

# Structure 4.6 Release Notes

 27<sup>th</sup> of March, 2018

Structure 4.6 introduces Automation Timeout safeguard and inline editing of the Status field.

[Download the latest version of Structure and its Extensions](#)

Try It: [Structure Sandbox Server](#) (no installation required)

## 1. Version Highlights

- Automation Timeout safeguards Jira against excessive load that may result from misconfiguration of Structure generators
- Inline editing of the Status field enables users to change an issue's status with a double-click on the Structure grid

## 2. Changes in Detail

### 2.1. Automation Timeout

The Structure [Automation](#) feature is very powerful, and very flexible. It gives the user the building blocks – generators – for constructing dynamic structures. As with any powerful and flexible feature, it's possible to unintentionally create an unanticipated load on the Jira server. The Automation Timeout is a safeguard that protects Jira from unintended server loads.

Some generators already have a considerable level of protection. For example, JQL Inserter offers an option to limit the maximum number of issues it adds to a structure. Extenders have a maximum number of levels to expand a hierarchy. Automation Timeout adds universal protection on top of that by limiting the amount of time Structure app can spend generating a particular structure.

Here's how it works.

When Structure app detects that a structure is being generated for an unreasonably long time (by default, 30 seconds), it **pauses Automation** for that structure. The Structure panel will show the "**Automation was paused**" message and the structure itself will miss the generated content. If the structure contains only dynamic elements, it will appear empty.

If the user clicks "More" link, they can inspect which generators took most of the time.

 Automation was paused. [Less ^](#) [Resume](#)

Structure generation was taking too long, so it was paused. Please consider reconfiguring your generators or removing some of them. Also you can adjust the time limit in structure settings.

|  |        |
|--|--------|
|  Add issues linked by <b>Blocks</b> : parent <b>blocks</b> children | 97.58% |
|  Insert issues: <b>project is not empty</b>                         | 2.42%  |

Any user who can view the structure can resume Automation by clicking **Resume** button. This will cause Structure to try and generate this structure again.

 Note that sometimes structure generation may time out because of general slowness of Jira caused by other factors. It is ok to click Resume and try again in that case, without changing the generators.

Documentation: [Paused Automation](#)

### 2.2. Inline Status Editing

It is now possible to edit an issue's Status inline by double-clicking on its current status in the Status column.

You can then select a new status from a list. The list will show only the statuses that could be the issue's next status, according to the workflow. If a particular transition includes a screen with additional fields, this transition cannot be executed through inline editing and will require [the use of Jira Actions](#).

Documentation: [Editing Issues](#)

### 2.3. Notable Fixes and Improvements

- Fixed: When exporting a structure to a Printable page in Automation Editing Mode, some rows are not shown
- Fixed: When exporting a structure to Excel, large texts are truncated to 16382 symbols
- Fixed: Issues can be added to archived Fix Versions
- Fixed: Progress based on Time Tracking takes folders and pages into calculation
- Fixed: Folders are not shown in the structure on a product page
- Fixed: Duplicates filter is not included in a shared perspective

## 3. Supported Versions

Structure 4.6 and all extensions support Jira versions from 7.2 to 7.8. All editions of Jira (Jira Core, Jira Software, Jira Service Desk) are supported. Jira Data Center is supported.

With respect to other add-ons and custom integrations, this release is backwards-compatible with Structure 3.4–4.5. [Structure.Testy](#) extension, [Colors](#), [Structure.Gantt](#) and integrations with third-party apps should continue working normally.

## 4. Installation and Upgrade

### 4.1. Installing Structure

If your Jira server does not have Structure yet, the installation is simple:

1. Download and install Structure app, either from Atlassian Marketplace or from [Download](#) page.
2. When Add-on Manager reports the successful installation, click Get Started to visit a page with important guidance for the Jira administrator. You may want to also check out the user's Get Started page, available under "Structure" top-level menu.
3. ⚠️ If you have Structure.Pages installed, make sure you've upgraded to version 1.3 or later, both on Jira and on Confluence side. If your Confluence version is not compatible with Structure Helper 1.3, you should stay with version 1.2 for Structure.Pages and Structure Helper apps, but please, note that there are limitations to its compatibility with Structure 4.2 and higher, so Confluence upgrade to version 6.1 or better is recommended.
4. Monitor `catalina.out` or `jira-application.log` for log messages from Structure.

### 4.2. Upgrading Structure



If you're upgrading from version 2.11.2 or earlier, please read [Structure 3.0.0 Release Notes](#).

Upgrade procedure from versions 3.x–4.5 is simple:

1. Consider backing up Jira data. Use **Administration | System | Backup System**. (If you have a large instance and have proper backup strategy in place, you may skip this step.)
2. Back up Structure data. Use **Administration | Structure | Backup Structure** menu item. If you have a lot of structures and a large Jira, consider turning off "**Backup History**" option to avoid long backup process.
3. Install the new version of the plugin.
4. Monitor `catalina.out` or `jira-application.log` for warnings or errors.

## 5. Enterprise Deployment Notes

Structure 4.6 has a number of changes that are particularly important for large installations and Jira Data Center instances.

### 5.1. Automation Timeout

One of the problems we've seen in the past is that one or more improperly configured structures would take a lot of time to generate, consuming system resources. In extreme and rare occasions, system load could make entire Jira instance unresponsive. To prevent that from happening, we're introducing a time limit for structure generation. If a structure takes longer time to get generated, it will be marked as "paused", and all generators in it will be temporarily disabled.

The default timeout is 30 seconds. If some of your structures currently take a long time to generate, they might get paused after the upgrade.

Therefore, we advise you to carefully check your all of your bigger and mission-critical structures in a staging environment before upgrading and adjust the timeouts as necessary. The structures with paused automation will be marked as such on the Manage Structures page.



Note that someone needs to actually open a structure for Automation to run the generation process. A structure that is not being looked at does not consume any system resources.

The timeouts can be adjusted for each structure using "Configure" link on the Manage Structure page. The global default can be adjusted in the **Administration | Structure | Defaults** menu section.

### 5.2. Synchronizers on Jira Data Center

Structure uses only one cluster node to run synchronizers. We've seen some problems with the mechanism used to select a cluster node for running synchronization and hand off its work to a different node in case the original node is stopped or loses connectivity. These problems could cause all synchronizers in the system to stop working.

In this version we're introducing a simpler and more reliable mechanism which uses conditional database updates and Jira's built-in node aliveness checks. A new database table, `AO_8BAD1B_ATOMIC_FLAG`, is introduced to support the new solution. In this version, it is expected to contain only a single row that shows which node is currently responsible for running synchronizers.

You can use the following steps to test synchronization node selection and hand-off:

1. Start Jira (Data Center), install Structure 4.6 and check application logs for messages from *ClusterExclusiveWorkNodeFlag* to determine which node is currently responsible for synchronization.
2. Shut down or disconnect that node and wait for up to 6 minutes, which is the maximum reaction time.
3. Check application logs for messages from *ClusterExclusiveWorkNodeFlag* to see that synchronization was taken over by another live node.

### 5.3. Lucene Index Monitoring

Structure relies on the Lucene index to check users' access to issues, or quickly read a single value for a large number of issues. The index may be temporarily inconsistent during full or project reindexing, index replication or recovery. Structure 4.2 introduced re-index monitoring as a part of a dark feature used to work around a race condition in Jira Data Center. We've updated this mechanism in Structure 4.6 to also handle index replication and automatic recovery, and now we're enabling it for all instances, so that Structure is able to recover when it detects massive index changes.

After the upgrade, a new, undeletable system custom field called "Structure Index Monitor" should appear. This field is used only to track the "reindex" events. It does not contain any data and will not have any effect on issues. If Structure is uninstalled, this custom field disappears.

### 5.4. Old Row Manager Implementation Removed

Structure 3.4 introduced a new implementation of one of its core components used to store temporary rows in the generated structures. The old implementation remained in the code with the ability to switch to it by setting a system property. In over a year since that release the new implementation has proven to be reliable, and we never advised our customers to switch to the old one. In Structure 4.6 we remove the old implementation and its dependencies completely.

If you see folders named *rows0*, *rows1*, etc., under *JIRA\_HOME/structure* folder, they can be safely deleted – they were a part of the older implementation.

### 5.5. Testing on Staging Environment

There are no particular special areas of interest for load testing and stress testing Structure 4.6. We advise running the same testing procedures as you've done for previous upgrades.



Should you have any questions on Enterprise Deployment, let us know at [support@almworks.com](mailto:support@almworks.com).