

Variable Formula Fields

The Variable Formula data type lets users create queries or formulas that include system variables, with or without operators, to produce a result. The result can provide a direct value or simply resolve to true or false. This result can be used by rules to insert values or determine whether a condition has been met that will result in some action. The field is displayed as an input box and Formula Help button.

Variable ?

`$contract_id.requester_id.manager_id`


Formula Help

Variable Formula Field

Placing conditions in a variable formula field is useful in a few ways:

- Variable formula fields can be reused in saved searches, actions, and other automation tools, without needing to enter the same conditions many times or update the same batch of conditions in many different places.
- While administrators are usually responsible for maintaining saved searches or If-Then-Else actions, power users can be given access to a variable formula field to maintain the conditions and make any necessary updates based on process or policy changes.
- It can be a helpful way to separate conditions into batches; for example, permanent conditions that always apply might be hard-coded into a saved search or If-Then-Else action, while you can use the variable formula field to store conditions that change periodically or are otherwise unrelated to the hard-coded conditions.

For some real-world examples, see the [Use Cases](#) section below.

 Working with variable formula fields requires you be familiar and comfortable with formulas in Agiloft. For more information about formulas generally, see [Formulas](#).

Constructing a Variable Formula or Query

You can create custom formulas or queries in standard syntax, making use of field variables and the operators shown below. Note that these operators do differ in some cases from operators that can be used in other kinds of formulas. Click Formula Help to access a list of field variables in linked tables that you can use to build variable chains.

Operator	Definition
==	Is equal to
!=	Is not equal to
>=	Is greater than or equal to
<=	Is less than or equal to

>	Is greater than
<	Is less than
	Or
&&	And
~=	Contains
%	Remainder
!	Is not



The Is Not operator (!) can be placed in front of another condition to negate it, and this is the only way to perform a search for "does not contain." For instance, `!($related_to~= "Contract")` will find records in which the Related To field does not contain "Contract."

Variable Formula Field Output Types

The output of a variable formula field can be one of three types:

- A numeric value



Example

Calculate a simple sum using integer fields in a linked record:

```
($linked_record_id.integer_2 + $linked_record_id.integer_1)
```

If integer_2 is 155 and integer_1 is 100, the result is:

255

- Text



Example

Display a string of text with field values inserted:

```
("I belong to " + $creator_login.teams)
```

Result is a text string that inputs the values from the Teams field in the user record linked by the Creator Login field, e.g.:

"I belong to Support Team,Company Team,Admin Team,Internal Implementation Team"

- True-False Logical expression

Example

Create a filter based on a linked user record:

```
($linked_record_id.department_main_contact == "John Barrows")
```

The result may be True or False, depending on the actual value in the Department Main Contact field.

It outputs True when the linked record Department Main Contact is John Barrows, otherwise it returns False.

Use Cases

The Variable Formula field stores a formula, variable chain, or both. A variable chain can be used to find and pull values from other records, populating specified fields with those values. Formulas can be used to find and evaluate records against the conditions defined by the formula. In practice, the field on its own is a passive element in a record, but it can be used in rules or actions, where it can be requested and evaluated, and used as a condition for the rule or action. Two primary examples of usage are in the Approval Template table and the Replacement Variables table of the out-of-the-box system.

Approval Template Table

The Variable Formula field is often used in the [Approval Template](#) table for conditional approvals. It works to evaluate records defined by the formula in the field and generate approvals. For example, the Variable Formula field might contain a formula like the following:

```
($latest_contract_id.jurisdiction=="USA") && ($latest_contract_id.contract_amount>"250000")
```

This formula evaluates the linked contract record found by `$latest_contract_id` and checks whether it satisfies two conditions:

- The Jurisdiction field is equal to "USA."
- The Contract Amount field is greater than "250000."

If the values in the Jurisdiction and Contract Amount fields satisfy the conditions, the result is true and the approval is generated. In this case, both conditions must be satisfied because the formula uses the `&&` operator. If the `|` operator was used, only one of the conditions would need to be satisfied for the approval to be generated.

Typically, when approvals are to be generated from a contract, the contract manager presses a Route for Approval button that causes a linked record action to push the Contract ID into the Approval templates that are linked to that contract type.

The Approval template is then updated with the Latest Contract ID, which triggers an If-Then-Else rule. The rule checks whether the conditions defined in the Variable Formula field (the Condition field) are true, and if so, runs another action to convert the template into an Approval record:

```
if (Latest Contract ID!=NULL and (Approval Usage='Required' or Approval Usage='Conditional' and Condition='TRUE')) {  
    Convert to Contract Approval  
}
```

if Latest Contract ID isn't null, Approval Usage is Required or Conditional, and Condition is True, run Convert to Contract Approval action

If the condition is false, no approval is generated.

Replacement Variables Table

The Variable Formula field is used in the Replacement Variables table to define a variable chain that pulls values from a linked field relationship for use in other tables. These values are typically used in template records, such as in the Approval Template table or Task Template table.

Replacement Variables

All Replacement Variables ▾

36 record(s) found, 2 page(s). [Click here](#) to count records again.

<input type="checkbox"/>	Edit	ID ↓	List Value	Related To
<input type="checkbox"/>		38	Date Submitted	Contracts
<input type="checkbox"/>		37	Date Signed	Contracts
<input type="checkbox"/>		35	Renewal Notification Date	Contracts
<input type="checkbox"/>		34	Contract End Date	Contracts

Replacement Variables table

Approval Templates and Task Templates pull in a link to this table so templates can be set up by selecting the replacement value from a drop-down list, such as Contract Owner. When the list value is selected, the variable field is pulled in at the same time.

When an approval or task is generated from a template, it pulls in the variable field from the template, and a creation rule copies the resolved value of the variable into the Approver or Assignee field.

```
if (Assign Approval Based On='Person from Contract') {  
    Set Approver Based on Variable  
}
```

Note that the variable can be arrived at through a chain of relationships, as in the following example:

```
$linked_record_id.requester_id.manager_name
```

This is starting from the contract (linked record ID), going to the requester record linked through requester ID, then finding the manager's name within the requester's record.

For more information on configuring the fields in a Replacement Variables record, see the [Replacement Variables Table](#) article.



For replacement variables, the Variable Formula field should use a variable chain that starts from the Approval or Task record—not from the Approval template or Task template!