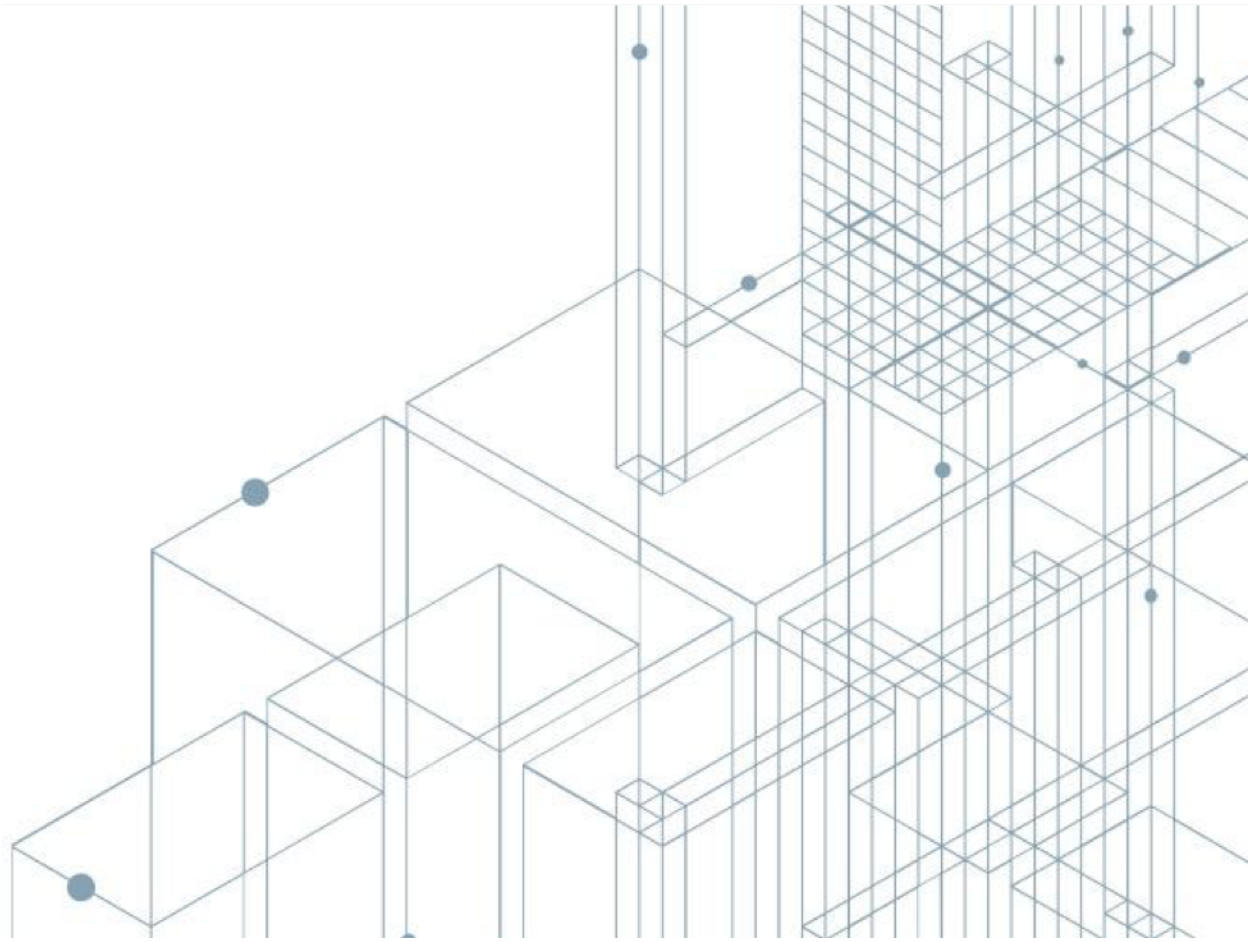
Transformation Rules Guide

Integrator.io



Revision summary

Version	Date	Revision class	Comments	Author
3.10.1	11/07/2017	major	Created guide	Rameez
4.0	12/03/2019	rewrite	New document	Schneider

Table of Contents

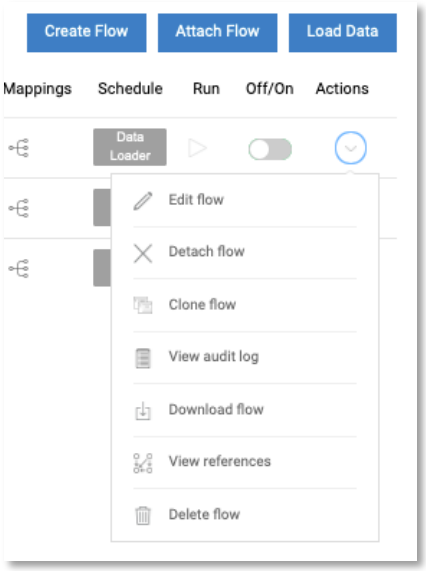
Welcome to integrator.io

This document describes how transformation rules are applied in the integrator.io environment. These rules will help you transform your record set or data for export or import using JavaScript Object Notation (JSON). The process is simple:

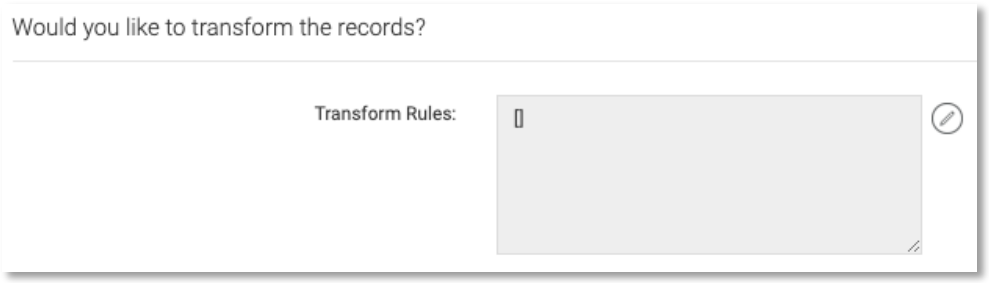
- Pre-Transformed Record
    - displays the raw data from the uploaded file and is the source file for the transformation.
- Transform Mappings
    - this is where you define the transformation rules which will map the Pre-Transform > Post-Transform data points.
- Post Transformed Record
    - displays the transformed record based on the mapping.

Export | Import Rules

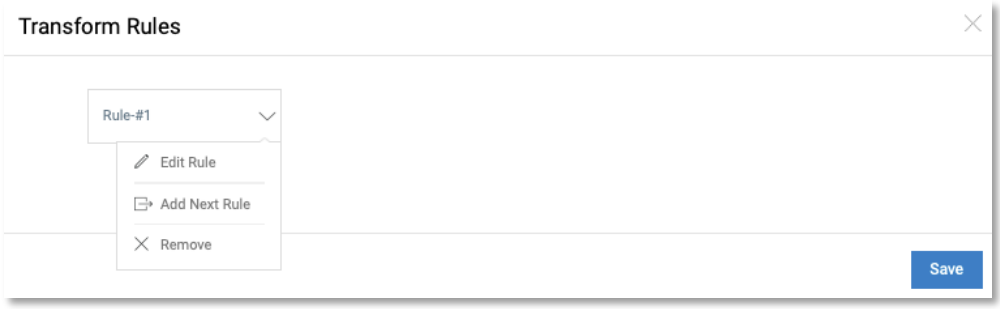
The integrator.io application allows modification of the format of your record in JSON format prior to export or import. From the Data Loader page, select "Create Flow", or the dropdown arrow next to the flow you want to transform, then click "Edit Flow".



To transform records, click the edit icon next to the Transform Rules field.



Select the rule to edit, or click [Rule #1] > [Edit Rule]

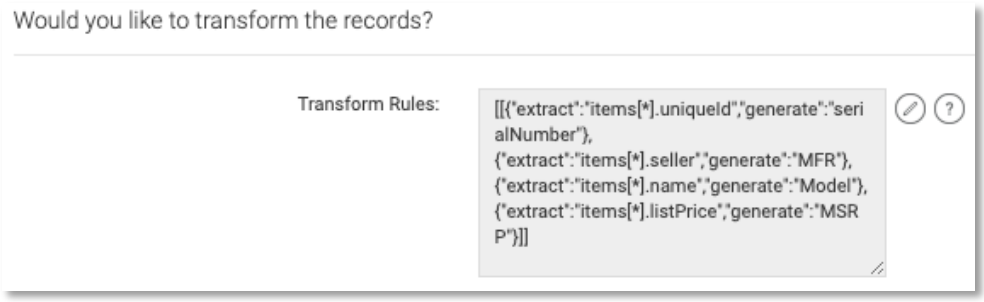


The following code sample shows the transformation process. Here, we have a record with some rather rigid database field names. These have been transformed into more industry-specific and business-friendly object names.

Pre-Transform	Transform Mappings	Post-Transform

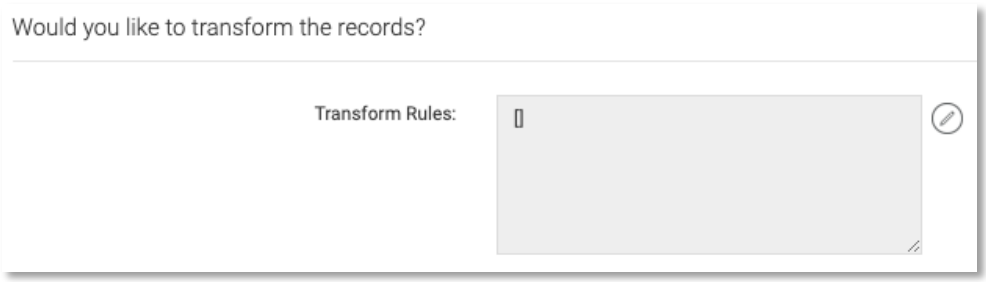
<pre>{
  "emailTracked":
"kawasaki@yahoo.com",
  "approvedBy": null,
  "items": [
    {
      "uniqueId": "000001",
      "seller": "Kawasaki",
      "name": "ZX-6R",
      "id": "ZX-6R",
      "productId": "11111",
      "refId": "100",
      "whsePrice": 8999,
      "listPrice": 9999,
      "ean": null,
      "lockId": "001",
      "quantity": 1
    }
  ]
}</pre>
	<table><tr><th>Extract</th><th>Generate</th></tr><tr><td>items[*].uniqueId</td><td>serialNumber</td></tr><tr><td>items[*].seller</td><td>MFR</td></tr><tr><td>items[*].name</td><td>Model</td></tr><tr><td>items[*].listPrice</td><td>MSRP</td></tr></table>	<pre>{
		  "serialNumber": "000001",
		  "MFR": "Kawasaki",
		  "Model": "ZX-6R",
		  "MSRP": 9999
		}</pre>

Click [Accept} and [Save]. The newly transformed record will now be displayed in the "Transform Rules" field on the main page.

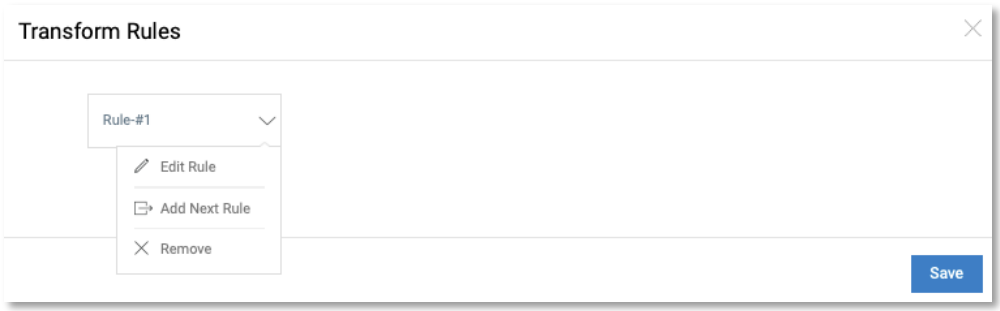


Create a new transform rule

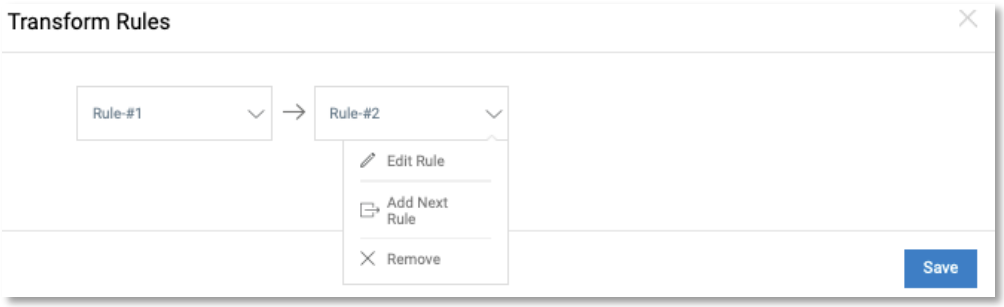
You can create multiple transformation rules which will first transform the existing record and apply a new set of rules. To do this, click the edit icon next to the Transform Rules field.



Click Rule #1 > Add Next Rule



The next transformation rule will populate to the right of the first.



Click "Edit rule" and perform the same steps to transform the field values from the example above to new values. In the example show below, the industry-specific and business-friendly object names have been transformed into dealer-specific database field names.

Pre-Transform	Transform Mappings	Post-Transform

<pre>{
  "serialNumber": "000002",
  "MFR": "Kawasaki",
  "Model": "ZX-2R",
  "MSRP": 4999
}</pre>
	Extract → Generate
	serialNumber → bin
	MFR → dealerSupplier
	Model → modelNumber
	MSRP → list
		<pre>{
		  "bin": "000002",
		  "dealerSupplier": "Kawasaki",
		  "modelNumber": "ZX-2R",
		  "list": 4999
		}</pre>

Merging Fields

Fields may be merged also, creating a list of values from the arrays in the record. In the example below, the items 1,2,3 and the prices 10, 20, 30 have been transformed into a merged record of items and their relative prices.

Pre-Transform	Transform Mappings	Post-Transform

<pre>{
  "itemsNames": ["item1", "item2", "item3"],
  "prices": [10, 20, 30]
}</pre>
	Extract → Generate
	itemsNames[*] → items[*].name
	prices[*] → items[*].price
		<pre>{
		  "items": [
		  {
		  "name": "item1",
		  "price": 10
		  },
		  {
		  "name": "item2",
		  "price": 20
		  },
		  {
		  "name": "item3",
		  "price": 30
		  }
		  ]
		}</pre>

Single to Multiple Object Transformation

Multiple objects may be combined into a single record by repeating the parent node automatically.

Pre-Transform

```
{
  "id": 100,
  "products": [
      {
  "name": "product1",
  "price": "10"
      } ,
      {
  "name": "product2",
  "price":  "20"
      },
      {
  "name": "product3",
   "price":  "30"
      }
  ]
}
```

Transform Mappings

Extract	Generate
id	*.id
products[*].price	*.price
products[*].name	*.name

Post-Transform

```
[
  {
    "price": "10",
    "name": "product1",
    "id": 100
  },
  {
    "price": "20",
    "name": "product2",
    "id": 100
  },
  {
    "price": "30",
    "name": "product3",
    "id": 100
  }
]
```

Delete Unwanted Fields

The following example demonstrates how to remove unwanted fields from an object. Here, the age and address fields have been removed for the record.

Pre-Transform

```
{
  "name":  "Estrella",
  "age": 50,
  "Address": "123 Anywhere St."
 }
```

Transform Mappings

Extract	Generate
name	name

Post-Transform

```
{
  "name":  "Estrella"
 }
```

Array within an Array

This example shows how an array can be transformed within another array. Here, we have transformed "items" to "products".

Pre-Transform	Transform Mappings	Post-Transform


```
{
  "items": [{
     "name" :  "perfumes",
    "price": 50,
     "flavours":  [{
      "weight": 3.14,
      "scent": "Obsession",
      "color": "black"
     } ,  {
      "weight":  2.26 ,
      "scent":  "Cool Water",
      "color": "blue"
     }]
   }]
 }
```


Extract	Generate
items[*].name	products[*].name
items[*].price	products[*].price
items[*].flavours[*].weight	products[*].weight
items[*].flavours[*].scent	products[*].scent
items[*].flavours[*].color	products[*].color


```
{
  "products": [
    {
      "weight": 3.14,
      "scent": "Obsession",
      "color": "black",
      "name": "perfumes",
      "price": 50
    },
    {
      "weight": 2.26,
      "scent": "Cool Water",
      "color": "blue",
      "name": "perfumes",
      "price": 50
    }
  ]
}
```


End of document