



SLA Rules Reference

How SLA Pulse for Jira measures your SLAs

PulseRook | Document version 2.0 | October 2026

In short. SLA Pulse for Jira calculates its own SLA outcomes from each issue's creation time, resolution time, issue type and priority. It does **not** read Jira Service Management's built-in SLA goals, so its numbers are independent of them and can differ. This document explains every rule so you can decide whether it fits your process.

1. How SLA targets are set

Each SLA target is a number of hours, set by **issue type** and then by **priority**. Targets are set **per project**. A Jira site administrator sets a **site default** that every project follows. A project administrator can give their own project custom rules, which **replace** the site default for that project only and never affect other projects. Issue types without their own rules use the Default rules.

Built-in default targets (before anyone changes anything):

Priority	Default SLA	Near breach starts after
Highest	12 hours	9 hours
High	24 hours	18 hours
Medium	36 hours	27 hours
Low	48 hours	36 hours
Lowest	72 hours	54 hours
Any other priority	60 hours	45 hours

2. Who can change what

Setting	Who can change it
Site default rules and site default working hours	Jira site administrators
A project's own rules, working hours and pause statuses	That project's administrators, and Jira site administrators
Everyone else	Can view the rules that apply to the project, read-only

These rules are enforced by the app on the server, using Jira permissions, so a project administrator cannot change another project or the site default.

3. When the clock starts and stops

- **Starts** when the issue is created.
- **Stops** when the issue is done: its Resolution is set, or its status is in the Jira "Done" status category. If there is no resolution date, the time of the move to Done is used.
- **Keeps running** while the issue is open. Assigning, commenting or moving between open statuses does not stop it, **unless** the project pauses the clock in that status (section 5).
- Time is counted in **continuous clock hours** (24/7) by default, or in **business hours** if the project has a working-hours calendar (section 4).

4. Working hours and holidays

A calendar makes the clock count only working time. It can be set for the whole site or for one project; a project calendar replaces the site calendar. The default is 24/7, so nothing changes until an administrator sets one.

- Choose a **time zone**, the **working days**, and the **start and end time** for each day (days can have different hours).
- Add **holidays** by date (up to 500). A holiday removes the whole day.
- Outside working time the clock does not run. A ticket created after closing starts counting at the next opening. A ticket resolved outside hours stops at the last closing time.
- The Settings page shows a live example, such as: a ticket created Friday 17:30 with a 4-hour SLA breaches Monday 12:30.
- **Changing a calendar recalculates every outcome** in the selected period, including finished issues, so compliance, MTTA, MTTR and alerts can change.
- Shifts that run overnight (for example 22:00 to 06:00) are not supported.

5. Pausing the clock

Each project can choose statuses that **pause** the clock, such as "Waiting for customer". Nothing is selected by default, and pause is set per project only because workflows differ.

- While an issue is in a pausing status, its clock does not run. Time spent there is excluded from SLA time, MTTA and MTTR.
- A paused issue keeps its On track or Near breach outcome and shows a **Paused** tag. It is not alerted on. If it had already breached, it stays Breached.
- An issue resolved while still in a pausing status stops at its resolution time.
- Pause uses each issue's status history. If the history cannot be loaded, the dashboard shows a warning that numbers may overstate breaches.

6. SLA outcome of each issue

Outcome	Meaning
Met	Done, and finished within the SLA hours.
Breached	Over the SLA hours: either still open and overdue, or done late. A late issue stays Breached after it is resolved. The decision uses exact time, not rounded hours.
Near breach	Open, not yet breached, and the remaining time is 25% of the SLA or less (never less than a 2-hour warning window).
On track	Open, not yet breached, and outside the warning window.

The four outcomes always add up to the total number of issues shown, and every count equals the number of rows in the list behind it. "Paused" is a tag, not a fifth outcome.

7. Compliance %

Compliance = Met / (Met + Breached)

- Only issues with a **decided** outcome count. Open issues that are On track or in Near breach have no outcome yet and are left out, so a new ticket never inflates the score.
- If no issue has been decided yet, compliance shows **N/A**, not 0% or 100%.
- The dashboard, reports, PDF export and Slack digest all use this same formula.

8. Which issues are included

- **All open issues** in the selected project, plus issues **finished within the selected period** (7, 30 or 90 days; 30 by default). Issues in a Closed status are included if they were resolved inside the period.
- The app reads up to the **4,000 newest** issues. If a project has more, a banner on the dashboard says so.

9. MTTA and MTTR

Metric	How it is calculated
MTTA (mean time to acknowledge)	Average time from creation to the first public comment written by someone other than the reporter . Comments by apps or automation and Jira Service Management internal notes are ignored. Issues with no qualifying comment are left out of the average (they are not counted as zero).
MTTR (mean time to resolution)	Average time from creation to the end of the clock, for issues done within the selected period. The median is shown next to it, because a few very old issues can skew an average.

Both use the same time basis as the SLA itself: clock hours, or business hours and excluding paused time when the project uses those options. The dashboard labels which one.

10. Alerts

- Slack alerts and the scheduled daily digest report **open** issues that are in Near breach or Breached. Issues that are done or paused are not alerted on, although done issues still count in totals and compliance.
- The same issue is not re-alerted more than once in 24 hours.
- Alerts are posted only to the Slack webhook that your administrator configures. Alerts are optional.

11. Worked examples

Using the default Medium target of 36 hours, counted 24/7 (Near breach from hour 27):

Situation	Result
Open for 10 hours	On track
Open for 30 hours	Near breach (6 hours left)
Open for 40 hours	Breached
Resolved after 20 hours	Met
Resolved after 40 hours	Breached (late, stays Breached)
Assigned to someone, not resolved, open for 40 hours	Breached (assignment does not stop the clock)
Open 40 hours, of which 10 hours in a pausing status	Near breach (30 active hours counted)

Business hours: with Monday to Friday 09:00 to 18:00, a ticket created Friday 17:30 with a 4-hour SLA uses 30 minutes on Friday and 3.5 hours on Monday, so it breaches Monday 12:30.

Compliance: 6 Met, 2 Breached and 3 open issues give $6 / (6 + 2) = 75\%$. The 3 open issues are not in the calculation until they are decided.

12. What is supported today

Please check this list against your own SLA policy before you rely on the numbers.

Requirement	Supported?
Different targets by issue type and priority	Yes
Different rules for different projects on one site	Yes
Business-hours and holiday calendars	Yes. One calendar per project or site.
Pausing the clock for waiting statuses	Yes, per project
Clock stops when an issue is resolved or Done	Yes
Overnight shifts (for example 22:00 to 06:00)	No
Different calendars for different priorities in one project	No
Reading Jira Service Management's built-in SLA goals	No. SLA Pulse uses its own rules.
Separate first-response and resolution targets	No. Resolution target only; first response is shown as the MTTA metric.
More than one SLA target per issue	No

Differences from Jira Service Management. Because JSM sets goals per request type with its own conditions, the same ticket can show a different SLA status in JSM and in SLA Pulse. SLA Pulse is designed as a simple, transparent, rule-based view of timeliness. It is not a replacement for JSM's contractual SLA engine.

Questions about these rules: support@pulserook.com