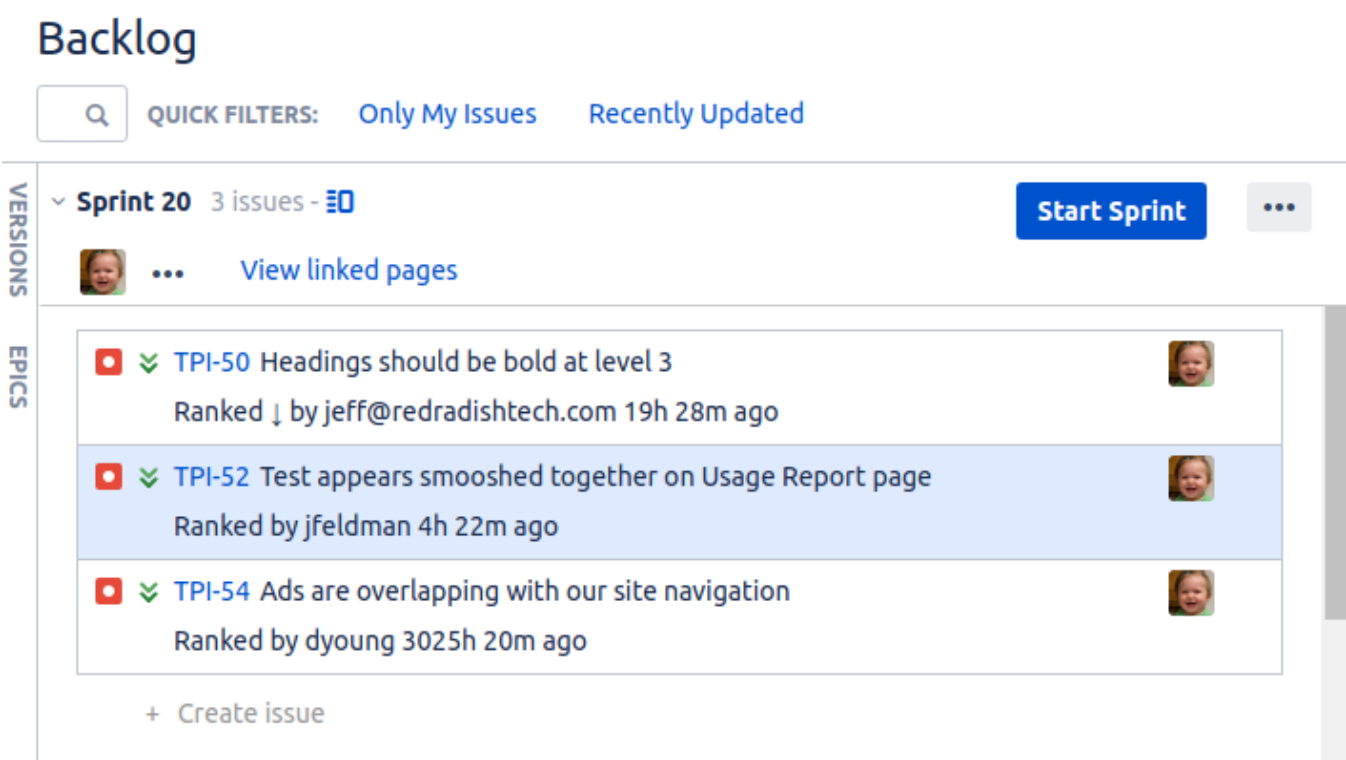


View who last re-ranked the Board

Jira Boards allow you to re-rank issues by dragging them up and down. Sometimes, when planning, it is useful to know *who* last re-ranked the issue, and *when*.

Here we describe a ScriptRunner-based solution that adds an info field to each issue:






Implementation

The implementation is in two parts:

1. A listener which is triggered on the re-rank event, and stores relevant information in hidden custom fields on the re-ranked issue.
2. A script field which interprets the hidden custom fields to display the final information.

Define the custom fields

First, define three regular Jira fields of the indicated types. These will store our rank info:

Rank Last Changed When the rank changed last. Used to render Rank Change Info script field. See https://...	Date Time Picker	Global (all projects)	0 screens	
Rank Last Changed By Who last changed the issue rank. Used to render Rank Change Info script field. See htt...	User Picker (single u...	Global (all projects)	0 screens	
Rank Prior To Last Change Stores historical internal ranking string. Used to render Rank Change Info script field. ...	Text Field (single line)	Global (all projects)	0 screens	

Create the listener

Save the following Groovy script to \$JIRAHOME/scripts/rankchangeinfo_listener.groovy:

rankchangeinfo_listener.groovy

```
/**
 * Populates 'Rank Last Changed', 'Rank Last Changed By' and 'Rank Prior To Last Change' custom fields, for
 * later use by the 'Rank Change Info' script field.
 */
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.issue.IssueManager
import com.atlassian.jira.issue.fields.CustomField
import com.atlassian.jira.datetime.DateTimeFormatterFactory
import com.atlassian.jira.datetime.DateTimeStyle
import com.atlassian.jira.issue.index.IssueIndexingService
import com.atlassian.jira.issue.Issue
import com.atlassian.jira.event.type.EventDispatchOption

def dateTimeFormatterFactory = ComponentAccessor.getComponentOfType(DateTimeFormatterFactory)
def formatter = dateTimeFormatterFactory.formatter().forLoggedInUser().withStyle(DateTimeStyle.DATE_PICKER)

IssueManager issueManager = ComponentAccessor.getIssueManager()
// event is of type com.atlassian.greenhopper.service.lexorank.balance.LexoRankChangeEvent. There is no JavaDoc
// on the web, so I had to figure out the structure by decompiling. There is a getIssueId() method, called here:
if (event && event?.issueId) {
    //log.warn "Event re-ranked: " + event.getProperties()
    def Issue issue = issueManager.getIssueObject(event.issueId as Long)
    if (issue) {
        def String dateStr = formatter.format(event.time)
        def currentUser = ComponentAccessor.getJiraAuthenticationContext().getLoggedInUser()
        CustomField timeCf = ComponentAccessor.customFieldManager.getCustomFieldObject(16121) // Rank Last
Changed
        CustomField userCf = ComponentAccessor.customFieldManager.getCustomFieldObject(16122) // Rank Last
Changed By
        CustomField pastrankCf = ComponentAccessor.customFieldManager.getCustomFieldObject(16123) // Rank Prior
To Last Change
        issue.setCustomFieldValue(timeCf, event.time?.toTimestamp())
        issue.setCustomFieldValue(userCf, currentUser)
        def List<String> pastranks = (issue.getCustomFieldValue(pastrankCf) as String)?.split(" ")??.toList()
        if (!pastranks) pastranks = []
        pastranks.add(0, event.newRank as String)
        pastranks = [pastranks[0], pastranks[1]]
        issue.setCustomFieldValue(pastrankCf, pastranks.join(" "))
        log.info("Updated ${issue} '${timeCf}' field value to ${dateStr}, '${userCf}' field value to
${currentUser}")
        def issueIndexingService = ComponentAccessor.getComponent(IssueIndexingService)
        issueIndexingService.reIndex(issue)
        // Useful guide to updating issues: https://community.atlassian.com/t5/Agile-articles/Three-ways-to-
update-an-issue-in-Jira-Java-API/ba-p/736585
        issueManager.updateIssue(currentUser, issue, EventDispatchOption.DO_NOT_DISPATCH, false);
    }
}
```

Ensure the file is readable by the Jira runtime user.

Next, in Jira, type 'gg' (admin shortcut) 'Script Listeners' to bring up the ScriptRunner Script Listeners page. Add a new Custom listener, listening to event LexoRankChangeEvent, and invoking our groovy script:

Custom listener

Write your own groovy class as a custom listener.

Note Store rank change info, for later use by 'Rank Change Info' field. See <https://c>

An optional note, used only for your reference.

Project(s)

× All projects

Filter on events for these projects. Some events, eg **User** events, are not associated with a project.

Events

LexoRankChangeEvent ×

×

▼

Which events to fire on .

Inline script

1

Enter your script here

→ ↶ ?

Script file

rankchangeinfo_listener.groovy

×

- × Script197.groovy:20
[Static type checking] - No such property: issueId for class: com.atlassian.jira.event.issue.IssueEvent
Possible solutions: issue @ line 20, column 14.
- × Script197.groovy:22
[Static type checking] - No such property: issueId for class: com.atlassian.jira.event.issue.IssueEvent
Possible solutions: issue @ line 22, column 48.
- × Script197.groovy:33
[Static type checking] - No such property: newRank for class: com.atlassian.jira.event.issue.IssueEvent
@ line 33, column 26.

Path to the script accessible on the server - or fully-qualified class name class in the form com.acme.MyListener

Preview

Update

Cancel

There will be some unavoidable static type checking errors because ScriptRunner is not able to find the LexoRankChangeEvent class, which is technically part of a plugin.

Note that the LexoRankChangeEvent class is very much not a public API (there isn't even Javadoc on the internet), so don't be surprised if Atlassian break compatibility at some point.

Create the display field

Next, we create the field that displays the final text on the Board.

First, create a file \$JIRAHOME/scripts/rankchangeinfo.groovy, containing:

rankchangeinfo.groovy

```
/**
 * The 'Rank Change Info' script field displays when the issue's Rank last changed, and who changed it.
 */
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.issue.fields.CustomField
import org.apache.commons.lang.StringEscapeUtils

import org.joda.time.DateTime;
import org.joda.time.Duration;
import org.joda.time.format.PeriodFormatter;
import org.joda.time.format.PeriodFormatterBuilder;
import org.joda.time.Interval;

public String direction(String rankHistory) {
    if (!rankHistory) return ""

    def (String cur, String prev) = rankHistory.split().toList()
    if (!prev) return ""
    if (cur < prev) { return "" } else { return "" }
}

public String fmt(Interval interval) {
    Duration duration = interval.toDuration();
    // Note: I have chosen to not display years/months/days, and instead
    // just roll them up into a large hours figure, on the assumption that
    // beyond 24h, users don't care exactly how long ago the modification
    // was made, and a large number conveys "long ago" cleaner than y/m/w/d
    // numbers. If you disagree, see
    // https://stackoverflow.com/questions/26291271/periodformatter-not-showing-days
    // for how to display those details.
    PeriodFormatter formatter = new PeriodFormatterBuilder()
        .appendHours()
        .appendSuffix("h ")
        .appendMinutes()
        .appendSuffix("m ")
        .toFormatter();
    String formatted = formatter.print(duration.toPeriod());
    if (formatted == "") { formatted="just now" } else { formatted += "ago" }
    return formatted
}

log.debug("rankchangeinfo(${issue})")
CustomField timeCf = ComponentAccessor.customFieldManager.getCustomFieldObject(16121L) // Rank Last Changed
if (!timeCf) return
CustomField userCf = ComponentAccessor.customFieldManager.getCustomFieldObject(16122L) // Rank Last Changed By
if (!userCf) return
CustomField pastrankCf = ComponentAccessor.customFieldManager.getCustomFieldObject(16123L) // Rank Prior To
Last Change
if (!pastrankCf) return
def changed = issue.getCustomFieldValue(timeCf)
if (!changed) return;
def changedBy = issue.getCustomFieldValue(userCf) as com.atlassian.jira.user.ApplicationUser
if (!changedBy) return;
def pastrank = issue.getCustomFieldValue(pastrankCf) as String

DateTime now = new DateTime()
DateTime then = new DateTime(changed)
Interval interval = new Interval(then, now);
return "Ranked ${direction(pastrank)} by ${changedBy?.username} ${fmt(interval)}"
```

In ScriptRunner's Script Fields page, create a **Rank Change Info** field that uses this:

Custom Script Field ?

Create your own custom scripted field.

Field Name

The name of the custom field that you are creating

Field Description

The description of the custom field that you are creating


Note

An optional note, used only for your reference.

Template

Template to represent your field. Match to the output of your script.

Script file

 Script184.groovy:19
[Static type checking] - Multiple assignments without list expressions on the right hand side are unsupported in static type checking mode
@ line 19, column 41.

Path to the script accessible on the server

Inline script

1

Enter your script here



Preview Issue Key

Issue key for preview. Only used for the preview function below.

Preview

Update

Cancel

Result Logs 0 Timing

Ranked ↓ by jeff@redradishtech.com 19h 52m ago

In relevant Board's configuration, edit the Card Layout to display our new **Rank Change Info** field:

Card layout

Cards can be configured to show up to three extra fields.

Backlog

Field Name

Affects Version/s

Add



Rank Change Info

Delete

Active sprints

Field Name

Affects Version/s

Add



Rank Change Info

Delete

All done. Now try dragging issues around on a Board. After a manual screen refresh (unfortunately necessary) you should see the custom field value update.