

PARIS ALTERYX USER GROUP MEETING

2019

5-06-2019





Meetup 2

WITH FANTASTIC SPEAKERS:



ALEXANDRU BADIU
Data Manager B2B



VINCENT PALASSE
BUSINESS INTELLIGENCE MANAGER



CYPRIEN DESCHEEMAEKER
CHIEF TECHNOLOGY OFFICER

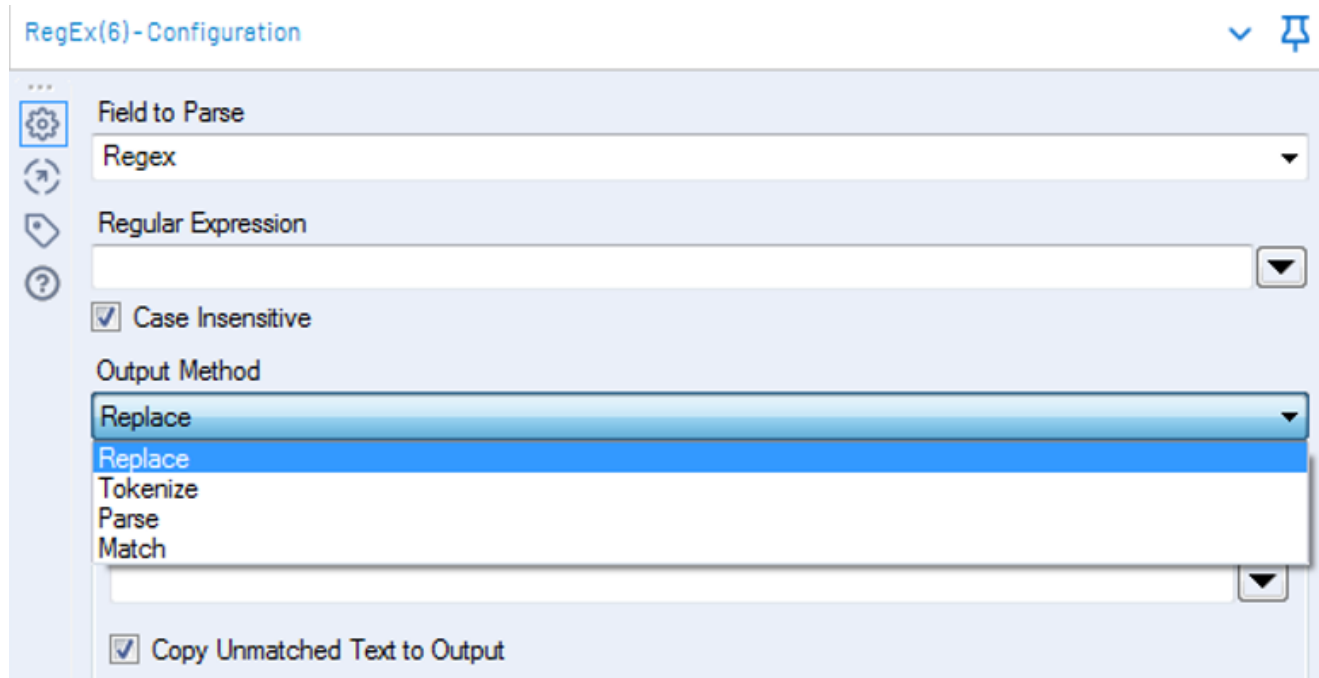


ARTHUR LADWEIN
CHEF DE PROJET DATA

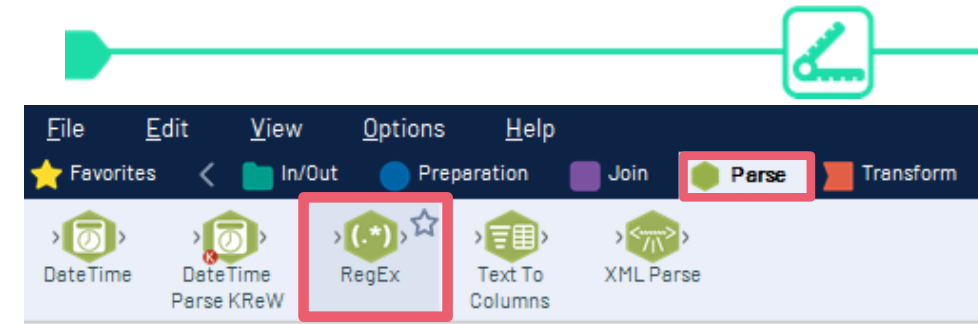
AGE- NDA

- **Introduction to RegEx**
- **Tips and tricks**
- **Weekly Challenge**
- **Business Use case**

REGEX Tool



```
REGEX_CountMatches(String, pattern, ica=1)  
REGEX_Match(String, pattern, ica=1)  
REGEX_Replace(String, pattern, replace, ica=1)
```



- Select the column containing data to parse
- Create a regular expression
 - Use library of symbology
 - Manual entry
- Choose an output method
 - Replace
 - Tokenize
 - Parse
 - Match
- Depending on selected output method, configuration will change

Arthur.



Alexandru Badiu

WARMING UP

LEARNING SPEAKING REGEX
WHY AND WHEN USING
REGEX?



alteryx
Public Information

The Thrill
of Solving





REGEX – MATCHING EXAMPLE

Regular Expression

\d

Emails	Time Match
Tom, don't forget our 30 minutes meeting at 4:00 tomorrow	False
Please be sure to send 1 follow-up to me tonight -BossWoman	False



REGEX – MATCHING EXAMPLE

Regular Expression

`.\+\d`

Emails	Time Match
Tom, don't forget our 30 minutes meeting at 4:00 tomorrow	False
Please be sure to send 1 follow-up to me tonight -BossWoman	False



REGEX – MATCHING EXAMPLE

Regular Expression

```
.+\\d.+
```

Emails	Time Match
Tom, don't forget our 30 minutes meeting at 4:00 tomorrow	True
Please be sure to send 1 follow-up to me tonight -BossWoman	True



REGEX – MATCHING EXAMPLE

Regular Expression

`.\+\d:.\+`

Emails	Time Match
Tom, don't forget our 30 minutes meeting at 4:00 tomorrow	True
Please be sure to send 1 follow-up to me tonight -BossWoman	False



REGEX – MATCHING EXAMPLE

- Assign a value of 1 or 0 if the expression did (not) match a string
 - 1 = TRUE
 - 0 = FALSE
- Case sensitive
- Option: Error if not matched
- Specify Boolean output column name

The screenshot shows the 'RegEx(3) - Configuration' tool in Alteryx. The 'Field to Parse' is set to 'Regex'. The 'Regular Expression' is '(d\d)-(d+)-(d+)'. The 'Case Insensitive' checkbox is checked. The 'Output Method' is set to 'Match'. In the 'Properties' section, the 'Match Status Field Name' is 'Regex_Matched' and the 'Error if not Matched' checkbox is unchecked.

Use case:

1. Check Email Address
2. Valid Phone number
3. Valid Credit Card number



REGEX – PARSING EXAMPLE

Regular Expression

000

Date

10-12-2015

Results

RegExOut1

RegExOut2

RegExOut3



REGEX – PARSING EXAMPLE

Regular Expression

```
(\d\d)-(\d{2})-(\d+)
```

Date

10-12-2015

Results

RegExOut1	RegExOut2	RegExOut3
10	12	2015



REGEX – PARSING EXAMPLE

- Using an expression, separate a single column of data into multiple fields
- Allows flexibility to change output by:
 - Rename fields
 - Changing fields data type
 - Changing data type size
- Specify fields by using « () »

RegEx(3) - Configuration

Field to Parse: Regex

Regular Expression: (d\d)-(d+)-(d+)

Case Insensitive

Output Method: Parse

Properties

Output Fields

	Output Field	Type	Size	Expression
1	RegExOut1	V_WString	254	(\d\d)
2	RegExOut2	V_WString	254	(d+)
3	RegExOut3	V_WString	254	(d+)

Use case:

1. allows you to *extract* useful information out of a block of text
2. Ex: Extract email domain names



REGEX – REPLACE EXAMPLE

Regular Expression

```
(\d+)-(\d+)-(\d+)
```

Date

```
10-12-2015
```

Replacement Text

```
$3-$1-$2
```

Results

Date

```
2015-10-12
```

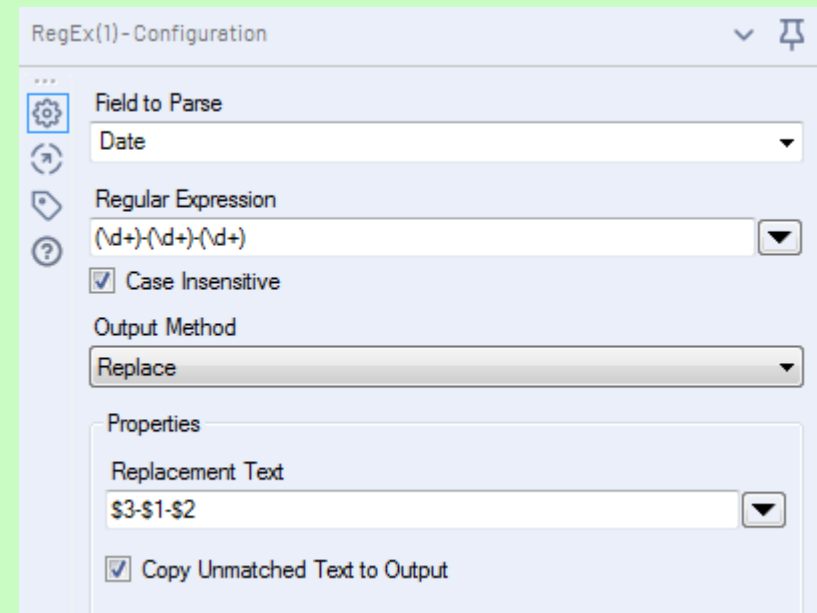


REGEX – REPLACE EXAMPLE

- Using an expression, re-order fields of data
- Specify fields by using « () »
- Use \$ symbol to re-order in the Replace Tab

Use case:

1. Replace part of a string
2. Reorder information inside a string





REGEX – TOKENIZE EXAMPLE

Regular Expression

```
\d\d\d-\d\d\d-\d\d\d\d
```

Phones
555-3--343343-720-501-292936-234-234234234-34-4505-340-4336509345

Results

1	2	3
720-501-2929	505-340-4336	Null

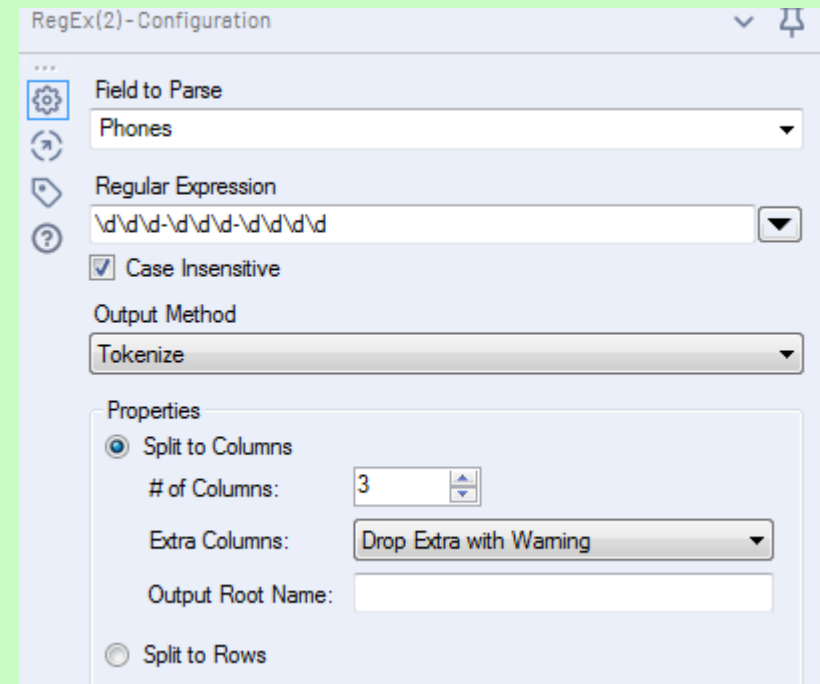


REGEX – TOKENIZE EXAMPLE

- Specify a regular expression to match on and separate into different fields (similar to parsing with Text to Columns Tool)
- Split columns into
 - Columns
 - Rows

Use case:

Filter that matches specific pattern:
identify all phone numbers in a string





WEEKLY CHALLENGE