entrinsicon

Powerscripts in Action

JavaScript for Advanced Use Cases

Tom Hartnett

Implementation Specialist

1. Power Scripts & JSON Objects
2. Power Script Reserved Variables
3. Dynamic Pivot
4. Flush Flow Steps
5. Aggregate of Aggregates

Power Script Overview

- Uses the NodeJS library
- Normal JavaScript with a sandboxed script context
  Special reserved variables
- Does not return a value by default
- Can create and modify multiple columns in a single Flow Step

JSON Objects

- **A collection of key:value pairs of any data type**
- **Refer to each of the values with dot notation:**
- `person.firstName;`
- `person.job.title;`
- **Or refer to a value with bracket notation, which can accept a variable**:
- `person['lastName'];`
- `var personalDetail = 'age';`
- `person[personDetail];`

```
var person = {
        "firstName" : "Trainer",
        "lastName": "Smith",
        "age": 35,
        "job": {
                "company": "Entrinsik
Inc.",
                "title": "Senior Trainer",
                "responsibilities": [
                        "Run trainings",
                        "Develop trainings"
                ]
        }
}
```

Power Script Reserved Variables

entrinsicon

$record	object	The current row. PowerScript are free to modify the record.
$index	number	The current row number (row 1 = index 0)
$omit()	function	When called, removes the current row
$fields	object	Field metadata
$field	function	A function to describe a field: $field('location').label('Location')
$inputs	object	User inputs
$local	object	Whatever you want! Is not cleared from one record to the next
_	object / function	Lodash v3. Learn more at https://lodash.com
moment	function	Date manipulation and formatting: https://momentjs.com

Power Script Reserved Variables - $record

- **An Object that stores all the field aliases and values for the current record**
- **Access existing columns via alias**

```
var name = $record['firstName'] + ' ' +
    $record['lastName'];
```

- **Can replace existing values**

```
$record['firstName'] = "John";
```

- **Create new columns**

```
$record['helloWorld'] = "Hello World";

$record['testArray'] = [1,2,3];
```

Before:

```
$record = {
        "firstName" : "Bob",
        "lastName": "Smith"
}
```

After:

```
$record = {
        "firstName" : "John",
        "lastName": "Smith",
        "helloWorld": "Hello World",
        "testArray": [1,2,3]
}
```

Power Script Reserved Variables - $local

- **Creates local, internal variables**

```
$local['localVariable'] = "123";
var person = "jsmith";
$local[person] = {};
$local[person]['firstName'] = "John";
$local[person]['lastName'] = "Smith";
$local[person]['occupation'] = "Developer";
```

- **Persists across rows**
- **Important to initialize**

```
$local['xyz'] = $local['xyz'] || 0;
```

- Does not appear in output

Before:

```
$local = {};
```

After:

```
$local = {
    $local['localVariable'] = "123";
    "jsmith" : {
                "firstName": "John",
                "lastName": "Smith",
                "occupation":
"Developer"
        }
        "xyz":0
```

Dynamic Pivot

Before:

Order	Order Date	Product Name	Category Name
10249	7/5/1996	Tofu Manjimup Dried Apples	Produce Produce
10248	7/4/1996	Queso Cabrales Singaporean Hokkien Fried Mee Mozzarella di Giovanni	Dairy Products Grains/Cereals Dairy Products

After:

Order	Order Date	Produce	Dairy Products	Grains Cereals
10249	7/5/1996	Tofu Manjimup Dried Apples		
10248	7/4/1996		Queso Cabrales Mozzarella di Giovanni	Singaporean Hokkien Fried Mee

Power Script Reserved Variables - $local

- Using a for loop

```
var arrToPivot = $record['categoryName'];
var arrOfValues = $record['productName'];

for(var i in arrToPivot){
   var title = arrToPivot[i].replace(/\./g, '_');
   $record[title] = $record[title] || [];
   $record[title].push(arrOfValues[i]);
}
```

Using lodash

```
var arrToPivot = $record['categoryName'];
var arrOfValues = $record['productName'];

_.forEach(arrToPivot, (value, i) => {
        var title = _.snakeCase(value);
        $record[title] = $record[title] || [];
        $record[title].push(arrOfValues[i]);
});
```

Flush Flow Step

Flush

Finish all pending changes to records before moving onto the next Flow Step. Great for two-pass calculations.

REMOVE STEP

Flushing the flow

Once the flow reaches this step, **all the records will be processed** before continuing on to the next steps in the flow.

This Flow Step is useful when all records need to be used to obtain a value (i.e., a count or total amount) to be used in subsequent Flow Steps.

For instance, if you wanted to figure out the percentage a value had, you would:

1. Add a Power Script step that kept track of a total value.
2. Add a Flush step.
3. Add another Power Script step that divided the row value by the total value.

Dynamic Pivot

Before:

Order Date	Margaret Peacock	Janet Leverling	Nancy Davolio	Laura Callahan	Andrew Fuller	Sum Order Total
Q3 2021	20,234.50	8,317.40	14,909.40	15,459.80	5,940.80	84,437.50
Q4 2021	32,880.30	10,914.40	23,879.60	7,701.60	16,893.90	141,861.00
Q1 2022	44,795.20	29,658.60	15,330.10	19,271.60	7,639.30	147,879.90
Q2 2022	26,138.75	34,808.75	15,520.90	8,209.15	25,667.50	151,611.09
Q3 2022	31,019.22	10,586.45	33,578.48	11,632.00	19,999.75	165,179.64
Q4 2022	37,524.53	36,734.81	33,104.10	20,663.77	21,652.05	193,718.12
Q1 2023	41,189.74	67,731.94	45,146.48	33,084.45	45,101.41	315,242.12
Q2 2023	16,405.21	14,298.95	20,674.65	17,278.66	34,854.55	154,529.22

After:

	Margaret Peacock		Janet Leverling		Nancy Davolio		La	
Order Date	Sum Order Total	% of Q Sales	Sum Order Total	% of Q Sales	Sum Order Total	% of Q Sales		Sum Or...
Q3 2021	20,234.50	23.96	8,317.40	9.85	14,909.40	17.66		84,437.50
Q4 2021	32,880.30	23.18	10,914.40	7.69	23,879.60	16.83		141,861.00
Q1 2022	44,795.20	30.29	29,658.60	20.06	15,330.10	10.37		147,879.90
Q2 2022	26,138.75	17.24	34,808.75	22.96	15,520.90	10.24		151,611.09
Q3 2022	31,019.22	18.78	10,586.45	6.41	33,578.48	20.33		165,179.64
Q4 2022	37,524.53	19.37	36,734.81	18.96	33,104.10	17.09		193,718.12
Q1 2023	41,189.74	13.07	67,731.94	21.49	45,146.48	14.32		315,242.12
Q2 2023	16,405.21	10.62	14,298.95	9.25	20,674.65	13.38		154,529.22

Aggregate of Aggregates – Part One

entrinsicon

Power Script Flow Step #1

```
//The following code will execute for every record in your result set
//Create variables for all unique groupings and values
var quarterGroup = moment($record['OrderDate']).format('[Q]Q YYYY');
var salespersonGroup = $record['salesperson']
var value = $record['orderTotal'];


//Create properties in $local to store all yours totals with conditional checks to make sure we don't to reset them back to 0.
$local[quarterGroup] = $local[quarterGroup] || {};
$local[quarterGroup]['total'] = $local[quarterGroup]['total'] || 0
$local[quarterGroup][salespersonGroup] = $local[quarterGroup][salespersonGroup] || {};
$local[quarterGroup][salespersonGroup]['total'] = $local[quarterGroup][salespersonGroup]['total'] || 0;


//Increment the totals for each group
$local[quarterGroup]['total'] += value;
$local[quarterGroup][salespersonGroup]['total'] += value;
```

Aggregate of Aggregates – Part Two

entrinsicon

Flush Flow Step

(Very important!)

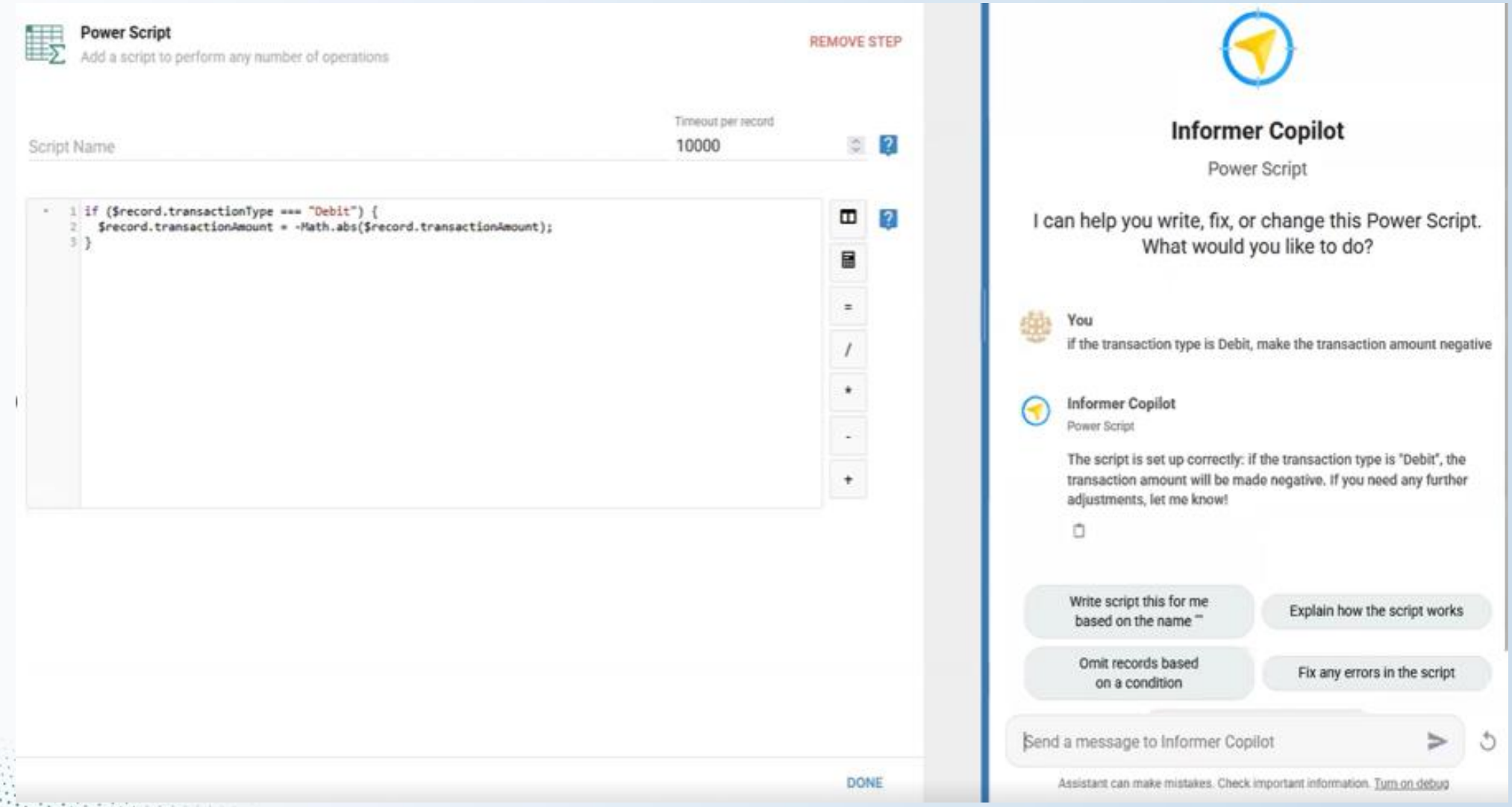
Power Script Flow Step #2

```
//The following code will execute for every record in your result set
//Create variables for all unique
var quarterGroup = moment($record['OrderDate']).format('[Q]Q YYYY');
var salespersonGroup = $record['salesperson']

//Output the desired aggregations into new fields
$record['quarterlyTotal'] = $local[quarterGroup]['total'];
$record['quarterlyTotalBySalesperson'] =
$local[quarterGroup][salespersonGroup]['total'];
$record['percentOfQuarterlySales'] = ($record['quarterlyTotalBySalesperson'] /
$record['quarterlyTotal']) * 100;
```

Using AI to Accelerate Powerscripting

entrinsicon



Thank You!

Tom Hartnett

thartnett@entrinsik.com