Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

LAB- Working with Objects and Arrays

Create new mule project with name 01-dw-objects-arrays

Understanding Object evaluation and construction

Create a new Mule configuration file with name “objectconstruction.xml”.

Drag a “Transform Message “ component in to a new flow.

In the header section, create 3 variables as shown below:

```
var numbers= ["one","two",3]

var object1={
     one: 1,      two:2
     }

var object2={
     two:2,
     three:3

}
```

In body expression, apply the constructor curly braces around object1

{object1}

You should see an error.

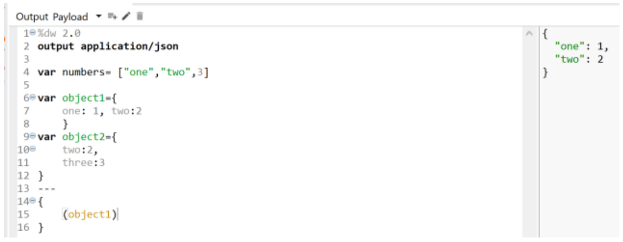
The evaluation parentheses had to be added so that the object constructor curly braces can extract all the key/value pairs from object1

Now change the expression inside the object constructor curly braces with evaluation paranthesis

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

You should observe the result as shown below:


Output Payload

```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
{
    (object1)
}
```

```
{
  "one": 1,
  "two": 2
}
```

Modify the expression as below and observe the result

```
{
        (object1), three:3
}
```

Now create a variable as shown below:

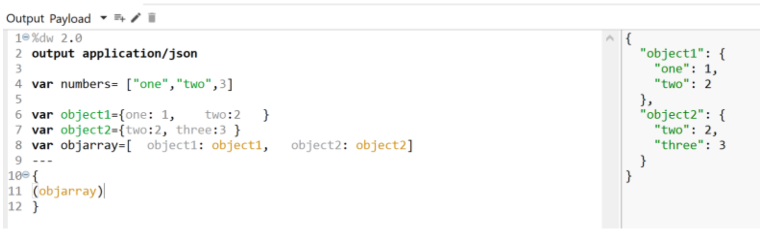
```
var objarray=[
        object1: object1,
        object2: object2
]
```

Now change the body as objearray and observe the preview

Then change the body as show below:

```
{
(objarray)
}
```

You should see that array is flattened in to an object


Output Payload

```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={one: 1,    two:2   }
var object2={two:2, three:3 }
var objarray=[  object1: object1,   object2: object2]
---
{
(objarray)
}
```

```
{
  "object1": {
    "one": 1,
    "two": 2
  },
  "object2": {
    "two": 2,
    "three": 3
  }
}
```

Now change the expression to

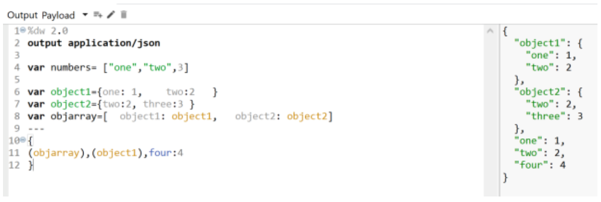
```
{
(objarray),(object1),four:4
```

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

}

Preview should look like below:



```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={one: 1,    two:2   }
var object2={two:2, three:3 }
var objarray=[  object1: object1,   object2: object2]
---
{
(objarray),(object1),four:4
}
```

```
{
  "object1": {
    "one": 1,
    "two": 2
  },
  "object2": {
    "two": 2,
    "three": 3
  },
  "one": 1,
  "two": 2,
  "four": 4
}
```

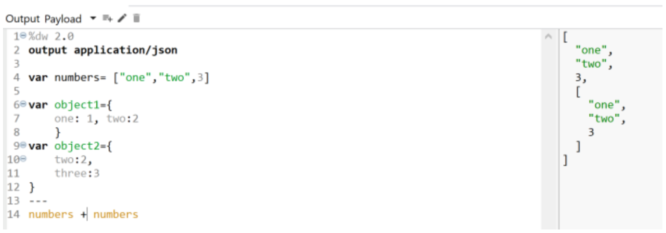
Understanding usage of + and ++ on Objects and Arrays

Now, let us see the difference between adding and concatenating 2 arrays.

In the body (after ---) of transform message component, write the below expression

numbers + numbers

You should observe the preview as shown below:



```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
numbers + numbers
```

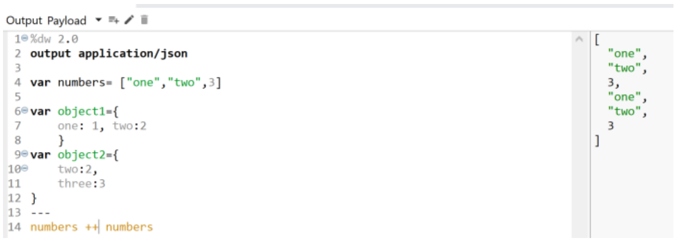
```
[
  "one",
  "two",
  3,
  [
    "one",
    "two",
    3
  ]
]
```

Now change the + operator to ++.

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

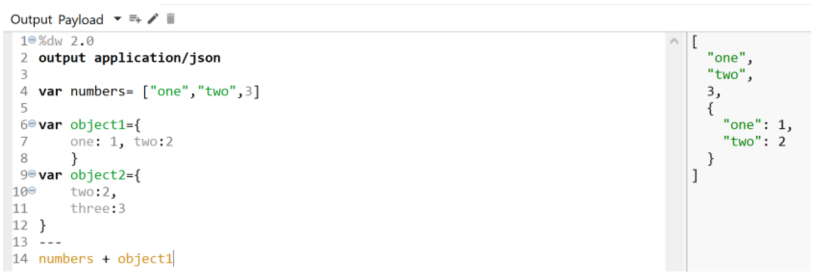
You should see the preview as shown below:



```
Output Payload
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
numbers ++ numbers
```

```
[
  "one",
  "two",
  3,
  "one",
  "two",
  3
]
```

Now change the expression to numbers + object1. You should observe that object1 is added as a fourth element as shown below:



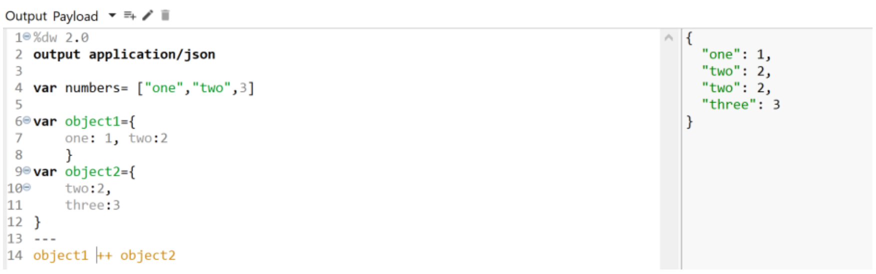
```
Output Payload
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
numbers + object1
```

```
[
  "one",
  "two",
  3,
  {
    "one": 1,
    "two": 2
  }
]
```

Now reverse the expression as object1+ numbers and observe that you get the error because arrays cannot be added to an object.

Now change the expression to numbers ++ object1 and observe that you get an error.

Now change the expression to Object1++object2 . You should observe the preview as shown below:



```
Output Payload
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
object1 ++ object2
```

```
{
  "one": 1,
  "two": 2,
  "two": 2,
  "three": 3
}
```

Now change the expression to object1+ object2 and observe that you get an error

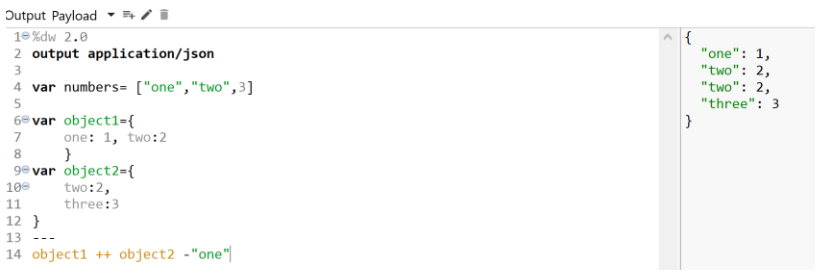
Try object1 ++ numbers and observe that u get an error

Now let us try to remove a key “one” from object1 ++ object2.
Change the expression to object1 ++ object2 – “one”

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

You will observe the below output which looks same as object1 ++ object2



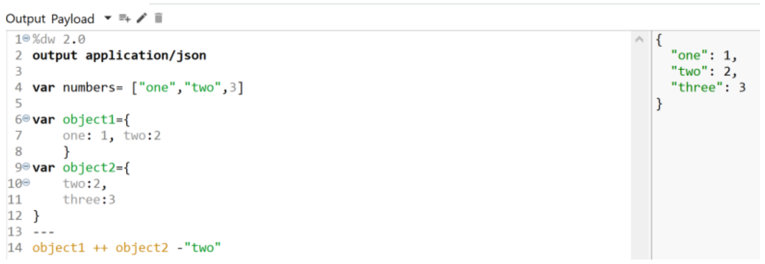
Output Payload

```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
object1 ++ object2 -"one"
```

```
{
  "one": 1,
  "two": 2,
  "two": 2,
  "three": 3
}
```

Actually, it will try to remove “one” only from the object2. As there is no key “one” in object2, there is no difference in the output. Actually, object1 ++ object2 –“one” is same as object1 ++ (object2-“one”)

Now try to remove “two” using expression object1 ++ object2 –“two” . You should observe the preview as shown below :

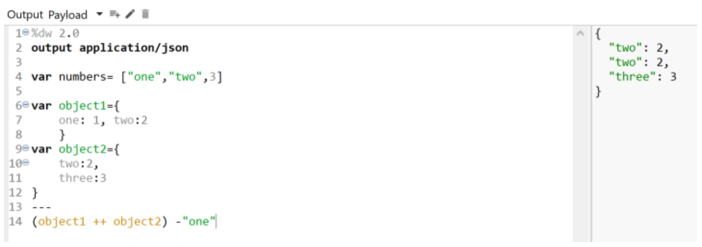


Output Payload

```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
object1 ++ object2 -"two"
```

```
{
  "one": 1,
  "two": 2,
  "three": 3
}
```

Now change the expression to (object1+object2)-“one”. You should observe the below preview:



Output Payload

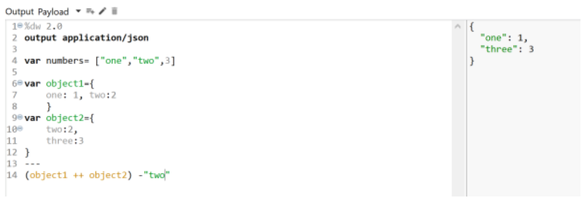
```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
(object1 ++ object2) -"one"
```

```
{
  "two": 2,
  "two": 2,
  "three": 3
}
```

Now change the expression to (object1 ++ object2)-“two” and observe the 2 key value pairs with key “two ” are removed as shown below.

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

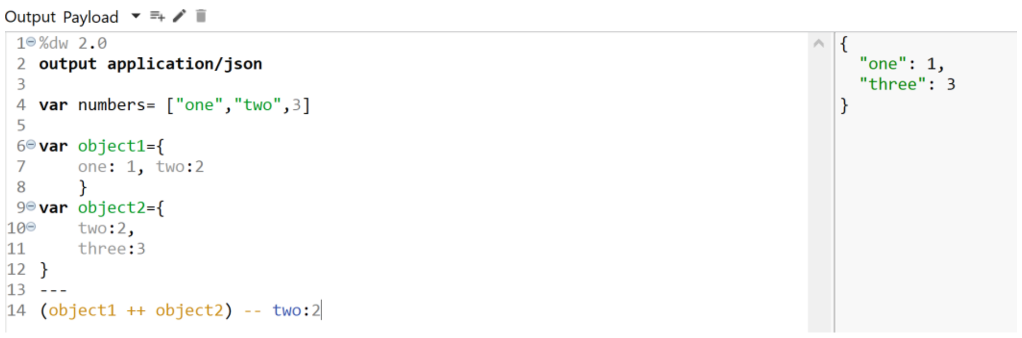

Output Payload

```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
(object1 ++ object2) -"two"
```

```
{
  "one": 1,
  "three": 3
}
```

Now change the expression to (object1 ++ object2) - -“two” . you should observe an error

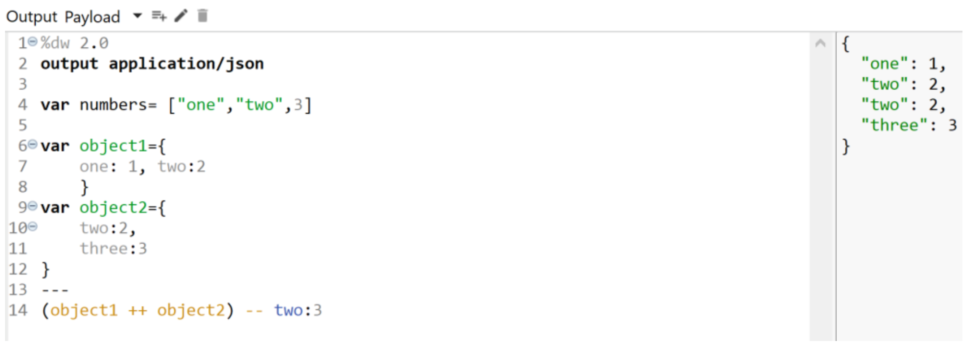
Now change the expression to (object1 ++ object2) - - two:2. You should see the preview as shown below :


Output Payload

```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
(object1 ++ object2) -- two:2
```

```
{
  "one": 1,
  "three": 3
}
```

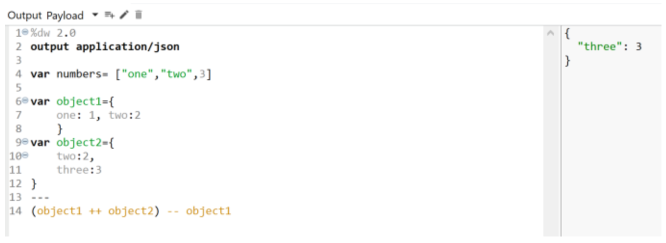
Now change the expression to (object1 ++ object2) - - two:3 . it should look like below:


Output Payload

```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
(object1 ++ object2) -- two:3
```

```
{
  "one": 1,
  "two": 2,
  "two": 2,
  "three": 3
}
```

Now change the expression to (object1 ++ object2) - - object1 and observe that all the matching key value pairs are removed as shown below:


Output Payload

```
%dw 2.0
output application/json

var numbers= ["one","two",3]

var object1={
    one: 1, two:2
    }
var object2={
    two:2,
    three:3
}
---
(object1 ++ object2) -- object1
```

```
{
  "three": 3
}
```

Congratulations on completing the Exercise