

Formula Column

Formula column displays a value that is calculated from issue fields or other attributes using a custom formula. You can use one of the predefined formulas or write your own using a simple expression language.

Configuring Formula Column

Use the following steps to define a new formula.

1. Write Formula

The formula is a simple expression that uses variables (for example, "priority" or "duedate"), arithmetic operations and functions. Variables are linked to issue fields or other attributes and the formula is calculated for every item in the structure. The first thing you need to do is express exactly what you'd like to calculate, using the variables you will need.

See [Expr Language](#) for an in-depth description of the language and examples.

2. Verify Formula's Correctness

When you stop writing the formula and click **Save**, it is verified and a green mark is displayed if the formula is ready to be used. If it's not, the problematic parts are highlighted in the formula editor with red color.

There might be several problems with your formula:

- A syntax error happens when the formula cannot be parsed, for example, if you forget a closing parenthesis. The editor will highlight the part that failed to get parsed.
 - A function resolution error happens when you use an unknown function (probably you have mistyped a known function's name). These functions are shown in red.
 - A variable resolution error happens when you have used a variable but it hasn't been defined yet. This error is perfectly ok and you should proceed to the next step, defining the variables.

3. Define Variables

Normally, the formula would involve some variables. (Otherwise the result will be the same for each row in the structure.) These variables need to be mapped to *attributes*, which could be issue fields, progress, a hierarchical total or something else. The configuration dialog displays the list of all variables used in the formula.

If you use a well-know field name for a variable, such as **Priority** or **Assignee**, Structure will automatically assign it to the respective attribute. If the variable name is unfamiliar to Structure, it will remain unassigned and will be clearly marked with red color and icon.

[illegible]

To assign a variable to an attribute, click it in the variable list or in the formula. An attribute selector opens up, where you can select from the list of standard attributes, use a customizable attribute or pick an attribute that is already being used by some of the columns on the screen.

The following names are automatically recognized by Structure:

- Names of the standard JIRA fields, such as **Summary** or **Priority**.
- Names of the custom fields, with all non-letters removed and all spaces converted to underscores, for example, **Story_Points**.
- Names starting with **Total_** or with **Sum_** and having a well-known name afterwards – such as **Sum_Story_Points** or **Total_Estimate**. These are converted to a Sum attribute of the given value. (Without duplicate removal option.)

Even if Structure automatically assigns an attribute to a variable, you might want to double-check it by clicking the variable name.

Please note that if you're selecting an attribute defined in another column (from the "Used in Columns" section of the attribute selection drop-down), you **copy** the attribute definition from another column at that moment, rather than creating a link to that column. That means that even if the other column is removed or if its content is changed, the Formula configuration will keep using the attribute as it was defined at the moment you configured the variable.

You can use nested formula as a variable. Once you select **Formula...** a text area for nested formula is shown. Nested formula can have variables as well, and these also can be formulas - no nesting level limit here. Variables in nested formula are not ones declared by parent one, variables do not overwrite each other even if they have the same name.

Once you finished setting nested formula up you can collapse the dialog by clicking to a gear button near menu you selected attribute. You can return to editing later by clicking the same button; formula can be updated also after the dialog will be closed and reopened. Beware that if you choose existing formula (they can present in menus "Used in Columns" and "Recently Used") this formula is copied into editor (i.e. formulas in original columns will not be modified).

4. Optionally, Turn On Aggregation

Select **Sum Over Sub-Items** to have each row display a total value, calculated as a sum of the value for the row and values for all sub-items. The usual options to exclude values from duplicates (multiple rows with the same item) and to operate on a filtered structure apply.



Note that sometimes it doesn't make sense to calculate a sum. For example, percent values usually cannot be added together.

Structure has no knowledge about the meaning of the values, so it will offer to sum them up anyway. It's up to the user to make sure that the calculated value makes sense.



Note that it's possible to use [aggregate functions](#) instead.

5. Optionally, Select a Number Format

If the result of your formula is a number, you might want to customize the way it is displayed. The following options are available:

- **General** is the usual way numbers are displayed, as-is.
- **Number** format lets you specify the number of decimal places that will always be shown. The value will be rounded up to the least meaningful digit in this format.
- **Percentage** is similar to number, however, the value is treated as a ratio (0.0 = 0%, 1.0 = 100%) and the percent sign is added.
- **Date/Time** option is used to show dates and times and allows to pick a format.
- **Duration** option is used to display a duration values (such as those stored in **Time Spent** field) and format them nicely as number of days/hours /minutes.



Note that dates, times and durations are all numbers in the Expr language.

Duration is represented as the number of milliseconds. Dates are represented as "Epoch milliseconds", the number of milliseconds between midnight January 1st, 1970 (GMT) and the specified date, not counting leap seconds. Negative values are allowed to represent earlier dates.

5.1 About Work Time Option

If option **Work Time** is selected, Structure will use JIRA's time tracking settings to convert the number of hours to the number of days and weeks. By default, JIRA is set up with 8 hours in a day and 5 days in a week. If that option is not selected, hours are converted to days and weeks on a calendar basis.

Whether you need to use that option depends on where the value is coming from.

If you subtract dates (for example, if you want to calculate the number of days the ticket remains open: `now() - created`), then you'd probably want to keep Work Time option off and see calendar duration.

If you operate with the values retrieved from issue's Original Estimate, Remaining Estimate and Time Spent fields (for example, to calculate overspending: `time_spent + remaining_estimate - original_estimate`), then you'd probably use Work Time option.

6. Give Column a Meaningful Name

Structure will try to provide a name based on the formula, but in most cases you'd want to name it according to the meaning of the calculated value.

Sharing Formula Columns

Formula columns are just like any columns, so you can make them a part of a View or create a perspective URL that would open the same configuration, including the formula column.

Sorting by Calculated Value

You can sort by the value calculated in the column by clicking the column header.

See Also

- [Bundled Formulas](#)
- [Expr Language](#)