

# Advanced Configuration and Dark Features

Certain advanced aspects of Structure's behavior might not have dedicated configuration pages, being controlled by application properties or system properties instead. Additionally, some features and behaviors are usually hidden from users by default. These dark features can be turned on for an instance by the Jira administrator.

## Setting Application Properties with the Structure Dark Features and Fine Tuning Interface

The easiest way to add and manage custom Structure properties and dark features is to use the Structure Dark Features and Fine Tuning interface.

- To add a new custom property or dark feature, enter the appropriate Property Key (see below for a list of available keys) and click **Add Property**.
- Once the key is added to your properties list, you can adjust its value by clicking the edit icon (pencil).
- To remove a custom property, click the trash icon.

Key	Value
com.almworks.jira.colors.dataVersion	1
com.almworks.jira.structure.system.refreshedOnStart	false

To access the interface, you must have Jira Administration permissions and enter the interface location directly into your browser: [https://YOUR\\_JIRA\\_ADDRESS/secure/admin/StructureDarkFeatures.jspa](https://YOUR_JIRA_ADDRESS/secure/admin/StructureDarkFeatures.jspa)

## Guidelines for Adding/Removing Property and Values

- When an invalid property value is entered in the table, the default value is applied.
- Spaces are not trimmed, and may result in an invalid value.
- When you delete a property from the admin table, it's property value is set to the default value:
  - If the property was added with our admin interface, the value is set to empty value and the property is removed from the table after a page refresh.
  - If you set the value to empty (without deleting the property), the property will not be removed.

## Setting System Properties

You can set System properties during Startup or using Script Runner.

Both of the following methods can also be used to set Structure properties; but we recommend using the Structure Dark Features and Fine Tuning interface as described above.

## Setting System Properties on Startup

You can set System properties using the `-D` Jira startup option, for example:

```
-Dstructure.sync.guard.email.admin.cycles=5
```

Configuring Jira startup options is described in [this article](#). You will need to restart Jira for the properties to take effect.

## Setting System Properties with Script Runner

You can also set system properties using the [Script Runner](#) add-on.

1. Install Script Runner.
2. Go to **Administration | Add-Ons | Script Runner | Script Console**.
3. Select **Groovy** as the Script Engine.
4. Enter the following code into the Script text box, adjust property name and value as needed, and click **Run Now**.

```
System.setProperty("structure.sync.guard.email.admin.cycles", "5")
```

The changes take effect after you restart Structure, but the properties will be reset to their default values when you restart Jira. In some cases, for settings to take effect you have to reinstall Structure.

## Structure size limit

Property	Default	Explanation
<code>com.almworks.jira.structure.AOBasedStructureManager.forestSizeLimit</code>	100000	The maximum number of rows that one structure can contain.

## Structure Automation limits

Property	Default	Explanation
<code>structure.gfs.generationTimeHardLimit</code>	600	The maximum amount of time that can be spent for Structure generation (in seconds).

## Automation Defaults

Property	Default	Explanation
<code>structure.generator.defaults.disableUpdates</code>	false	When adding generators: <ul style="list-style-type: none"><li>If set to "false" (default) - the "allow changes" box is initially checked.</li><li>If set to "true" - the "allow changes" box is initially unchecked.</li></ul>

## Manual adjustments

Property	Default	Explanation
<code>structure.gfs.manualAdjustments.enable</code>	true	Setting this property to false will disable manual adjustments for the entire Jira Instance. All adjustment-related UI elements and controls will disappear. Existing manual adjustments will be kept in the database, but will not be applied.
<code>structure.gfs.manualAdjustments.maxAdjustmentsPerStructure</code>	2000	The maximum number of manual adjustments per one structure. When this limit is reached, adding new manual adjustments will be impossible. If you reduce this limit, you may have to remove all manual adjustments for the structures that exceed it.
<code>structure.gfs.manualAdjustments.maxAdjustmentsPerAction</code>	200	The maximum number of manual adjustments per one user action. If this limit is exceeded, the action will be aborted without making any changes.

## Hidden Issue Links

Property	Default	Explanation
<code>structure.feature.hiddenLinks.enabled</code>	false	Set to true to enable support for hidden issue links.

## Index Consistency Checks

Property	Default	Explanation
<code>structure.indexConsistencyChecker.disabled</code>	false	Set to true to disable periodical checks of Lucene index consistency.

## Alternative initial values for project/type when creating an issue in dialog

Normally, when the user creates new issues through dialog, Structure remembers the selected project and issue type and offers those the next time by default. This dark feature enables a different algorithm, which used to work in a previous version of Structure: the initial project and issue type are taken from the issue that was focused when "+Create" or "+Next Issue" was pressed.

System property	structure.feature.altInitialValuesInDialog
Options to add in setenv.sh / setenv.bat	-Dstructure.feature.altInitialValuesInDialog=true
Internal feature name	altInitialValuesInDialog
Introduced in version	2.11.0

## Synchronizers

[Synchronization](#) lets you keep Structure issue hierarchy in sync with some other issue properties. This dark feature is disabled by default.

Property	Default	Explanation
structure.feature.synchronizers.enabled	false	Set to true to enable Synchronizers within Structure.

## Synchronizer Cycle Guard

The [cycle guard](#) is a component that detects conflicting synchronizers and prevents them from cycling forever, overriding each other's changes. The table below describes the system properties that control the cycle guard.

Property	Default	Explanation
structure.sync.guard.disable	false	Set to true to disable the cycle guard. Conflicting synchronizers will not be prevented from running forever. <b>Not recommended.</b>
structure.sync.guard.maxAutosyncsWithoutUserChanges	10	The maximum number of times that a synchronizer is allowed to run, processing the changes generated by another synchronizer. If this limit is exceeded, the two synchronizers are considered to be in conflict.
structure.sync.guard.stop.disable	false	If true, conflicting synchronizers will not be disabled automatically. The cycling may repeat after a user-generated change.
structure.sync.guard.email.owner.disable	false	If true, the cycle guard will never send e-mail notifications to synchronizer owners.
structure.sync.guard.email.admin.disable	false	If true, the cycle guard will never send e-mail notifications to Jira administrators.
structure.sync.guard.email.admin.cycles	10	<p>The minimum number of times a cycle must be detected for a synchronizer before an e-mail notification about that synchronizer is sent to Jira administrators.</p> <p>The counter is reset when a synchronizer is automatically disabled, so if this number is greater than 1 and automatic disabling is on, the administrators will not be notified.</p>

## Resolved icon(green tick)

Property	Default	Explanation
structure.doneAttribute.byResolution	false	<p>false - the "Resolved icon" shown when the Resolution field of an issue is non-empty</p> <p>true - the "Resolved icon" is shown when the issue is in a Done status category (StatusCategory.COMPLETE)</p>

## Time in Status - Refresh Period

Property	Default	Explanation
----------	---------	-------------

<code>structure. timeinstatus. refreshPeriod</code>	3600000	<p>Sets update period for Time in Status columns. Value is in milliseconds.</p> <p>Time is Status is updated any time a status change occurs in Jira or once per hour if status remains unchanged. This option allows you to update the Time in Status more or less often, when issues remain in the same status.</p>
---	---------	---