

Rewriting Jira Admin Documentation - procedures and outcomes

by Eddie Meardon

This page covers the problems facing a legacy documentation set in Jira, outlines the steps taken to resolve these problems, and covers the impact of this work.

Part 1: The State of the Docs

Many of the the Jira Admin docs are as old as Jira itself. For years, we (myself included) have promised to ‘someday’ give these docs the overhaul they so badly need. In the meantime, they continued to grow organically, making large scale renovations more daunting and necessary. This page covers my work on two specific sections of the Jira Admin docs: the Fields and Permissions sections.

I rewrote the Fields sections because in 2025, our year of Labubu, I was part of a team that redesigned how Fields work in Jira which hadn’t been meaningfully redesigned since 2008. As part this work, I (as Content Designer) was to update the documentation, which also hasn’t been meaningfully updated since 2008. However, the docs were so out of shape that there was no way to simply update them, and the current metrics of doc health suggested these pages were in need of deep repair.

I rewrote the Permissions docs because it’s one of the most disliked experiences in Jira. While redoing Permissions at the same level as Fields wasn’t in the cards, the documentation faced the same quality problems. After consulting with our Customer Support Services team (the CSS team), we estimated that the number of people writing in for help because of these docs quality was costing the company approximately \$600k a year in support costs.

The planning stage



I created a plan for updating our documentation that focused on a few key improvements that specifically addressed the concerns of an organically-grown documentation set.

Readability and accuracy

The docs were full of contradictory information, extremely specific edge cases (clearly prompted by a customer support case), and outdated facts in illogical places. All of which made it so I, he who works on Jira, **could not understand how Jira worked based on our docs.**

And I wasn't alone. The Fields pages had a total of 5,714 views in the prior 30 days. In that time span, for every user that marked the documentation as helpful, 10 marked it as not helpful. Users noted that:

- concepts weren't clearly defined
- instructions were often incomplete, outdated, and contradictory
- pages didn't clearly correlate to product-specific documentation, and
- the overall experience was insufficient to guide users through key user journeys.



Scalability and ease of updates

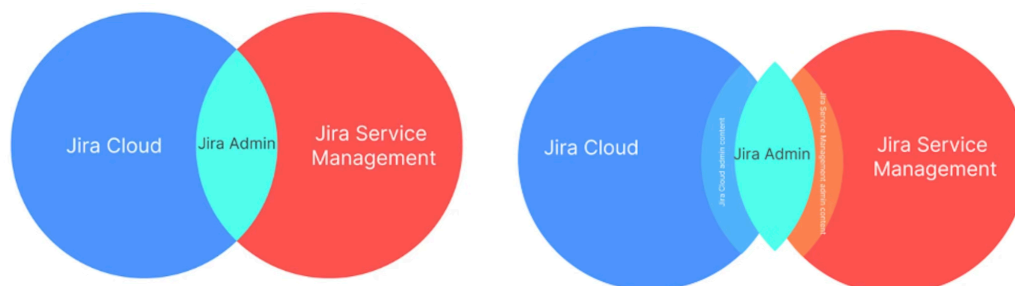
Because the overall structure had been ignored for so long, even small updates were extremely time-consuming. Because these pages didn't use the topic-based authoring concept we've been using for years now, one small change required making the same (or very similar) changes on several pages.

In rewriting these docs, I'd adapt them to the topic-based authoring strategy which reduces redundancies, makes content easier for readers to consume, makes updates easier to apply, and makes for flexible content that can be used in different ways.

Product strategy

These rewrites also address a larger strategic schism in our documentation structure. All flavors of Jira use the same Admin controls experience, so this doc set is intended to be product agnostic. However, because of that pesky organic growth, these lines blurred; admin information was duplicated into product-specific spaces and updated to be only relevant to that product, and procedures for specific products were added to this shared Admin space.

And so the model became less focused:



As part of cleaning up these two spaces, I could restore the clearcut lines which would make navigating, reading, and updating the docs easier.

AI readiness

Lastly, as Rovo (our AI agent) was gaining traction, we planned to train it on our documentation so it can answer specific questions users have about Jira. However, because of all the quality problems already mentioned, Rovo had hilariously low accuracy benchmarks when trained on the existing documentation. In order to improve the accuracy of our AI answers, we needed to clean up the docs.

Cleaning up the docs

With both the Fields and Permissions doc sets, I first reviewed and audited what was currently there. I flagged content based on the following categories:

- what could be kept with minimal changes
- what we should keep but update
- what needed to be removed entirely and
- content that we needed to keep, but move to new sections.

I tried to use AI to speed up my audit, but the same fears I expressed in the “AI Readiness” section above made this a moot point. The content was simply too messy, and it required that the auditor (in this case, me) to know the context behind what it was trying to say.

The majority of the work was simply removing old content. I flagged the extreme edge cases or niche user experiences for removal. Sometimes, they were only helpful to a very small cohort of users, and other times they addressed specific misconfigurations; either way, they were more confusing than helpful. I noted the main takeaway of the content, if there was one, to be worked into future rewrites more intuitively.

I also removed concepts and features that are no longer in the product, such as something that was once called Renderers. Many years ago, this concept had been reduced to a simple (and far more intuitive) dropdown menu, yet the page about it had nearly 1,500 words.



The next major task was to consolidate redundant procedures. For example, one page had four tasks on it. While each ostensibly had different outcomes, the instructions themselves were largely redundant. In the image below, the lines highlighted in orange were the same across all four examples; green was shared with three, pink with two, and the blue lines are the truly unique content from each section:

- | | | | |
|--|---|---|--|
| 1. Select > Work items. | 1. Select > Work items. | 1. Select > Work items. | 1. Select > Work items. |
| 2. Under FIELDS , select Custom fields. | 2. Under FIELDS , select Custom fields. | 2. Under FIELDS , select Custom fields. | 2. Under FIELDS , select Custom fields. |
| 3. Find the field you want to delete and select Actions (...) > Move to trash. | 3. Select the Trashed tab at the top of the table. | 3. Select the Trashed tab at the top of the table. | 3. Select the Trashed tab at the top of the table. |
| | | 4. Find the field you want to restore and select Actions (...) > Restore field. | 4. Find the field you want to restore and select Actions (...) > Permanently delete field. |

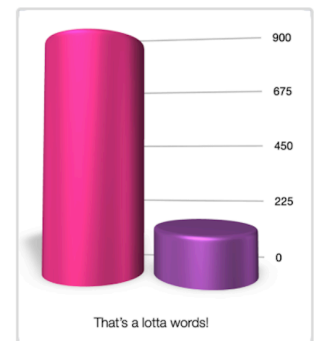
These were reduced to one set of instructions with different final steps, based on the nuances of how users wanted to remove a field.

I also removed content that was factually accurate, but functionally unnecessary like:

*You can find [fields you've marked for deletion] in the **Trashed** tab.*

This is obvious from the UI, and can be removed from the docs.

This type of cleanup work constituted the bulk of my changes, and ruthlessly whittled the word count on one page from 811 to 160.



Rewriting the docs

As with any good hero's journey, only once you've reached the your lowest point can you earn redemption. At this point, my notes were a cacophony of stricken content and ill formed notes. But now that I'd removed the content that had no place in this doc set, I started to rebuild it:

- I arranged the content (in its roughly edited format) into new pages that aligned with concise topics
- I tested procedures for configuring Fields and Permissions in product, and made corrections to ensure they were up to date
- I added new content that explicitly associated concepts whose relationship was inferred before. These pages included things like "How do I set up fields in



my Jira site?” which outlined the overall process from end to end, with links to specific procedures at each stage

- I wrote generic but still helpful examples of how concepts work, including how unrelated decisions can impact their experience

While doing so inflated my previously-ruthlessly low word count, it was actually relevant and filled in the gaps left by the previous documentation.

Measuring impact

At the time of writing, it’s too early to have user metrics that show improvement in either the Permissions or the Fields documentation, but I have already won because the content is better even just by our own definitions and standards.

The Fields section started with 26 pages of content. It now has 21 new and focused topics. I deleted 7 topics and created 5 new ones. Those new pages have an average Flesch-Kincaid readability score of 6.7 (our goal is 7), 7,300 fewer words, and it only consumed a paltry 20% of my sanity.

Section title	Flesch-Kincaid	# words before	# words after	difference
Configure fields that can be used in any project	7.2	3,772	1,544	2,228
Configure field contexts in your site	7.5	6,160	1,082	5,078
Manage field configurations in work items	7	1,112	1,367	-225
Manage work item screens	5	590	302	288
Total:	6.7 (average)	11,634	4,295	7,369

The Permissions content started with 22 pages of content with 33,271 words. I unpacked that into 34 topics, which might seem like a step backwards, but they contain 25,567 fewer words. *Animal Farm* by George Orwell is about 29,000 words, so I’ve basically removed a novella from this section of our documentation.

Again, we won’t see any user statistics for a few months at least. That said, early indications from the CSS team suggest that we’re on track to realize the majority of the estimated \$600k in support costs.

Lastly, these updates have improved the accuracy of Rovo's answers after being trained on them. Though early, the accuracy metric rose from 21% to 45% (note: the models are still being built and are expected to improve over time).

Anecdotally, the procedures I've been given during testing have been more workable than previous versions. Still not amazing, but the names of buttons and the order of operations are far more intuitive than they were before.