

# Synchronizers Overview and Performance



Synchronizers were the original mechanism for building dynamic structures, which was replaced by Automation in Structure 3.0 and is gradually faded out. If it's a new instance, where no synchronizers exist, it's recommended to [remove permissions to use them](#) from all users and use Automation instead. If they are already in use, we recommend to gradually "migrate" all structures to use Automation until Synchronizers are not used at all.



Starting from version 5.0 Synchronizers are hidden from the UI for all new installations. If you have an existing installation and but you have no synchronizers installed anymore, you can hide them from the UI through the system option.

Synchronizers are the processes which you setup for individual structures and which run continuously in the background tracking changes both in Jira and in structures and when there is a change, make the required updates. For example, a Filter synchronizer will be tracking all changes done to your Jira issues and once the change happens, check if any changes should be done to the structure (add a new issue, which now matches the filter JQL or remove the one that now doesn't). Link synchronizer will be checking for the changes done to issue links and if a link is updated for one of the issues in a structure, it will adjust the structure accordingly. If you change the structure, for example, move a child issue under a new parent, it will remove the existing link and will create a new one.

Note that synchronizers make changes to the "static" content. You can think of them as of a person, who manually changes either the structure, or issue data to keep things in sync. If you switch off or remove the synchronizers, the synchronization will stop and the structure will stay in its last state. You can edit it further manually.

## Potential Conflicts

Since synchronizers run as independent processes each responsible for its particular part, they can interact with each other. This may result in some unexpected changes and this is one of the main reasons, why they were eventually replaced with Automation. Here is one common example:

A user wants to see the issues that are the result of a certain JQL query on the top level and all issues linked to them on the level below. The user also wants the top level issues removed from the structure if they no longer pass the JQL. So the user sets up a Filter synchronizer for the JQL with the Remove option and a Links synchronizer, which pulls in linked issues. Here is what may happen as a result:

1. Filter synchronizer pulls in the top level issues.
2. Link synchronizer sees there are some issues added to the structure, checks if they have any issues linked to them with the selected link type and pulls them in.
3. Filter synchronizer sees there are new issues pulled in and checks if they pass the JQL filter. If they don't, it removes them from the structure. The child issues do not pass the query of the parents, so they are removed.
4. Link synchronizer sees that the issues are removed - for this synchronizer it means that someone doesn't need these child issues, so it removes the links between the parents and the children.

As a result the user would lose a number of links, without realizing it. There are ways to avoid such situations (for example, do not select the remove option for the Filter synchronizer and then for removal add another one, which will only remove the entire branches, not just the child issues). But this will work for the more experienced users only - others still can make this or similar mistake.

You can also potentially get into situations, when several synchronizers revert each others changes until the built-in mechanism stops this. Since they operate with Jira data, if you have several structures with the same sub-sets of issues, you may end up with a situation, when synchronizers in one structure react to synchronizes in another structure.

As confusing as it can be, the good news there is a way to quickly revert such changes if they occur with the synchronizers Undo feature.

## Synchronizers Undo

To revert changes done by the synchronizers, do the following:

1. Go to Jira Administration | Structure | Support.
2. Click View Synchronizer Audit Log.
3. Specify the time period when the synchronizer made the unwanted changes.
4. If you know what exact structure was affected, specify the id of structure for which the synchronizers were configured. You can see the structure id in the URL when you have the structure open.
5. If you know the exact synchronizer, which actions you need to undo, specify the id of this synchronizer. You can find it if you go the Synchronizers Settings page of a particular structure and hover over any of the links in the Operations column. You'll see the URL with the id in the browser status bar.
6. Click the Show button to see the list of all changes done by the synchronizer. The changes are done in batches, so you'll see the number of "actions" in each batch - you can click Expand to see all the details.
7. Select the ones you'd like to undo and switch to the Undo tab.
8. Review the list of actions, which will be reverted and click Undo.

## Synchronizers Performance Considerations

When it comes to performance, there are a few things you need to keep in mind. Once you set them up, all enabled synchronizers will be running continuously in the background, checking for issues updates and changes in the structures for which they have been setup and making the required changes. This means that even if you are not using some structure, but have some synchronizers setup for it, they will keep running and will be making updates. If you have a large number of structures with synchronizers (a few hundreds), their total impact may become noticeable, so always make sure to disable them for structures, which are not used anymore. If you delete or archive a structure, its synchronizers won't be running anymore.

Another important thing is that all synchronizers are running in a single thread. They make their updates one by one, which means if there is a long queue of updates, there may be quite a long delay between the time you change something in Jira or Structure and the time, when you see the synchronizers reaction to it. In some extreme cases it may take over several hours and if the system is slow, the time may grow as more and more updates are added to the queue.

## Mixing Automation and Synchronizers

While it's not recommended, it's possible to use Synchronizers together with Generators. The main principle here is that the Generators see the results of the work of Synchronizers, but not the other way around. For example, if you use Filter Synchronizer to pull in issues into a structure, the Extend generator will see these issues and pull in issues linked to them. But if you use an Insert generator to pull in a number of issue to a structure and use the Link Synchronizer to show the child issues, such a setup won't work. The synchronizers do not "see" the dynamic content that is added by the generators.