

Formulas Overview - Variables, Operators, & Functions

Last Modified on 01/23/2025 10:03 am EST

Overview

Formulas in Resolver are used to perform calculations resulting in a value. Formulas comprise numeric data pulled from the following system elements select lists, numeric and date fields, values from other formulas, and workflow states.



Note:

When a relationship or reference is selected as a formula variable, it cannot be deleted to prevent issues with formula calculations.

Related Information/Setup

Please see the following articles for related information and setup.

- [Creating a New Select List](#)
- [Creating a New Numeric Field](#)
- [Creating a Data & Time Field](#)
- [Formula Examples](#)

Formula Components

A Formula is made up of variables, operators, and functions.

Variable: A variable is the data unit used to represent a value (e.g., TOTALLOST = \$150,000, TOTALLOST = Variable and \$150,000 = Value). Variables represent values from a select list, numeric field, another formula, or workflow state and can be relationships, references, fields, and formulas attached to an object type.

Operator: Operators perform arithmetic actions (e.g. + for add or * for multiply) and can also group or compare variables. Operators are used in formulas that contain a function. Multiple operators can be added to a single formula.



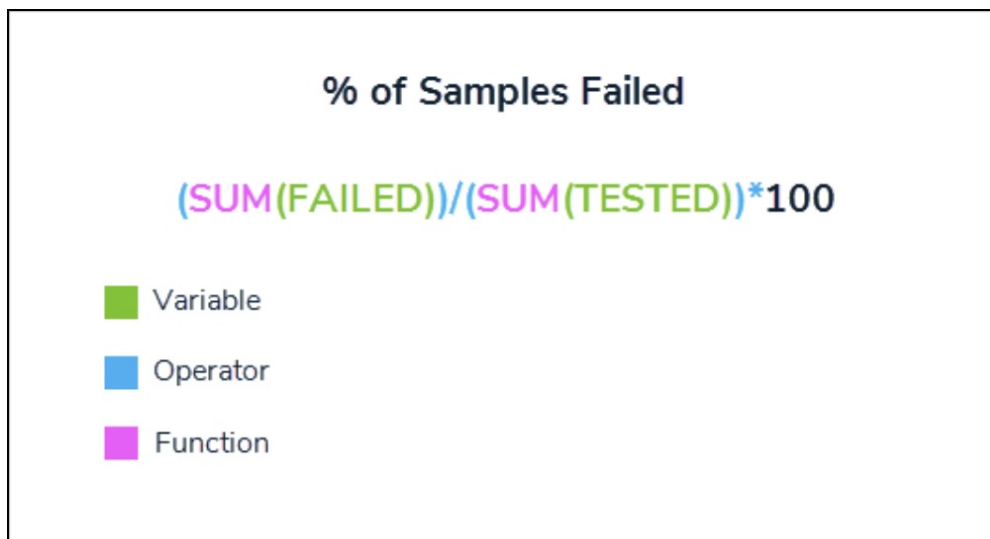
Formula Displaying Variables and an Operator

- Operators are used to perform basic calculations, compare, assign, or group data, or compare data to return a true or false (Boolean) result.
 - **Grouping ()**: Groups variables and operators using brackets () within a formula to create a single value (e.g., $2 * (3 + 4) = 14$).
 - **Add (+)**: Adds multiple variables, creating a single value (e.g., $5 + 4 = 9$).
 - **Subtract (-)**: Subtracts multiple variables, creating a single value (e.g., $5 - 4 = 1$).
 - **Multiply (*)**: Multiplies multiple variables, creating a single value (e.g., $5 * 4 = 20$).
 - **Divide (/)**: Divides multiple variables, creating a single value (e.g., $4/2 = 2$).
 - **Equal To (==)**: Checks if one value matches the value entered following the == symbol. If so, the result is **true** (e.g., $x == 4$, If X has a value of 4, the result is true).
 - **Unequal (!=)**: Checks if one value of two variables differs. If they're different, the result is **true** (e.g., $2 != 3$).
 - **Conditional Expression (?:)**: Checks if one value of a variable is **true** or **false**. If **true**, the result is the value entered after the (?) symbol. If **false**, the result is the value entered after the (:) symbol (e.g., $15 > 100 ? 1 : -1 = -1$).
 - **Less Than (<)**: Checks if one variable is less than another. If yes, the result is **true** (e.g., $2 < 3$).
 - **Greater Than (>)**: Checks if one variable is greater than another. If yes, the result is **true** (e.g., $3 > 2$).
 - **Less Than or Equal To (<=)**: Checks if one variable is less than or equal to another. If yes, the result is **true** (e.g., $2 <= 3$).
 - **Greater Than or Equal To (>=)**: Checks if one variable is greater than or equal to another. If yes, the result is **true** (e.g., $2 + 4 >= 6$).
 - **Logical And (and)**: If both variables are **true**, the result is **true**. If either or both variables are **false**, the result is **false** (e.g., $X == 5 \text{ AND } Y == 5$).
 - **Logical Or (or)**: If either or both variables are **true**, the result is **true**. If one or both

variables are **false**, the result is **false** (e.g., $X == 4$ OR $Y ==$).

- **Access to Indexes of Arrays []:** Allow users to access indexes of arrays within a formula. If non-supported operators (e.g., math.js operators) are used, the formula will be validated, but no value will be returned (e.g.,
 - $100 - [(3 - 1) + (7 \times 8)] = 42$).

Function: A function is a block of code that performs calculations on multiple values (e.g., SUM(LOSS)). **Relationship** and **Reference** variables are used with functions as they have multiple values. Performing functions using fields or formulas added directly to an object type is possible.



Formula Displaying Functions

Functions perform a task or calculate a value. Because functions require multiple values to complete the calculation, **Relationship** and **Reference** variable types are typically used with functions; however, it's possible to perform functions using fields or formulas added directly to the object type.

- **Max:** Calculates the maximum value from a variable.
- **Mean:** Calculates the mean value from a variable. The sum of all numeric values in a variable is divided by the number of values in the set (e.g., $5 + 8 + 4 + 4 / 4 = 5.25$).
- **Min:** Calculates the lowest numeric value from a variable.
- **Sum:** Calculates the total of all numeric values from a variable.
- **timeDiff:** Compares the days, months, or seconds between variables. This function requires date-related data from two **Date & Time** fields or a **Date & Time** field and the **today()** function. See the [Time Functions](#) article for more information.
- **timeOffset:** Offsets a variable by several seconds, days, or months. This function requires a **Date & Time** field variable. This function returns the offset date in the Unix timestamp. See the [Time Functions](#) article for more information. Formulas can display dates using date formats. Existing formulas will need to be updated to date to display

dates using date formats instead on Unix code.

- **Today:** Returns the current date. This function does not accept any parameters (variables inside the parentheses). However, it can be used as a parameter inside the **timeDiff** function.

Variable Sub-Type: A variable sub-type specifies how the data from multiple objects is compiled, calculated, and displayed. You must select a Variable Sub-Type if you create a variable using a field, formula, or state from a relationship or reference object type.

The screenshot shows a configuration form for variables. The 'Variable Type' is set to 'Relationship', the 'Relationship' is 'Issues', and the 'Available Components' is 'Due Date'. The 'Sub Type' dropdown menu is open, displaying a list of options: Array (selected), Sum, Count, Average, Every, Max, and Min. The 'Name' field is filled with 'DUEDATE' and the 'Description' field is empty.

Sub-type Variable Dropdown Menu

The following is a description of each Sub-type Variable:

Field and formula variables represent the values pulled from numeric fields, date fields, formulas, or select lists, the data on which the formula calculations are performed.

- **Array:** Creates a set of values using the variable. To display these values in a formula, they must first be converted into a single value using a function (e.g., sum, min, max, or mean). This is the default Sub-type.
- **Sum:** Calculates a total from the variable's set of values and returns a single number. This subtype is not available for select list variables.
- **Count:** Returns the number of times a variable has been added to an object, thereby counting the number of objects in the relationship/reference.
- **Average:** Calculates an average number from the variable's set of values. This subtype

is not available for select list variables.

- **Every:** Checks if the variable contains a value on the objects in the relationship/reference. A 1 (representing a **true** result) is returned if all objects contain any value. If some or all objects are missing a value, a 0 (representing a **false** result) is returned. This subtype does not check for specific values and is typically used in a [workflow condition](#) to control if an object can move into the next state.
- **Max:** Calculates the highest number from the variable's set of values. This subtype is not available for select list variables.
- **Min:** Calculates the lowest number from the variable's set of values. This subtype is not available for select list variables.

Workflow state variables represent the current state of one or more objects in a **Relationship** or **Reference** variable type using states from the object type's workflow or related assessment workflows.

- **All:** Checks if all objects in the relationship or reference are in the state selected in the Available Components dropdown menu. If all the objects are in the specified state, a 1 (representing a **true** result) is returned. If some or all objects are not in the specified state, a 0 (representing a **false** result) is returned. This is the default sub-type for workflow state variables.
- **Any:** Checks if any objects in the relationship or reference are in the state selected in the Available Components dropdown menu. If some or all objects are in the specified state, a 1 (representing a **true** result) is returned. If none of the objects are in the specified state, a 0 (representing a **false** result) is returned.
- **Count:** Returns the number of objects in the relationship or reference currently in the state selected in the Available Components dropdown menu.