

Advanced Configuration with System Properties

Certain advanced aspects of Structure's behavior might not have dedicated configuration pages, being controlled by system properties instead. This page lists Structure-related system properties and describes how to set them.

Setting System Properties on Startup

You can set a system property 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

If you don't want to restart JIRA, you may use the free [Script Runner](#) add-on to set system properties.

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 immediately, but the properties will be reset to their default values when you restart JIRA. If you want the changes to be permanent, please use the `-D` startup option as described above.

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. Not recommended .
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>

Changing Synchronizer Order for Incremental Synchronization

When multiple synchronizers are installed on a structure, it may be possible that the outcome of the incremental synchronization depends on the order in which the synchronizers are activated. In the normal configuration, the order is not important – the result will be the same regardless of which synchronizer runs first. However, in some configurations, the result may vary.

The following option allows the administrator to control which synchronizers are run first.

Property	Default	Explanation
<code>structure.sync.order</code>	<i>not set</i>	Could be set to <code>true</code> or to a list of criteria that defines the order of synchronizer execution.

For the changes of the property to take effect, Structure should be disabled/enabled. If ordering is enabled, a warning is written to the logs with the words "*structure synchronizer ordering turned on*".

Specification of the order is a list of conditions, delimited by comma. Each condition could be:

- a number, in which case it is the synchronizer ID;
- one of the following: *links*, *agile*, *subtasks*, *filter*, *rollup*;
- module key for a custom synchronizer.

The order in which a synchronizer is run is defined by the earliest matching condition in this list. Those synchronizers that don't match any conditions run last.

Examples of property values:

- `links` — run links synchronization first, then everything else
- `subtasks,agile` — run subtasks sync, then agile sync, then everything else
- `123,filter,agile,456` — run synchronizer #123 first, then all filter synchronizers, then all agile synchronizers, then synchronizer #456. Note that if synchronizer #456 is a filter synchronizer, it will be run along with other filter synchronizers before agile.
- `agile,my.company.plugin:smart-sync` — run agile synchronizers, then custom synchronizers with key `my.company.plugin:smart-sync`, then everything else
- `true` — use default values (cannot be combined with any conditions)

The default order, which is selected by `true` property value, is this: `links,agile,subtasks,filter,rollup`.

The order is applied to all synchronizers working on all structures. That is, with the default order, first all links synchronizers on all structures are executed, then all synchronizers on all agile structures are executed, and so on.

The order has no effect when the synchronizers are not invoked together, for example, when a synchronizer reacts to the change from another synchronizer.