Formula

This article has been generated from the online version of the documentation and might be out of date. Please, make sure to always refer to the online version of the documentation for the up-to-date information.

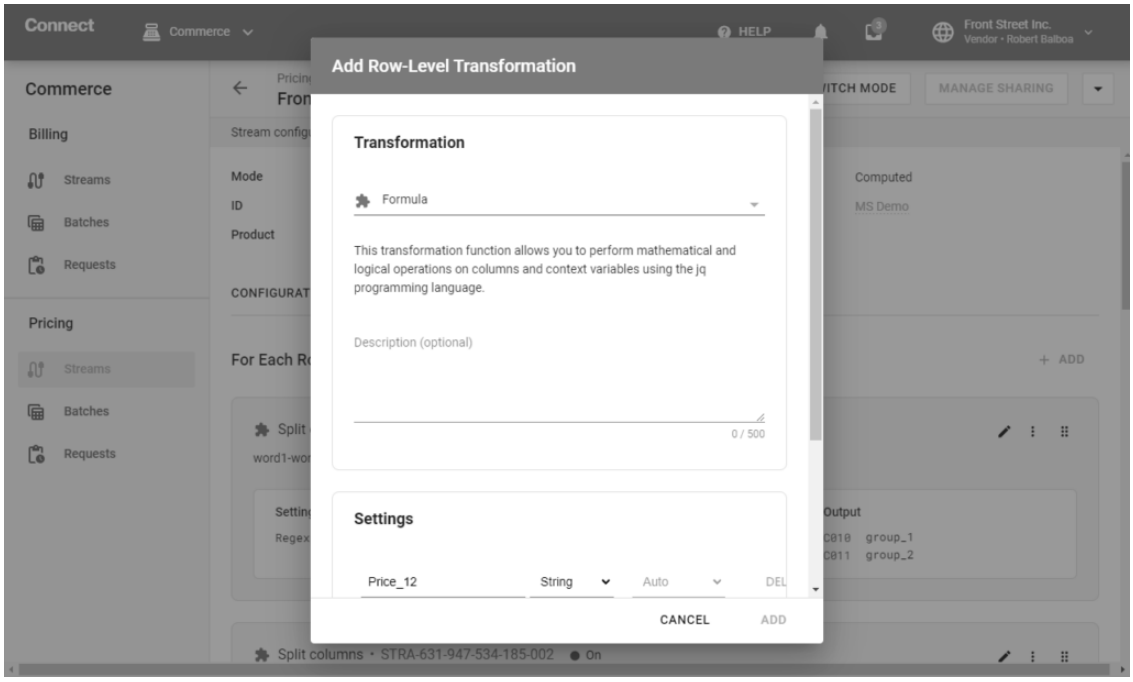
Auto-generated at May 10, 2024

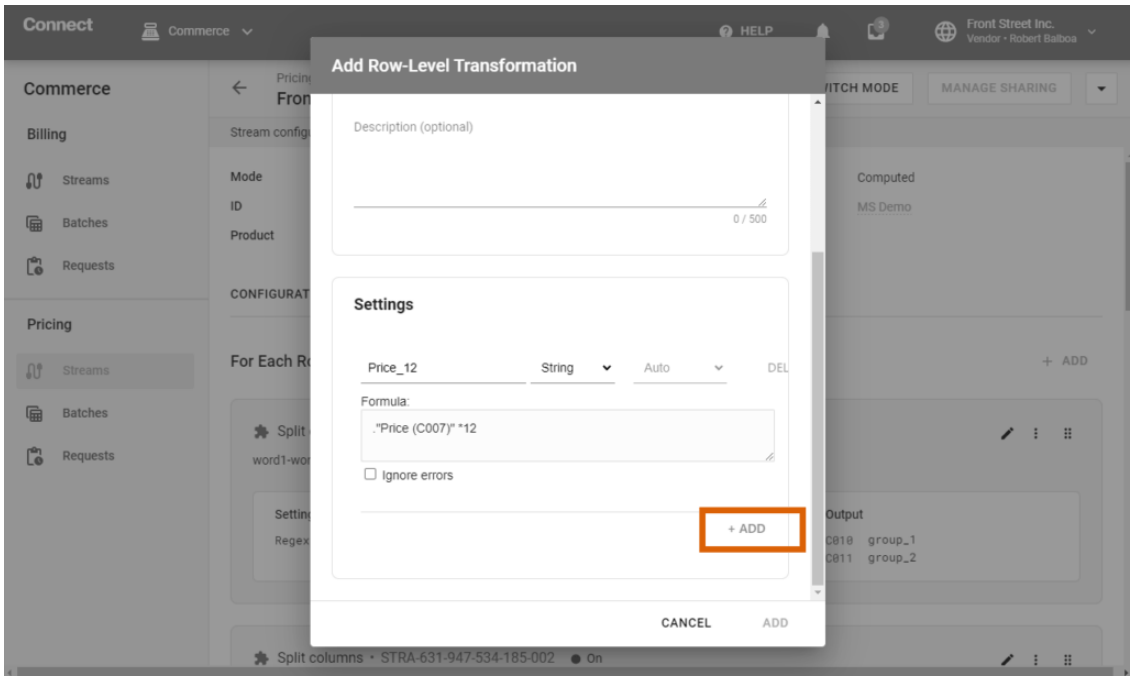
© Ingram Micro, Inc. All Rights Reserved.

Introduction

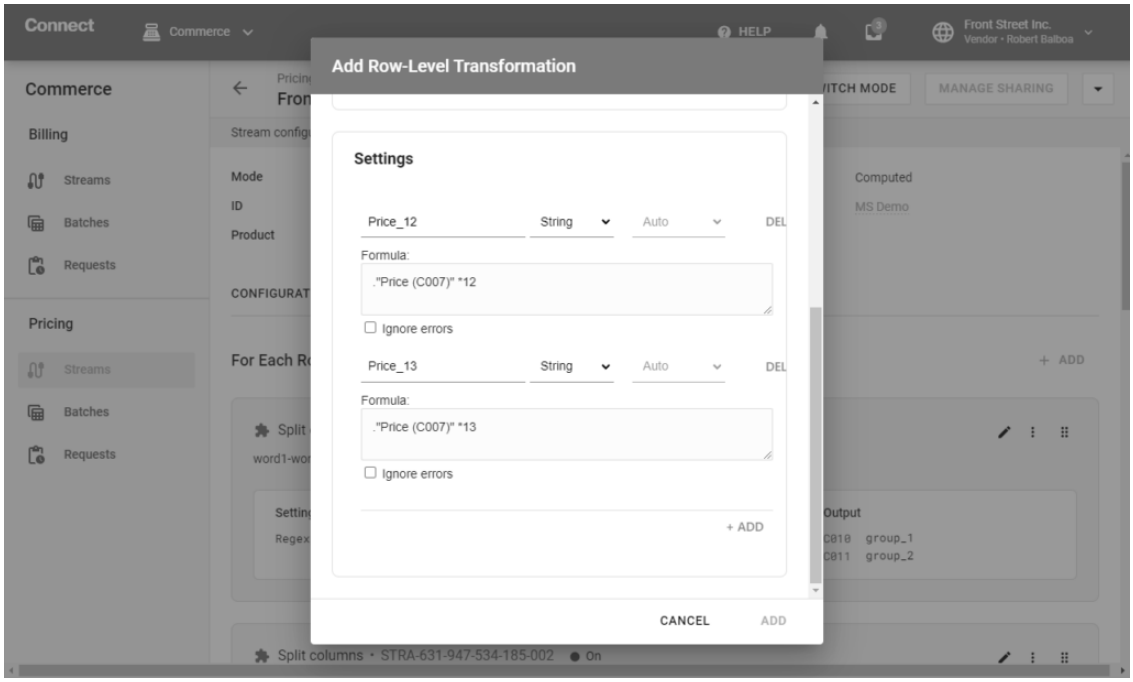
Formulas allow you to perform mathematical and logical operations on columns and context variables using the jq programming language.

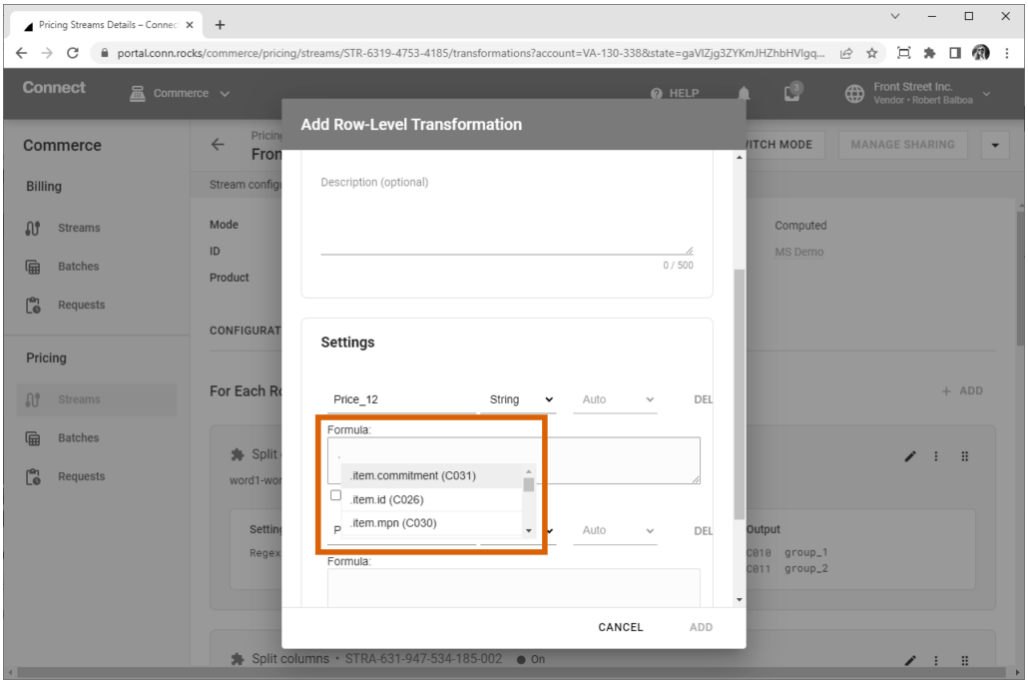
You can add several formulas to one transformation rule. For example, first, you may need to get price multiplied by 12 and then price multiplied by 13 – you can make this calculation using two different transformations or add both operations to one transformation.



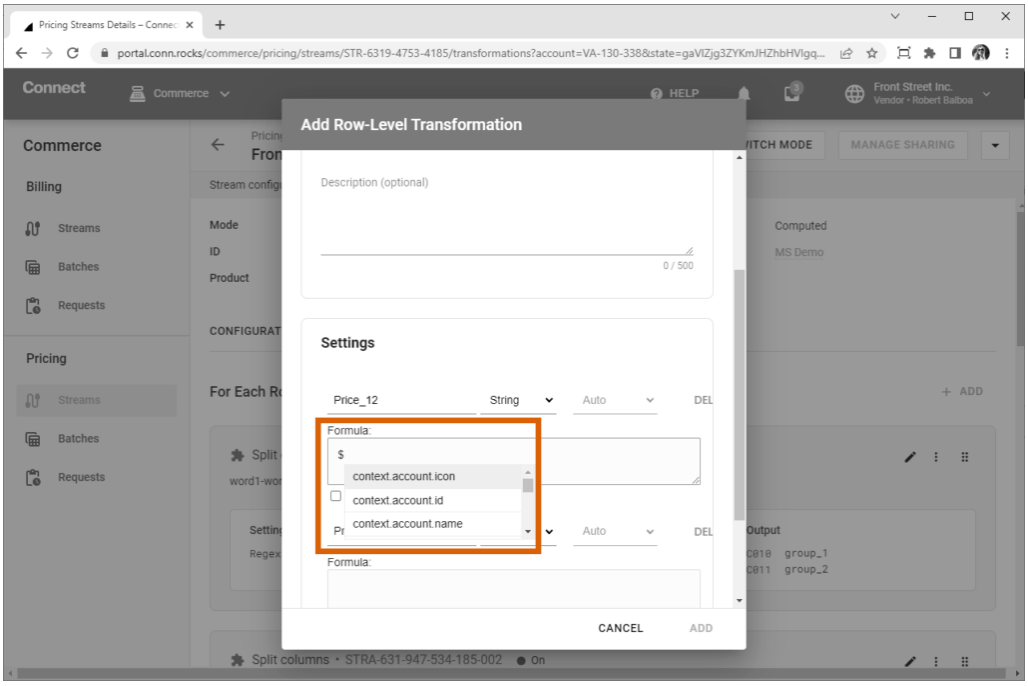


© Ingram Micro, Inc. All Rights Reserved.





© Ingram Micro, Inc. All Rights Reserved.



> **Information**
>
> When adding formula-based transformations, you can see the list of suggested columns by typing “.” and the list of context variables by typing the “$” sign.

Mathematical Operations

Some of the commonly used formulas are arithmetic operations like multiplication, division, exponentiation, rounding up, etc.

Multiplication

```
.Price * .Multiplicator
```

Division

```
.Price / .Multiplicator
```

Exponentiation

```
pow(.Price, 3)
```


© Ingram Micro, Inc. All Rights Reserved.

Complex Formulas

```
(.ListPrice + 10) * .Margin
```

Rounding

You can perform an operation of rounding the result up to a certain number of digits after the decimal point using the **round** function. For example, you can round up to a whole number or up to two digits after the decimal point:

```
round
round(2)
```

The result of these functions will be the following:

121.56311 | round -> 122

121.56311 | round(2) -> 121.57

Row Concatenation

You can combine (concatenate) data from multiple rows into one cell. For example:

```
$context.account.name + " " + $context.product.name
```

Type Conversion

You can change an expression from one data type to another. For example, you can convert a string to a number or vice versa:

```
.NumberString | tonumber
.Number | tostring
```

Operation Chains

You can perform various operations, one after another. For example, divide values, multiply, and then round up the result:

```
.ListPrice / .Cost * 100 | round(2)
```

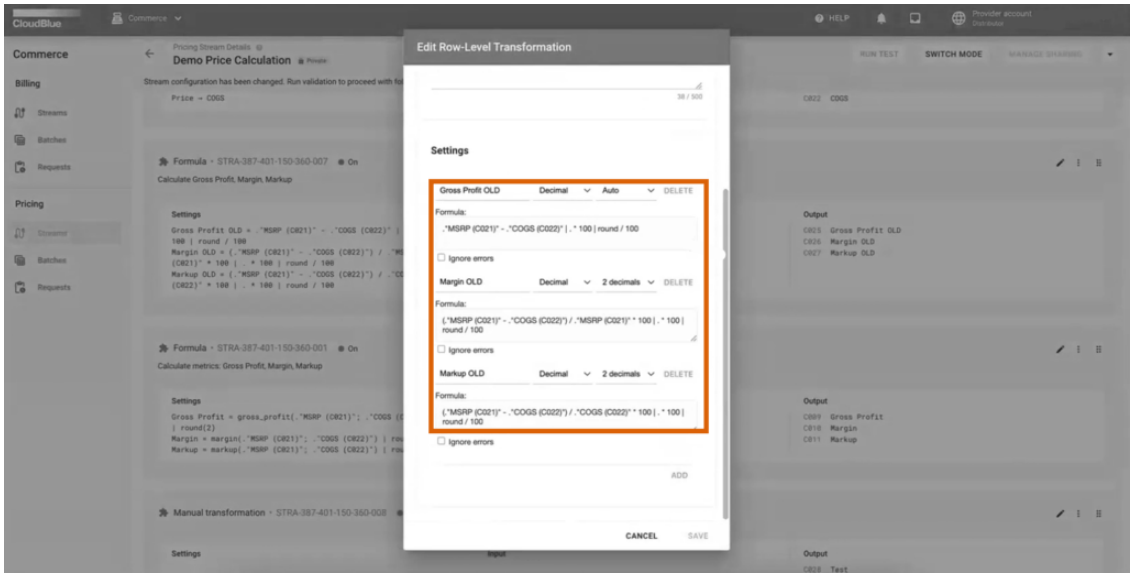

© Ingram Micro, Inc. All Rights Reserved.

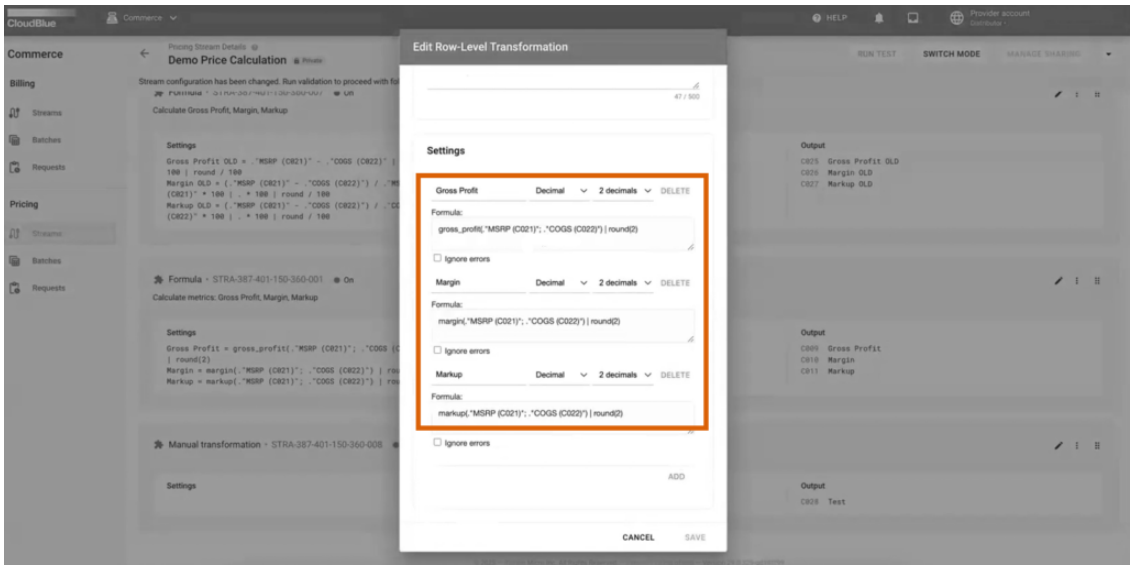
Predefined Formulas

You can use predefined formulas to calculate gross profit, margin, and markup.

```
gross_profit(.ListPrice, .COGS) = List Price / COGS * 100
margin(.ListPrice, .COGS) = (List Price - COGS) / List Price * 100
markup(.ListPrice, .COGS) = (List Price - COGS) / COGS * 100
```

Here are the examples of using standard jq formulas versus predefined formulas to calculate gross profit, margin, and markup rounded up to 2 digits after the decimal point:





© Ingram Micro, Inc. All Rights Reserved.