Formula Building Basics

Last Modified on 08/20/2019 1:06 pm EDT

Already familiar with how formulas work and need a quick reference of what's available? Click here to skip to the Formulas Cheat Sheet for syntax and definitions for all available Salsify formulas.

Formula Basics

Use formulas to meet endpoint requirements by combining and transforming stored property values. For example, for *Brand Name*, you may store your brands with trademark symbols, in proper case: Acme™.

A retailer requires *Brand* to be in all capital letters and not include special characters. Instead of storing another version of your brand name, you can use a formula to meet the requirement.

Or you may have data delivered by your ERP with inconsistent formatting. You can use formulas in computed properties to clean up formatting to use in catalogs, internal teams, and in readiness reports.

Click here to learn more about Computed Properties.

In-app Formulas vs. Templated Exports

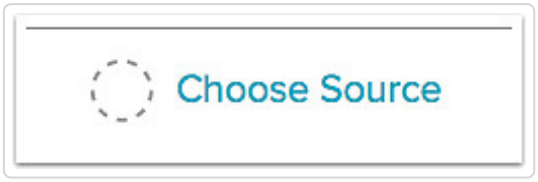
Use the SALSIFY_ prefix for all formulas in templated exports. In-app formulas (in-app computed properties and readiness reports) don't need the prefix. See the examples below for how to use the prefix.

Click here to learn more about templated exports.

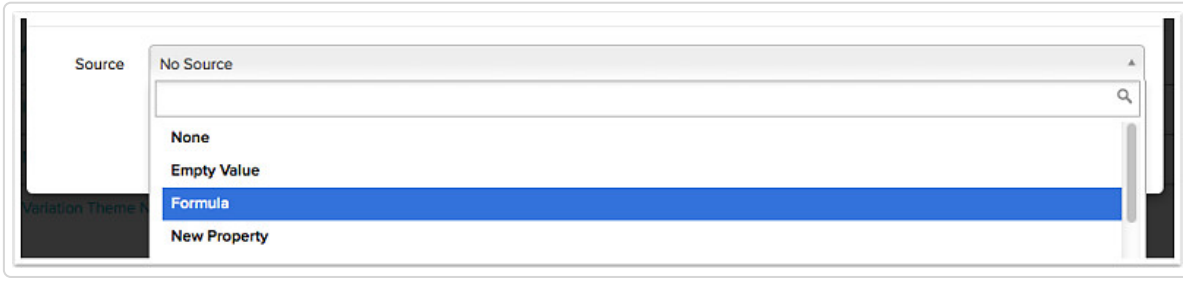
Using the Formula Builder

To use the formula builder in readiness reports:

1. Click *Choose Source* to the right of the attribute you want to fill with the formula.

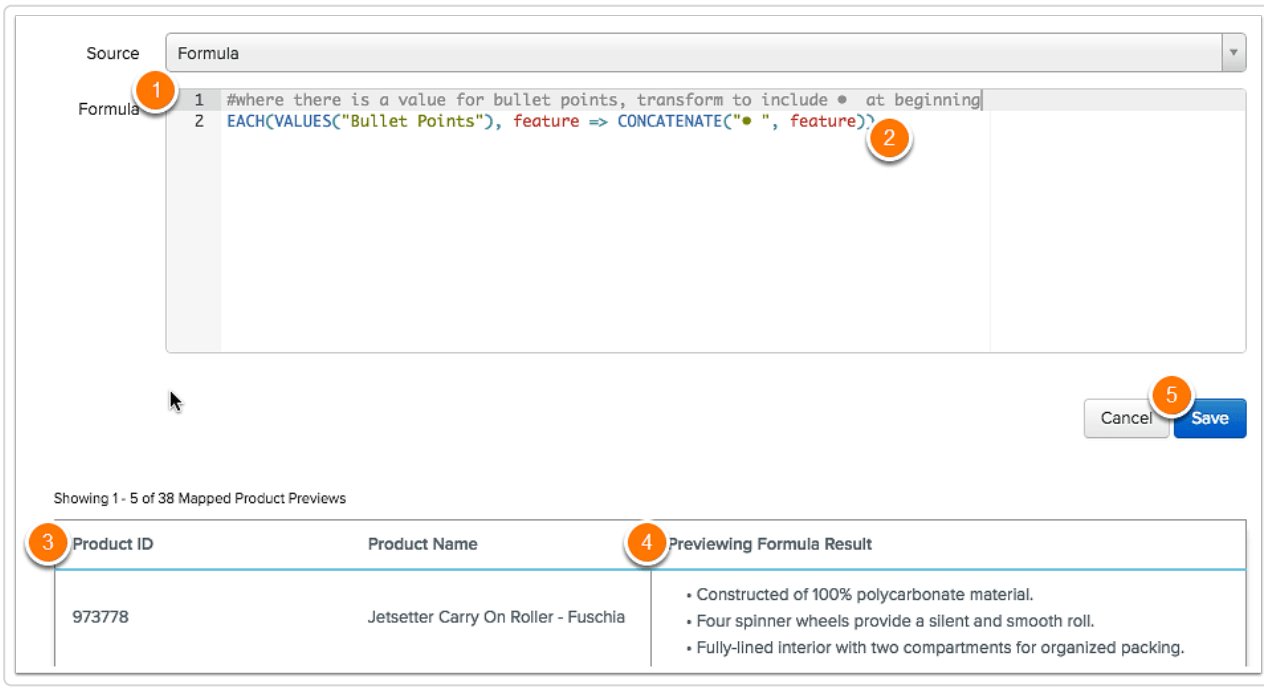


2. Select *Formula* from the *Source* dropdown menu.



3. Add the formula and any comments you want to include, and choose *Save*. If save is not available, troubleshoot your formula.

Formula Builder Overview

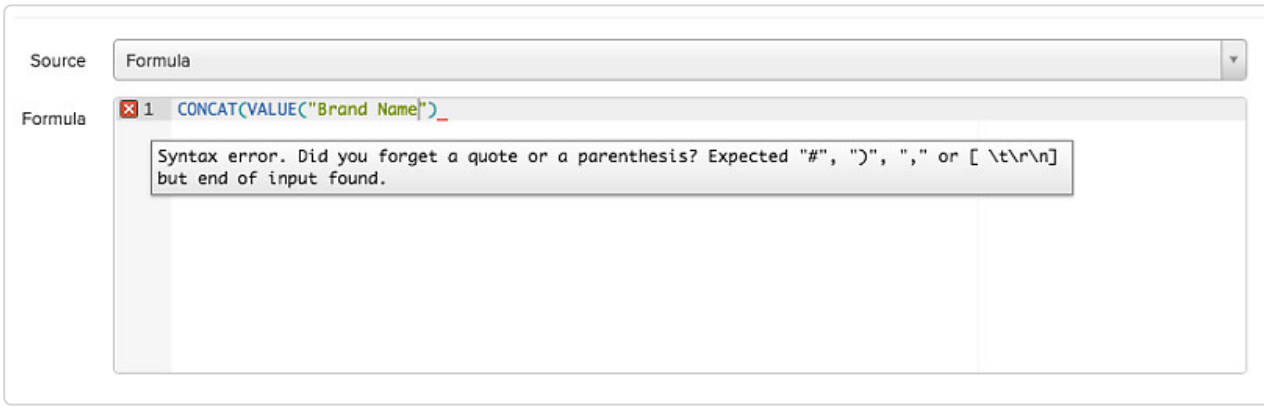


1. Use # anywhere you want to include notes or not process a line or part of a line the formula. Salsify will not try to process anything after the #.
    - Use at the beginning of a line to include a full-line comment.
    - Use to save (but not process) a formula line while testing and troubleshooting.
    - Add a # and comment after a formula line to explain what that line does.
2. Type or paste in your formula, either on a single line or multiple lines. The builder will try to validate the formula when it's complete.
3. A list of the first five products in your channel's product set.
4. Previews the result of the formula. Review to ensure the formula created the results you expect.

Adjust the formula until you do get expected results. As you adjust the formula, it will validate automatically.

5. Use *Next* to page through and check other products' formula results.
6. When you get the result you expect, click *Save*. *Cancel* will clear any changes you've made since the last time you saved the formula.

If you have errors in your formula, you'll see a red x next to its line number. Hover over the x to see the error message. In this example, the closing parenthesis is missing for `CONCAT`.



Visit the Formula Cheat Sheet for a full list of formulas, syntax, best practices and troubleshooting tips.

Formula Types

There are many formula types and formulas to use in combination to get the result you need. See the types of formulas in the list below, along with a few examples of how they work. Click the title to skip to the formula type you're interested in.

- Pulling Out Values
- Conditionals
- Combining & Transforming Values
- Working with Numbers
- Category Hierarchy Values
- Digital Asset URLs & Metadata
- Arrays & Advanced Formulas

Visit the Formula Cheat Sheet for the full list of formulas with their definitions and syntax.

If you're ever stuck and can't figure out a way to do what you need, click to contact us or talk to your Customer Champion. We're here to help!

Inserting a Set Value

The simplest version of a formula is setting a hard value for all products or product subset.

It's used for cases where you need to pass the same value for each product. In-app, insert that value with quotes wrapped around it. In a templated export, use just the phrase you want to insert (without quotes).

For example, if you need to fill in "1 year limited warranty" for each product, it would look like this:
In-app:

```
"1 year limited warranty"
```

Templated Export:

```
1 year limited warranty
```

It's useful where you need to fill in a Y or N answer for a retailer as well. Combine with conditionals to limit the answer only to a subset of products.

For example, the retailer requires an answer to the question "Batteries Included?" for one category of products, *R/C Cars*. You can write a formula to only apply the answer for that category:

In-app formula:

```
IF(VALUE("Category"),"R/C Cars","Y")
```

Templated export formula:

```
SALSIFY_IF(SALSIFY_VALUE("Category"), "R/C Cars", "Y")
```

The formula finds only the products with the value *R/C Cars* for *Category*, and applies `Y` for the requirement. All other products do not receive a value.

In a case where the answer is yes for one category, and no for all the others, we can use a variation of this formula:

In-app formula:

```
IF(VALUE("Category"), "R/C Cars", "Y","N")
```

Templated export formula:

```
SALSIFY_IF(VALUE("Category"), "R/C Cars", "Y", "N")
```

In this version, the formula checks for the category value; if it's *R/C Cars*, `Y` gets filled in. For all other products it fills in `N`.

Pulling Out Values

The building block for most formulas is the `VALUE` formula. This tells Salsify to look for and return the value that's being stored in a particular property. You can use any property to pull information from.

In readiness reports, the `VALUE` formula looks like this:

```
VALUE("Bullet Points")
```

`VALUE` is the formula function, what's inside the parenthesis is what that is acting on, and what's inside the quotation marks is the variable or value for the formula to generate a result from.

In this case, the formula says to Salsify, go find the value stored in the propertyID Bullet Points and return the value there. So, where your property is *Bullet Points* and you have Soft and Delicate stored in the first value, and Warm and Cozy in the second value, the formula will return Soft and Delicate.

This formula can also accept a position number, so where you have multiple values stored, the position tells Salsify which value to pull. For example, if you wanted the second value in *Bullet Points*, you'd use this formula:

```
VALUE("Bullet Points",2)
```

This formula would go to *Bullet Points*, look for the 2nd value stored and return it: Warm and Cozy

In cases where you want to return all the values stored in a property, use `JOIN_VALUES`. To get the values stored from our example above, we'd use:

```
JOIN_VALUES("Bullet Points",",")
```

or the formula can be written in a way that's easier to read as well:

```
JOIN_VALUES(
          "Bullet Points",
          ", "
)
```

Written in this style, it's easier to see where each parentheses pair and quote mark pair starts and ends.

For this example, `JOIN_VALUES` pulls all the values it finds in *Bullet Points*, and the second part of the formula inserts a delimiter between the values. In this case we want to insert a comma and space, so we wrap it in quotes to tell the formula to use it between the values. The formula's output is:

Soft and Delicate, Warm and Fuzzy

Using Variables

Variables provide a way of labeling data with a descriptive name, so formulas can be easier to understand. Using variables can also simplify otherwise very complex formulas.

Working with Numbers

These Salsify formulas do math with numeric values. They are often used to make conversions between standard and metric measurements, and in combination with conditionals to return values that meet certain criteria. Function examples include `ADD`, `SUBTRACT`, `ROUND`, `LENGTH`.

Combining and Transforming Values

These Salsify formula functions let you combine multiple values together and transform values. Examples include `CONCATENATE`, `COALESCE`, `LOWER`, `UPPER` and `PROPER`.

Conditionals

Conditionals, in their simplest form, test values or formulas. They are very similar to Excel formulas. IF, AND, OR, EQUAL and NOT are all examples of conditional formulas. They are often used in combination with other formulas to test and return a value based on a condition. For example:

```
IF(VALUE("Brand"),CONCATENATE(VALUE("Brand"),"™"))
```

In this example, we're checking to see if there is a value in Brand and if there is, we're adding a trademark symbol behind it. Without IF, the formula would have inserted a ™ symbol even where no brand name was filled in.

Note that for IF, the formula assumes that it will insert nothing if the "false" condition is not defined. In our formula we could have added an instruction for what to do if the condition is false, or Brand is empty. That would look like this:

```
IF(VALUE("Brand"),CONCATENATE(VALUE("Brand"),"™"),"Acme™")
```

Category Hierarchy Values

These formulas work with hierarchies, pulling out and transforming levels of the hierarchy.

Digital Asset URLs & Metadata

Use these formulas to insert and/or apply transformations in the Salsify URL and/or associated metadata for your digital assets attached to products. See Transforming Image Files for information on

transforming assets and delivering them in a download file.

Transforming Digital Asset Names

Digital assets can be renamed for delivery to endpoints in the naming convention they require and continue to be stored with your own naming convention. You can use any property or metadata information in the name, and like product content formulas, they can be combined to achieve the result you need.

Arrays & Advanced Formulas

Array and advanced formulas handle more complicated use cases. Use the Cheat Sheet for syntax and definitions.

For details on available formulas, best practices and troubleshooting tips, click here to visit the Formula Cheat Sheet .