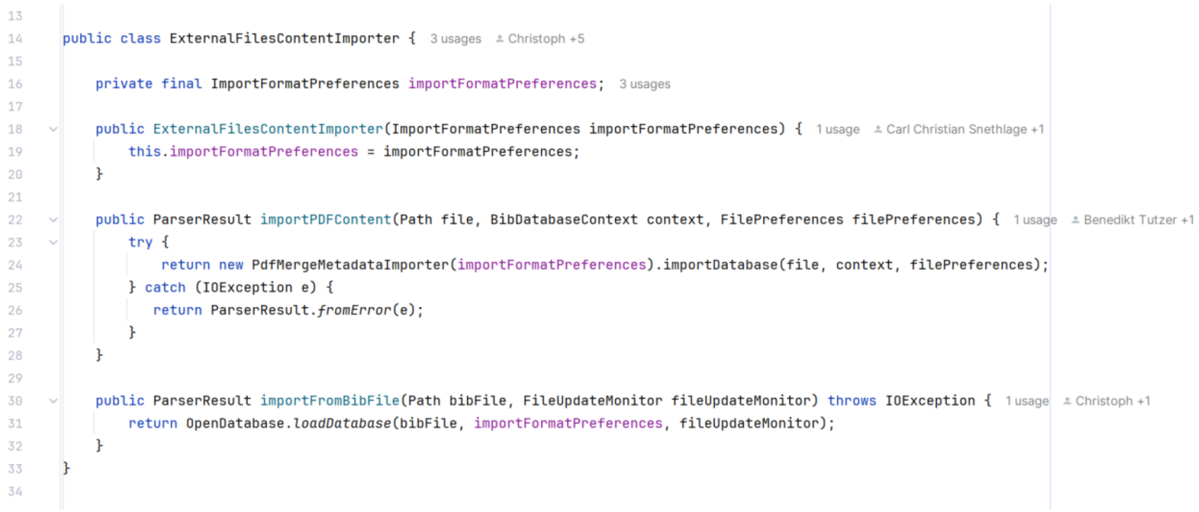
The start of the `ImportHandler` class as shown in IntelliJ, including the parameters for the constructor.



```
public BackgroundTask<List<ImportFilesResultItemViewModel>> importFilesInBackground(final List<Path> files, final BibDatabaseContext bib
    return new BackgroundTask<>() {  Oliver Kopp +4
        private int counter;  3 usages
        private final List<ImportFilesResultItemViewModel> results = new ArrayList<>();  2 usages

        @Override  Oliver Kopp +4
        public List<ImportFilesResultItemViewModel> call() {
            counter = 1;
            CompoundEdit ce = new CompoundEdit();
            for (final Path file : files) {
                final List<BibEntry> entriesToAdd = new ArrayList<>();

                if (isCancelled()) {
                    break;
                }

                UiTaskExecutor.runInJavaFXThread(() -> {
                    updateMessage(Localization.lang( key: "Processing file %0", file.getFileName()));
                    updateProgress(counter,  max: files.size() - 1d);
                });
```

The start of the `importFilesInBackground` function as shown in IntelliJ.

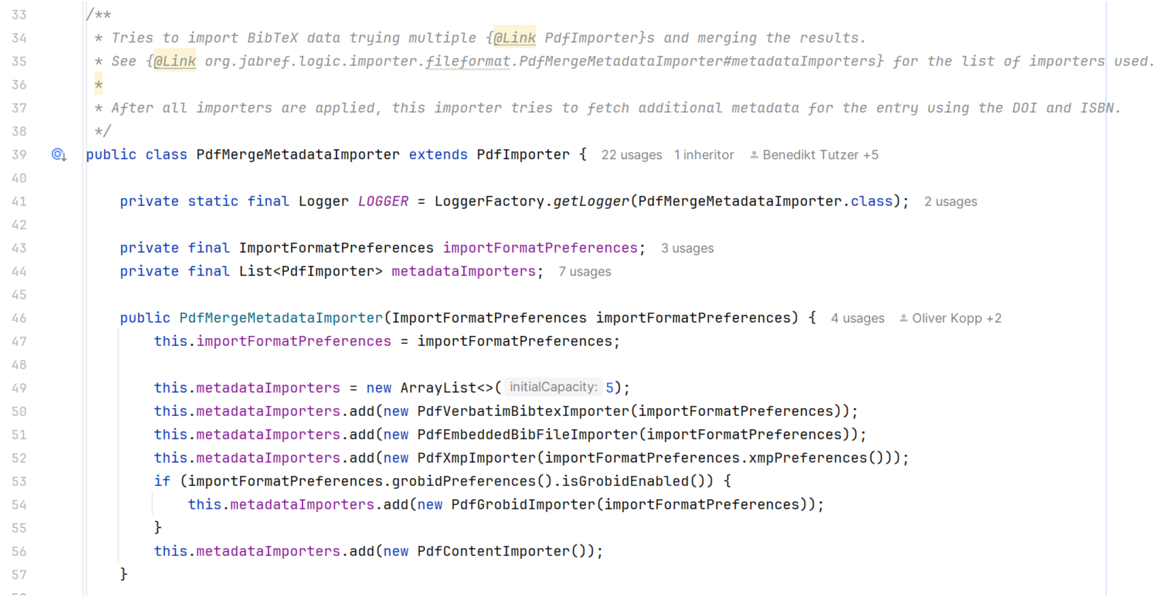
The `ImportHandler` class acts as the initial point where the logic used for importing data is run, serving mostly to call functions from instances of the `Importer` class – or subclasses of it. The function that this issue focuses on is `importFilesInBackground`, which runs the `importPDFContent` function from the `ExternalFilesContentImporter` class.



```

public class ExternalFilesContentImporter {  3 usages  Christoph +5

    private final ImportFormatPreferences importFormatPreferences;  3 usages

    public ExternalFilesContentImporter(ImportFormatPreferences importFormatPreferences) {  1 usage  Carl Christian Snethlage +1
        this.importFormatPreferences = importFormatPreferences;
    }

    public ParserResult importPDFContent(Path file, BibDatabaseContext context, FilePreferences filePreferences) {  1 usage  Benedikt Tutzer +1
        try {
            return new PdfMergeMetadataImporter(importFormatPreferences).importDatabase(file, context, filePreferences);
        } catch (IOException e) {
            return ParserResult.fromError(e);
        }
    }

    public ParserResult importFromBibFile(Path bibFile, FileUpdateMonitor fileUpdateMonitor) throws IOException {  1 usage  Christoph +1
        return OpenDatabase.loadDatabase(bibFile, importFormatPreferences, fileUpdateMonitor);
    }
}

```

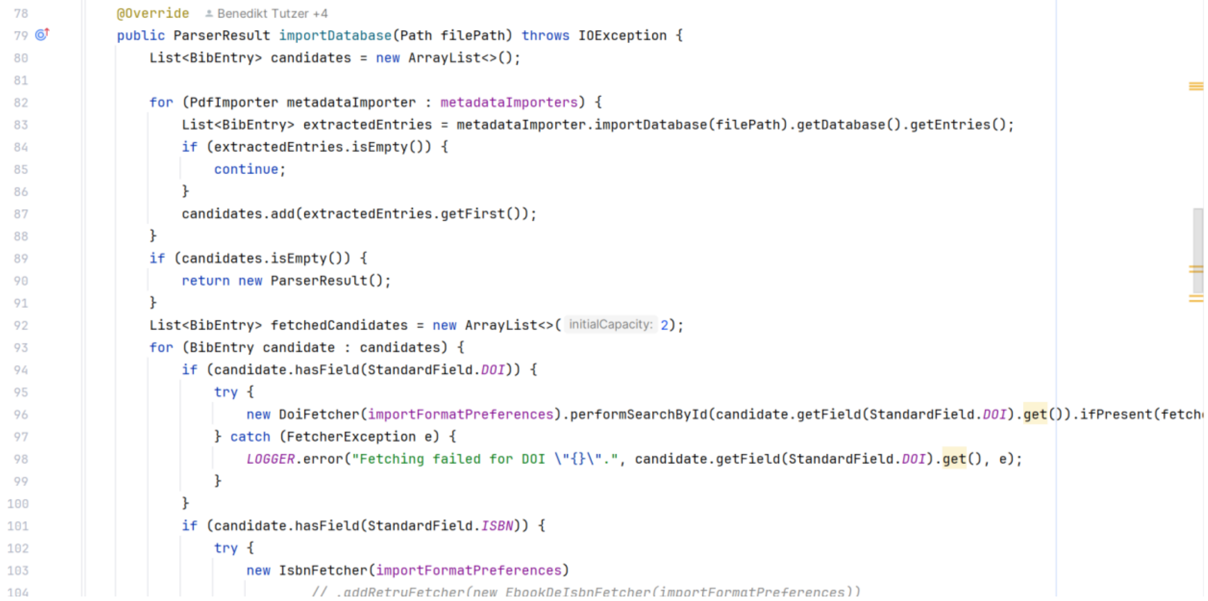
The entirety of the `ExternalFilesContentImporter` class, excluding imports and including the updated `importPDFContent` function.

The `ExternalFilesContentImporter` class is small in comparison with the other two involved classes and in this context is used solely to run the importDatabase function from the `PdfMergeMetadataImporter` class (a subclass of `Importer`) via the `importPDFContent` function.



```java
/**
 * Tries to import BibTeX data trying multiple {@link PdfImporter}s and merging the results.
 * See {@link org.jabref.logic.importer.fileformat.PdfMergeMetadataImporter#metadataImporters} for the list of importers used.
 *
 * After all importers are applied, this importer tries to fetch additional metadata for the entry using the DOI and ISBN.
 */
public class PdfMergeMetadataImporter extends PdfImporter {

    private static final Logger LOGGER = LoggerFactory.getLogger(PdfMergeMetadataImporter.class);

    private final ImportFormatPreferences importFormatPreferences;
    private final List<PdfImporter> metadataImporters;

    public PdfMergeMetadataImporter(ImportFormatPreferences importFormatPreferences) {
        this.importFormatPreferences = importFormatPreferences;

        this.metadataImporters = new ArrayList<>( initialCapacity: 5);
        this.metadataImporters.add(new PdfVerbatimBibtexImporter(importFormatPreferences));
        this.metadataImporters.add(new PdfEmbeddedBibFileImporter(importFormatPreferences));
        this.metadataImporters.add(new PdfXmpImporter(importFormatPreferences.xmpPreferences()));
        if (importFormatPreferences.grobidPreferences().isGrobidEnabled()) {
            this.metadataImporters.add(new PdfGrobidImporter(importFormatPreferences));
        }
        this.metadataImporters.add(new PdfContentImporter());
    }
```

The start of the PdfMergeMetadataImporter class, including the constructor and class description.



```java
    @Override
    public ParserResult importDatabase(Path filePath) throws IOException {
        List<BibEntry> candidates = new ArrayList<>();

        for (PdfImporter metadataImporter : metadataImporters) {
            List<BibEntry> extractedEntries = metadataImporter.importDatabase(filePath).getDatabase().getEntries();
            if (extractedEntries.isEmpty()) {
                continue;
            }
            candidates.add(extractedEntries.getFirst());
        }
        if (candidates.isEmpty()) {
            return new ParserResult();
        }
        List<BibEntry> fetchedCandidates = new ArrayList<>( initialCapacity: 2);
        for (BibEntry candidate : candidates) {
            if (candidate.hasField(StandardField.DOI)) {
                try {
                    new DoiFetcher(importFormatPreferences).performSearchById(candidate.getField(StandardField.DOI).get()).ifPresent(fetch
                } catch (FetcherException e) {
                    LOGGER.error("Fetching failed for DOI \"{}\".", candidate.getField(StandardField.DOI).get(), e);
                }
            }
            if (candidate.hasField(StandardField.ISBN)) {
                try {
                    new IsbnFetcher(importFormatPreferences)
                            // .addRetryFetcher(new EbookDeIsbnFetcher(importFormatPreferences))
```

The start of the importDatabase(Path) function – the override of the equivalent function in the base Importer class.

The PdfMergeMetadataImporter class is where most of the logic is run. In the importDatabase function, it takes a Path variable (filePath) as its argument and parses the database in the location specified by filePath.

The issue I was assigned involves this function – the code does not relativize the filePath variable. This issue affects the "Search for unlinked local files" function, handled by the FindUnlinkedFilesAction class.

How I solved the issue

I mostly followed the plan I had made, updating all the required functions and considering the reviews and feedback from the previous PR.

In `ExternalFileContentImporter`, I updated `importPDFContent` to include parameters for the .bib file and preferences for linked files and created a separate `importDatabase` function in `PdfMergeMetadataImporter`, so that I could use the `getFileDirectories` function for relativising the file paths. The arguments used for the `importPDFContent` call in `ImportHandler` were changed accordingly.

The separate `importDatabase` function was created because the main `importDatabase` function serves to override the function of the same name from the superclass `Importer`. By keeping the functionality of the overriding function identical, I can avoid modifying the arguments for all uses of that function (when it is being used from `PdfMergeMetadataImporter`) and can instead just focus on changing the arguments for the most important instances of the function for the issue.

This version of the `importDatabase` function uses the same parameters as `importPDFContent(Path filePath, BibDatabaseContext context, FilePreferences filePreferences)` to allow access to the `getFileDirectories` function (which requires both `BibDatabaseContext` (as the class where the function is defined) and `FilePreferences` (as the function's sole parameter)). This is needed because the directories form the required parameter for `FileUtil.relativize` – the key function that relativises `filePath` and forms the solution for this issue.

My attempt at automated testing of the solution was not successful. I can confirm via both comments made by koppor and my own manual testing that my solution works, but the automated test I wrote exemplifies a significant problem I ended up experiencing while making this PR – I was mostly reliant on the previous PR when it came to understanding the codebase, even though the solution I had written contained a number of significant differences in how it was handled. This meant that I ended up having to use guesswork to create an automatic test for relativisation (which is necessary for updates to functionality), which lead to a test that did not run correctly.

Opening and Submitting the PR

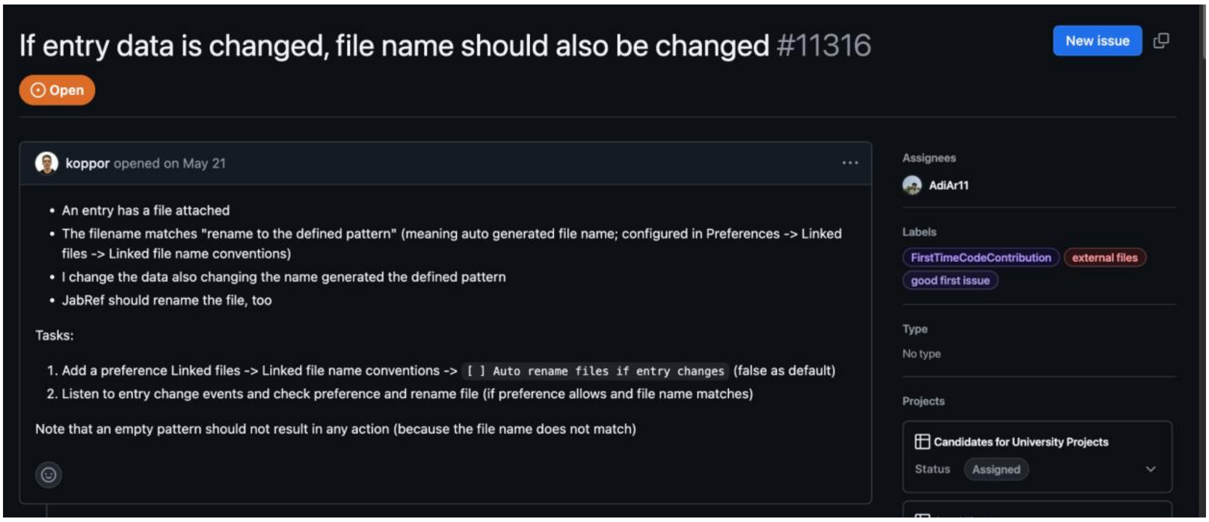
It was very easy to open and submit the PR.

GitHub Username: u7676493 (https://github.com/u7676493)

Collaborator Contributions: subhramit - Reviewed my work and suggested changes, koppor

– Reviewed my work and approved of changes

Documenting Issues #11316: Aditya Arora



If entry data is changed, file name should also be changed #11316

koppor opened on May 21

- An entry has a file attached
- The filename matches "rename to the defined pattern" (meaning auto generated file name; configured in Preferences -> Linked files -> Linked file name conventions)
- I change the data also changing the name generated the defined pattern
- JabRef should rename the file, too

Tasks:

1. Add a preference Linked files -> Linked file name conventions -> `[ ] Auto rename files if entry changes` (false as default)
2. Listen to entry change events and check preference and rename file (if preference allows and file name matches)

Note that an empty pattern should not result in any action (because the file name does not match)

Assignees

AdiAr11

Labels

FirstTimeCodeContribution external files good first issue

Type

No type

Projects

Candidates for University Projects

Status Assigned

Issue: https://github.com/JabRef/jabref/issues/11316

Pull Request: https://github.com/JabRef/jabref/pull/12024

Complete code on my Fork: AdiAr11/jabref at fix-for-issue-11316

GitHub Profile: AdiAr11 (Aditya Arora)

Code changes, as there are many and it is difficult to write in the report.

Issue Description

Automatic Renaming of Linked Files when Entry Data Changes

In the JabRef reference manager, users can link external files (e.g., PDFs) to bibliographic entries. The user can rename these files based on the data entries and the selected “Filename format pattern” in the settings. However, when users update the entry data (e.g., changing the tile), the linked file names do not automatically update to reflect these changes.

Current Behavior

1. An entry in JabRef can have attached files

2. These files can be named according to a defined pattern (configured in Preferences -> Linked files -> Linked file name conventions)
3. Currently, if you change the entry data (like author, title, year, etc.), the attached file name doesn't automatically update to match the new pattern

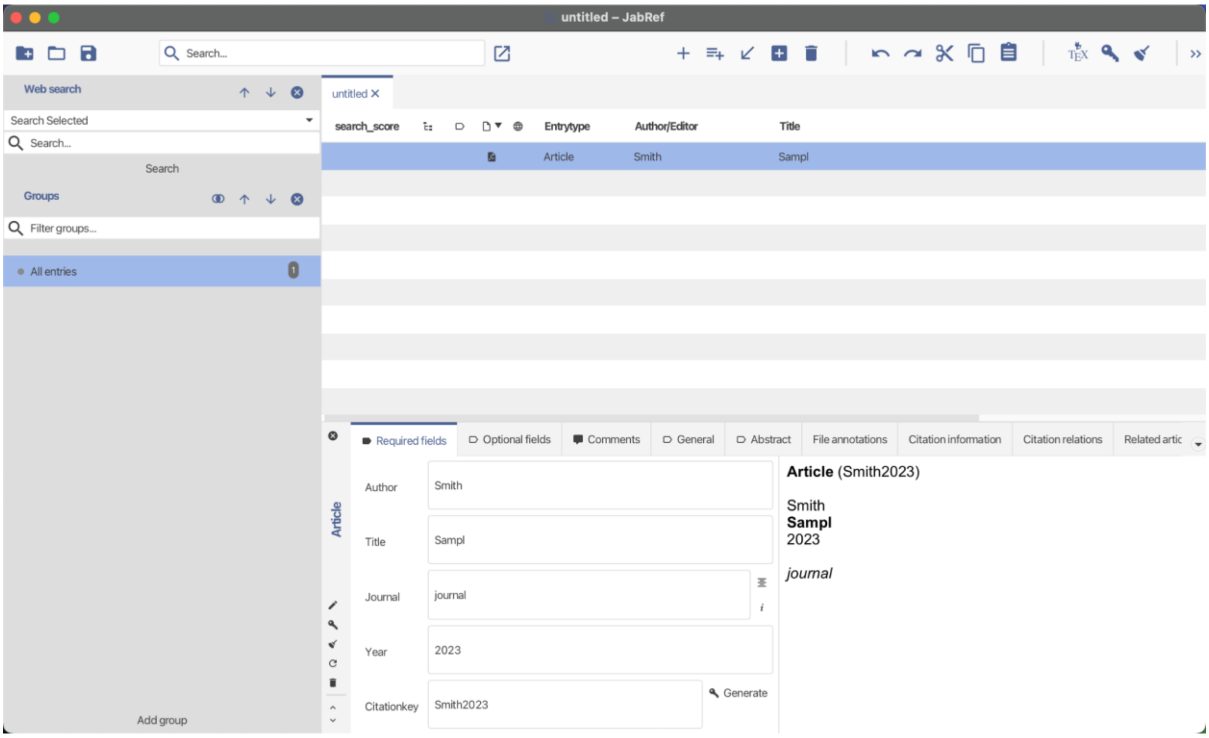
This results in a discrepancy between the file names and the updated metadata.

How the Issue Is Triggered:

1. **Initial Setup:**
    a. The user defines a filename pattern (e.g., [bibtexkey] - [title]) in the preferences.
    b. Users fills the required entry fields and links a file to an entry. The user then right clicks on the file and selects “Rename file to defined pattern” and it is named according to the current entry data.

2. **Updating Entry Data:**
    a. The user edits the entry's fields that are part of the filename pattern (e.g., updates the citation key or modifies the title).

3. **Resulting Discrepancy:**
    a. Despite the changes, the linked file's name does not update to reflect the new metadata.
    b. This causes the file name to be out of sync with the entry's data.



Entrytype	Author/Editor	Title
Article	Smith	Sampl

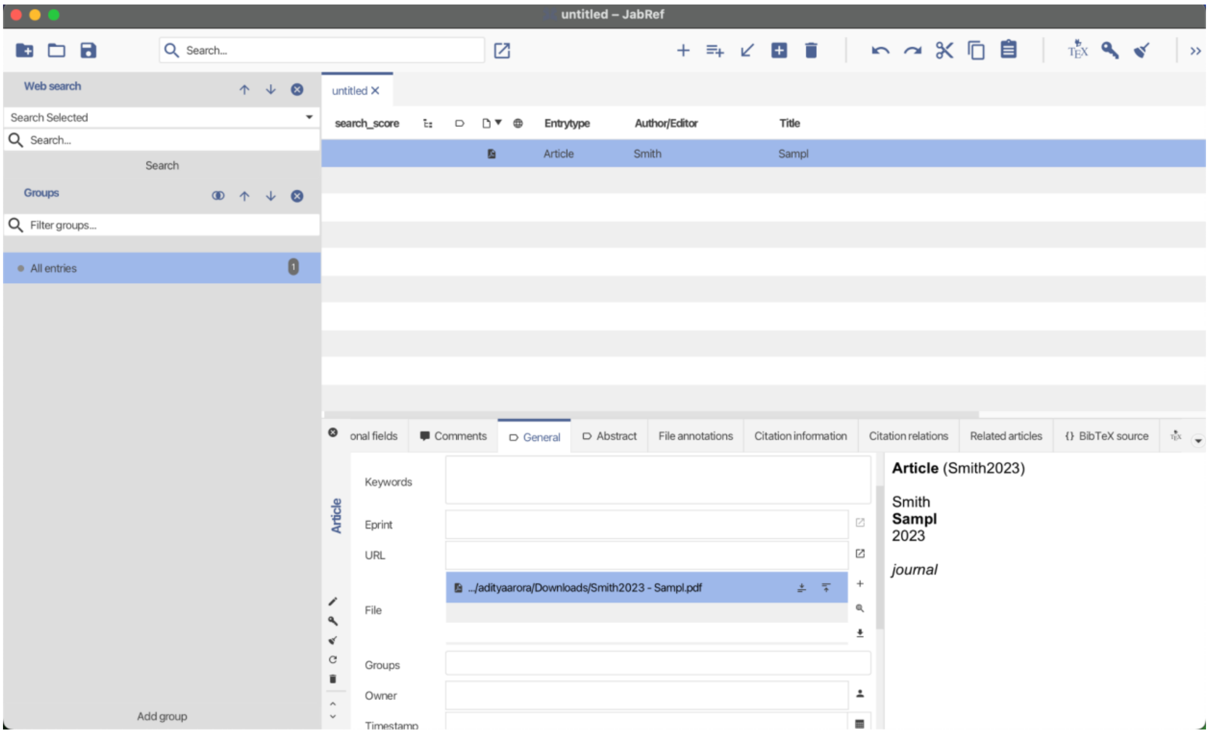
Author: Smith
Title: Sampl
Journal: journal
Year: 2023
Citationkey: Smith2023

Article (Smith2023)

Smith
Sampl
2023

journal

Figure: User creates a new Library and creates a new Entry of type Article by clicking Command + N and fills in the required fields.



Entrytype	Author/Editor	Title
Article	Smith	Sampl

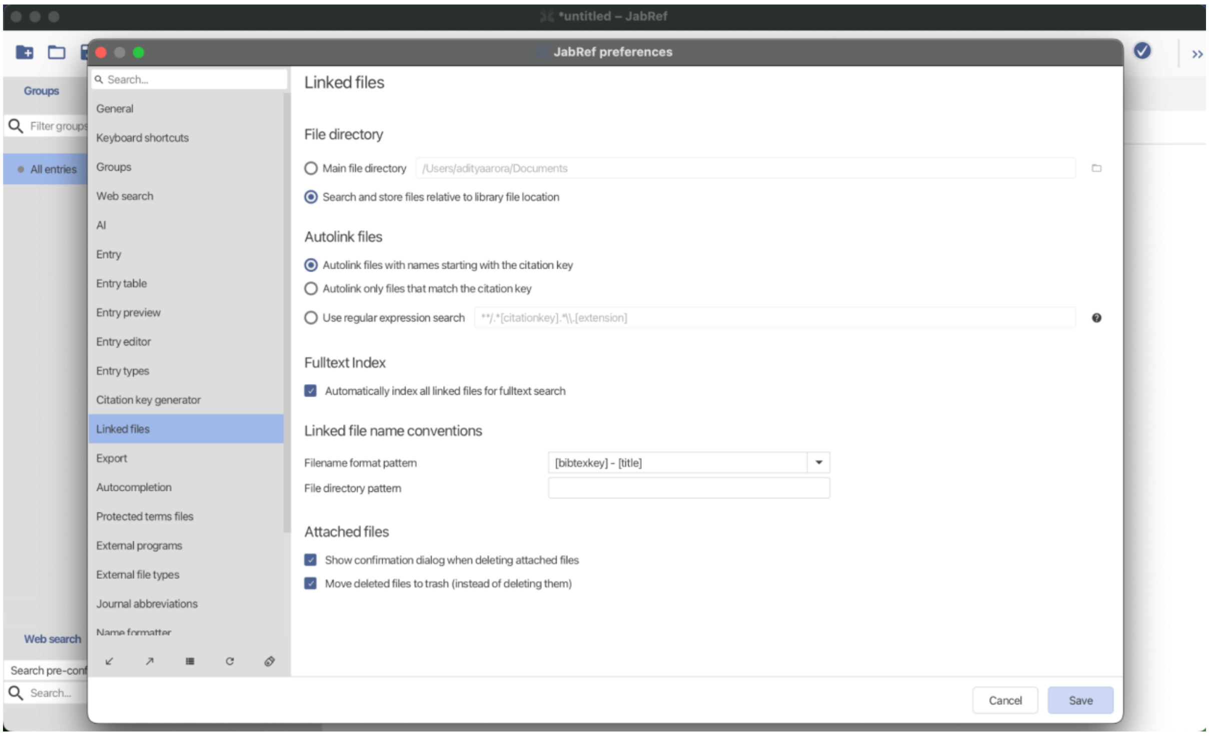
Keywords
Eprint
URL
File: ../adityaarora/Downloads/Smith2023 - Sampl.pdf
Groups
Owner
Timestamp

Article (Smith2023)

Smith
Sampl
2023

journal

Figure: User links a file, right clicks on it and selects” Rename file to defined pattern”



Linked files

File directory

○ Main file directory /Users/adityaarora/Documents

● Search and store files relative to library file location

Autolink files

● Autolink files with names starting with the citation key

○ Autolink only files that match the citation key

○ Use regular expression search **/.*[citationkey].*\\.[extension]

Fulltext Index

☑ Automatically index all linked files for fulltext search

Linked file name conventions

Filename format pattern [bibtexkey] - [title]

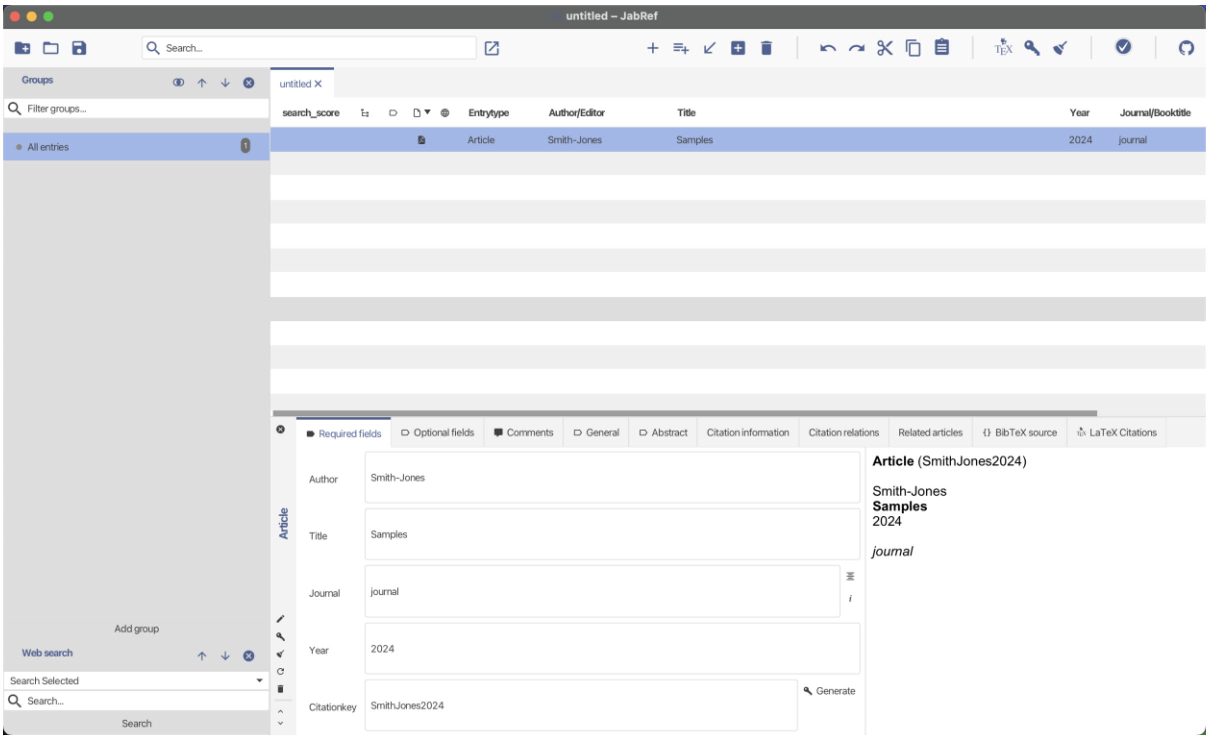
File directory pattern

Attached files

☑ Show confirmation dialog when deleting attached files

☑ Move deleted files to trash (instead of deleting them)

Figure: Linked file name gets changed to the selected Filename format pattern.



search_score	Entrytype	Author/Editor	Title	Year	Journal/Booktitle
	Article	Smith-Jones	Samples	2024	journal

Author Smith-Jones

Title Samples

Journal journal

Year 2024

Citationkey SmithJones2024

Article (SmithJones2024)

Smith-Jones
Samples
2024

journal

Figure: User changes the entry fields. The Linked filename does not change, creating discrepancy.

Example Scenario

1. **Before Change:**
    a. Entry has bibtexkey: Smith2023
    b. Entry has title: Sampl
2. Linked file name after selecting "Rename file to defined pattern" (using pattern [bibtexkey] - [title]): Smith – Sampl.pdf
3. **After Change:**
    a. User updates bibtexkey to SmithJones2024
    b. User updates title to Samples
    c. Linked file name remains: Smith – Sampl.pdf

Requested Solution:

- Add a new preference checkbox option (at Location: Preferences -> Linked files -> Linked file name conventions).
- Implement an automatic renaming feature that updates linked file names whenever the relevant entry data changes.

After understanding the issue, I planned my approach at a high level before starting the actual implementation. My strategy was as follows:

1. **Add a User Preference Checkbox**: Introduce a new checkbox in the preferences under **"Preferences → Linked files → Linked file name conventions"** to allow users to enable or disable the automatic renaming feature.
2. **Listen for Entry Field Changes**: Implement an event listener that detects when any relevant field in a bibliographic entry changes. I intended to use print statements to trace where these events occur in the codebase.
3. Change the filename to the Selected "Filename format pattern"

- I had to go through the following documentation, which was provided in the issue only, to understand how this worked:
    - https://docs.jabref.org/advanced/externalfiles
    - https://docs.jabref.org/finding-sorting-and-cleaning-entries/filelinks
- I also went through all the steps in their developer documentation to setup the project and get it running on my local machine
    - https://devdocs.jabref.org/getting-into-the-code/guidelines-for-setting-up-a-

local-workspace/
    - Even JabRef's code style: https://devdocs.jabref.org/getting-into-the-code/guidelines-for-setting-up-a-local-workspace/intellij-13-code-style.html
- Also went through their Contribution documentation:
    - https://github.com/JabRef/jabref/blob/main/CONTRIBUTING.md

Brief Overview of Specifics of my Implementation

1. **Adding the Preference Checkbox:**

    a. Locating UI File:

    i. Used global search for "Linked file name conventions"

    ii. Found text in LinkedFilesTab.fxml

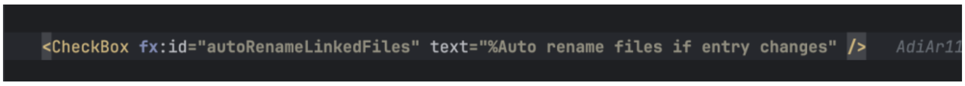
    iii. Located related files:

    1. LinkedFilesTab.java
    2. LinkedFilesTabViewModel.java

    b. **Adding the Checkbox**:

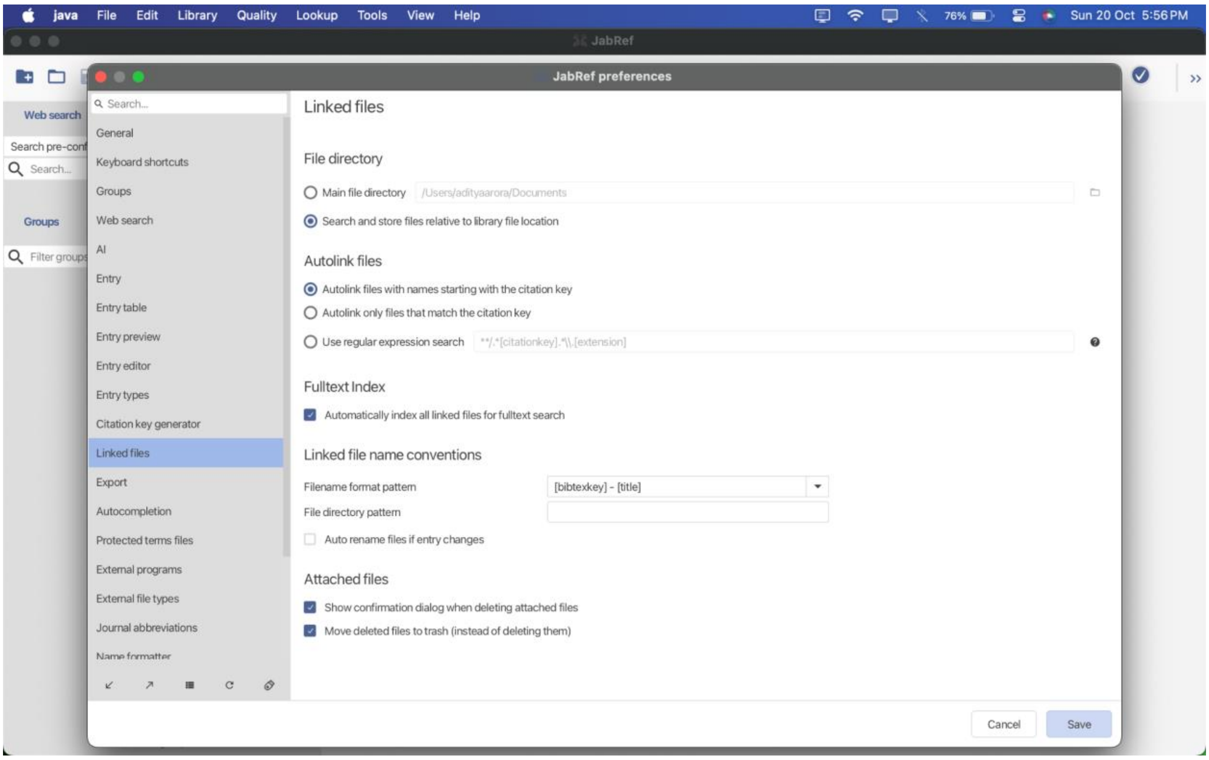
    i. I noticed that the fxml UI file already had existing checkboxes for other preferences.

    ii. To maintain consistency, I copied one of the existing checkbox elements and modified it's: *id* and *text*



```
<CheckBox fx:id="autoRenameLinkedFiles" text="%Auto rename files if entry changes" />   AdiAr11
```

    iii. Connected it to the appropriate property in the LinkedFilesTabViewModel.java to store the user's preference.



Linked files

File directory

○ Main file directory /Users/adityaarora/Documents

● Search and store files relative to library file location

Autolink files

● Autolink files with names starting with the citation key

○ Autolink only files that match the citation key

○ Use regular expression search **/.*[citationkey].*\\.[extension]

Fulltext Index

☑ Automatically index all linked files for fulltext search

Linked file name conventions

Filename format pattern: [bibtexkey] - [title]

File directory pattern:

□ Auto rename files if entry changes

Attached files

☑ Show confirmation dialog when deleting attached files

☑ Move deleted files to trash (instead of deleting them)

2. **Listening for Entry Field Changes:**

   a. Identify Event Dispatch Points:

      i. Inserted print statements in the code to trace where event is triggered when a field changes.

      ii. It is determined that this event is dispatched when relevant fields in a BibEntry are modified.

   b. Implement Event Listener:

      i. Created a new class LinkedFileAutoRenamer.java

      ii. Added a method annotated with @Subscribe to listen for this event.

      iii. Registered this listener with the database in LibraryTab.java to receive events.

3. **Renaming Linked Files:**

   a. **Extract Relevant Fields from Pattern:**

      i. Used BracketedPattern.expandBrackets with a custom function to extract field names from the filename pattern.

      ii. Enhanced FileUtil.java with renameLinkedFileToName method.

      iii. Utilized this function in the data entry change event listener to change the file names of all linked files.

PR Process:

1. **Forking and Cloning the Repository:**
    a. **Forked** the JabRef main repository on GitHub to create a personal copy.
    b. **Cloned** the forked repository to my local machine to begin development.
2. **Setting Up the Local Workspace:**
    a. Followed JabRef's setup-guide to configure my environment in IntelliJ IDEA.
    b. Ensured that the project ran correctly and adhered to JabRef's code style conventions.
3. **Creating a New Branch:**
    a. Created a dedicated branch for the feature, named fix-for-issue-11316
    b. git checkout -b fix-for-issue-11316
4. **Implemented the Code Changes and regularly made commits with good message description.**
5. **Manually tested the implementation**
6. **Pull Request Submission**
    a. Created PR #12024, which was a Draft Pull Request initially, when I was still working on implementing the feature.
    b. Opened the Draft Pull Request for Review when I was done with my changes
    c. Added "fixes #11316" to auto-close the issue
    d. Attached Screenshots for UI changes
7. **Review Process**
    a. Monitored the PR for comments and feedback from maintainers.
    b. Made requested changes promptly, such as refining code logic and removing code duplicity.
    c. Pushed additional commits to the same branch, which automatically updated the PR.

- I don’t think I can explain all relevant parts of the repository that relate to my issue. For example, I found a method already existing in FileUtil.java to parse file name pattern. I do not understand how it works, I just apply Abstraction principle of OOP, where I believe that it will work, and use it to implement the feature requested.

- My solution integrates neatly with the JabRef project by utilizing existing classes like BracketedPattern.java and LinkedFileHandler.java, ensuring consistency and trying to avoid code duplication. I adhered to the project's coding standards and UI patterns when adding the new preference checkbox in LinkedFilesTab.fxml.

- The solution I wrote should be easy enough to understand for someone who is already familiar with the codebase, although it can take some time to understand it as it is quite long.

Activity 5 - Work/Consistency in the Repository

It was included in the previous content.