

Enterprise Deployment

Performance

Structure.Gantt is a fairly complex product that works on top of Jira and Structure platforms. It is also a very flexible product, just as Structure app is, so a lot of choices can be made by the user – like how much data is in one Gantt Chart and where this data comes from.

Therefore, the performance impact of using Structure.Gantt depends on a number of factors:

- The number of issues in the structure. (The single most important factor.)
- Using Jira's calculated fields (such as Script Runner fields, but *not* Structure's formulas) as the source of attributes, for example, Start Date. (This is rarely the case, but it may have massive performance impact.)
- Structure.Gantt CPU and memory footprint and its optimizations.
- Structure and Jira performance.
- Lucene index performance.
- Database performance (simple data retrieval queries).

Performance impact from a single Gantt Chart does not depend on the total number of issues in JIRA, total number of Gantt Charts, or total number of users. The amount of consumed memory will depend on the number of concurrent users working with Structure.Gantt and with Structure, as the products maintain caches of per-user data.

Performance Target: 10,000 issues

As of version 1.0, we are targeting Structure.Gantt to work well on configurations with 10,000 issues in one structure / Gantt Chart. Such configurations show reasonable amount of CPU and memory impact, as well as good responsiveness for the end-users.

We are also testing Structure.Gantt on structures with 100,000 issues. At the moment, we do not recommend to have such large configurations as the recalculation of the Gantt Chart may take up to 30 seconds, during which Structure.Gantt extracts a lot of information from Jira database and Lucene index.

Testing for Potential Impact

When testing Structure.Gantt for potential performance impact, watch out for the following indicators.

- Memory consumption, which may be influenced by the Gantt configuration.
- Frequency of Lucene index queries.
- Frequency of database retrieval queries.

Security and Data Access

When developing Structure and Structure.Gantt, we follow the best practices and respect the settings that are configured in Jira.

- Jira permissions are respected. All changes that are made through Structure.Gantt are executed in the context of some Jira user account, and with respect to that user's permissions. For example, if a user does not have permissions to create issue links, that action will not be possible through Structure.Gantt as well.
- Jira issue security and browsing permissions are respected. If a user does not have the permissions to see a particular issue, they will not be able to see it in Structure.Gantt too.
- Application security is continuously checked to make sure that our products do not introduce XSS or other vulnerabilities to Jira.

Caching of Issue Access

For the sake of performance, Structure and Structure.Gantt cache the result of access check (whether a particular user has access to a particular issue). The cache is invalidated if the issue is changed and every 5 minutes. The cached result can potentially be used when some other factors has changed and the access has changed as well.

This means that if a user *gains* access to an issue through any means other than setting the issue's Security Level, Structure apps might still hide that issue from the user for up to 5 minutes. The same happens if the user *loses* access – Structure apps may still show that issue's data for up to 5 minutes.

Limiting Access to Gantt Charts

Structure.Gantt works on top of Structure app. In particular, a Gantt chart is created based on a structure. Therefore, by limiting who has access to Structure app and who can create structures, a Jira admin can control the exposure of the users to Structure apps.

See [Gradual Deployment](#) in the Structure documentation for more details on how to gradually introduce the app to the system.