SMARTVISION
Europe

The new RX nodes in IBM SPSS Modeler

John McConnell – Services

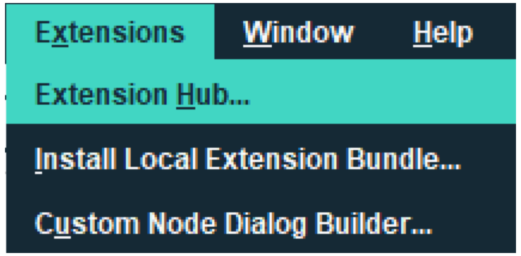
Rachel Clinton – Business Development

FAQ’s

- **Is this session being recorded?** Yes
- **Can I get a copy of the slides?** Yes, we’ll email a PDF copy to you after the session has ended.
- **Can we arrange a re-run for colleagues?** Yes, just ask us.
- **How can I ask questions?** All lines are muted so please use the chat facility – if we run out of time we will follow up with you.

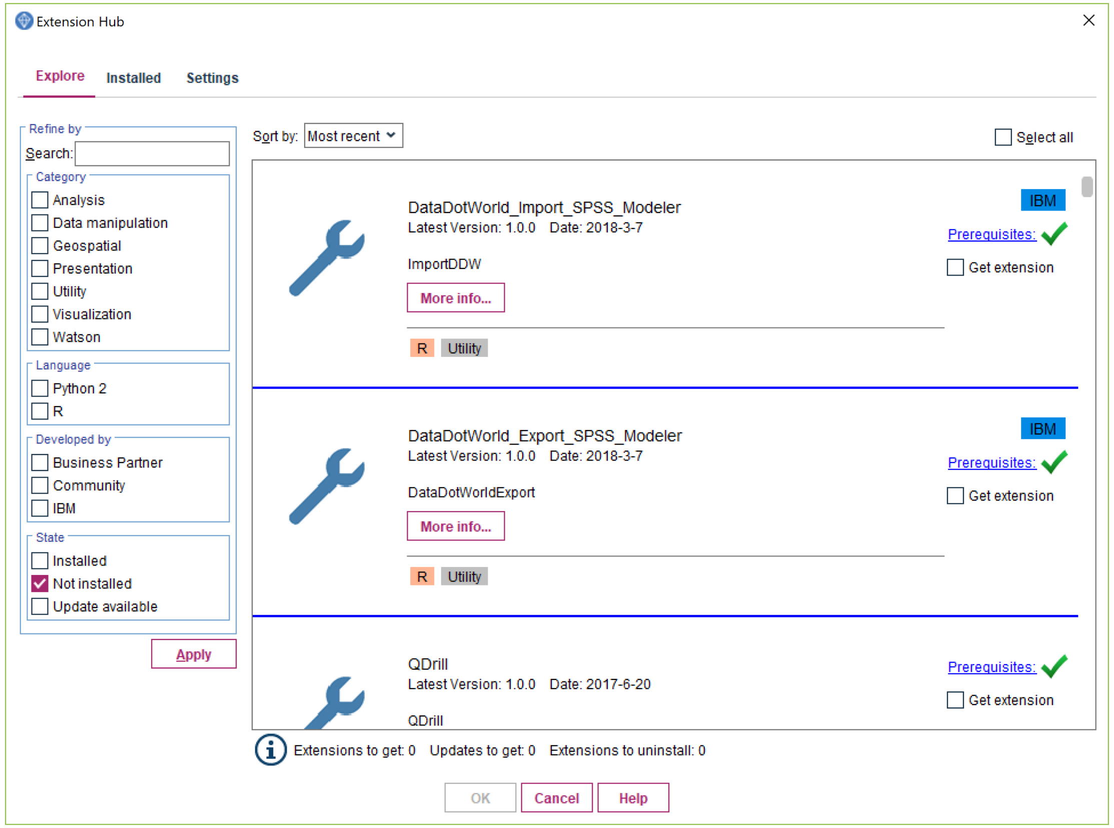
Exploring/creating Extensions (to SPSS Modeler … and SPSS Statistics)

- The **Extension Hub…** is available in both SPSS tools from the **Extensions** menu


Extensions
Window
Help
Extension Hub...
Install Local Extension Bundle...
Custom Node Dialog Builder...

Free and charged extensions are available in the Extension Hub and usually in GitHub too:

https://github.com/IBMPredictiveAnalytics



Refine by

Search:

Category
- □ Analysis
- □ Data manipulation
- □ Geospatial
- □ Presentation
- □ Utility
- □ Visualization
- □ Watson

Language
- □ Python 2
- □ R

Developed by
- □ Business Partner
- □ Community
- □ IBM

State
- □ Installed
- ☑ Not installed
- □ Update available

Sort by: Most recent

□ Select all

DataDotWorld_Import_SPSS_Modeler
Latest Version: 1.0.0 Date: 2018-3-7
ImportDDW
More info...
R Utility
IBM
Prerequisites: ✓
□ Get extension

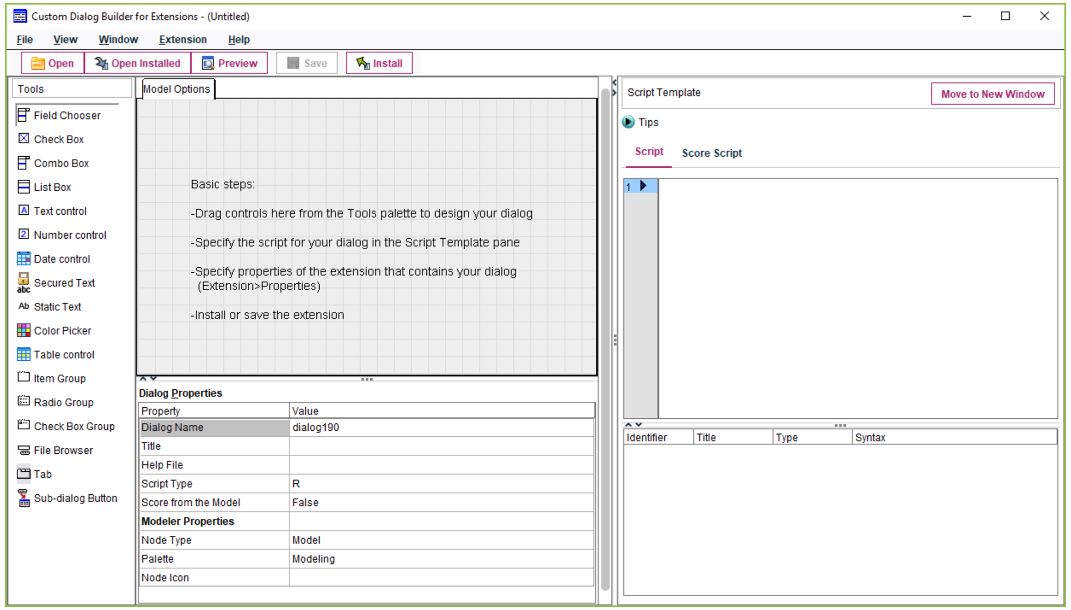
DataDotWorld_Export_SPSS_Modeler
Latest Version: 1.0.0 Date: 2018-3-7
DataDotWorldExport
More info...
R Utility
IBM
Prerequisites: ✓
□ Get extension

QDrill
Latest Version: 1.0.0 Date: 2017-6-20
QDrill
Prerequisites: ✓
□ Get extension

Extensions to get: 0 Updates to get: 0 Extensions to uninstall: 0

2 ways to build Extensions in IBM/SPSS Modeler

1. Using the **Custom Dialog Builder** (available in both SPSS Modeler and SPSS Statistics)
   - Intended to make extension creation for R and Python more accessible
2. **CLEF** (IBM SPSS Modeler CLEF Developer's Guide)
   - More flexible but more technical
   - The RX nodes are built on CLEF
   - CLEF = **CL**ementine **E**xtension **F**ramework



Basic steps:

-Drag controls here from the Tools palette to design your dialog

-Specify the script for your dialog in the Script Template pane

-Specify properties of the extension that contains your dialog (Extension>Properties)

-Install or save the extension

Dialog Properties

Property	Value
Dialog Name	dialog190
Title	
Help File	
Script Type	R
Score from the Model	False
Modeler Properties	
Node Type	Model
Palette	Modeling
Node Icon	

Identifier	Title	Type	Syntax

What is Regex (or Regexp)?

- A standard syntax for defining text search patterns
  - And therefore code that can be used to programmatically manage/prepare text data
- Utilised within broader programming languages … e.g. Python, Java and C++
  - And Perl which has specific syntax for Regex
- Used – under the bonnet in:
  - Search engines
  - Word Processors
  - Text Analytic engines
  - Web forms
  - Etc.
- Can sometimes be used directly in tools e.g.
  - Notepad++ will let you search and replace with regular expressions
    - E.g. **[^\x00-\x7F]+** will find/replace any non-ASCII character
- More detail at https://en.wikipedia.org/wiki/Regular_expression

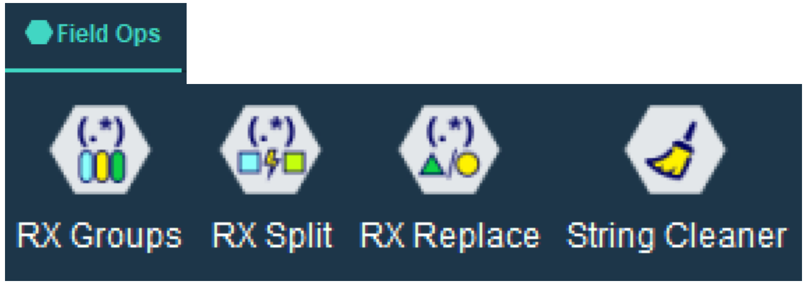
Why Regex in SPSS Modeler?

- Modeler already has some powerful string handling **functions** accessible through the **Expression Builder** in various nodes; Derive, Filler, Select etc.
- The **Text Mining** node (Modeler Premium) can also do some clever text extraction when it does it's Text Analytics
  - Much of which is done using regular expressions under the hood
- But both these areas have limits e.g.
  - If we wanted to split a string multiple times based on the same delimiting character we would need to combine multiple Derive nodes and use multiple functions in concert:
    - loccchar(), substring() … repeat (several times)
  - Text Analytics is very good at identifying specific types of text, like emails, within a field… but it can't tag them, replace them or extract them into a separate field

The 4 RX extension nodes in Modeler

RX Groups
RX Split
RX Replace
String Cleaner

They install into the **Field Ops** palette in Modeler

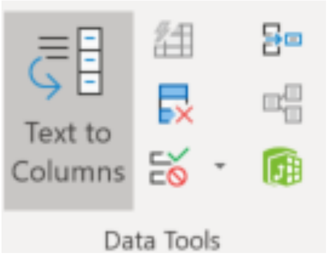


The 4 new RX nodes are available on the Smart Vision Europe site
Under the hood they call one of the most widely used Regex libraries:
ICU Regular Expression Library

RX Split

The RX Split Node

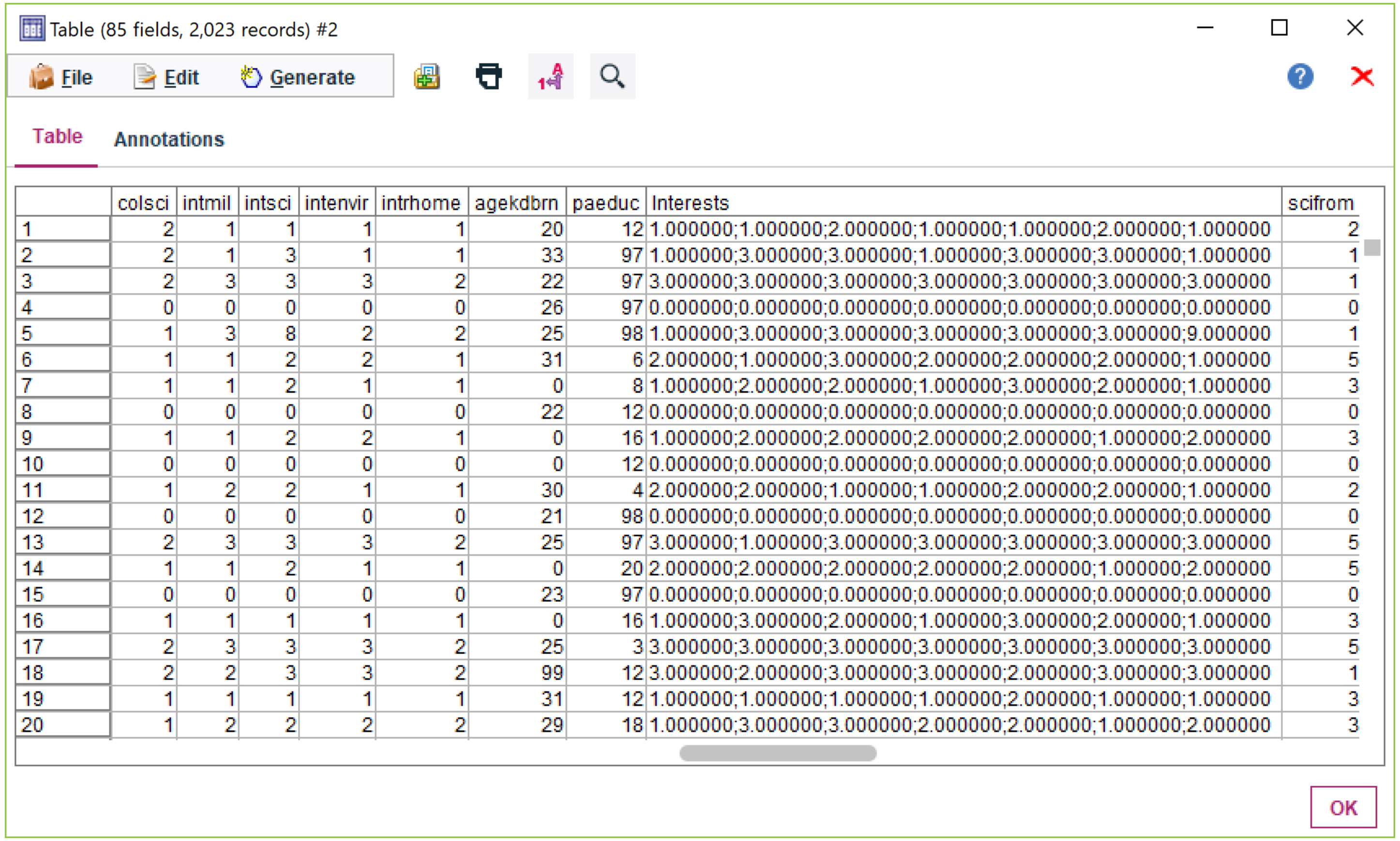
- If you have ever used the **Text to Columns** tool on the **Excel Data Ribbon**…



- … the **RX Split node** gives you the same capability in Modeler
- If we have multiple fields all separated by the same delimiting character e.g. a period or a comma… we often import them – erroneously - as a single text field
  - This often happens when we receive data from on-line survey tools
  - Standard fields have one delimiter and multiple response fields have another
- The RX Split node will split them out and create one field per delimiter

RX Split

The RX Split Node – Source Data

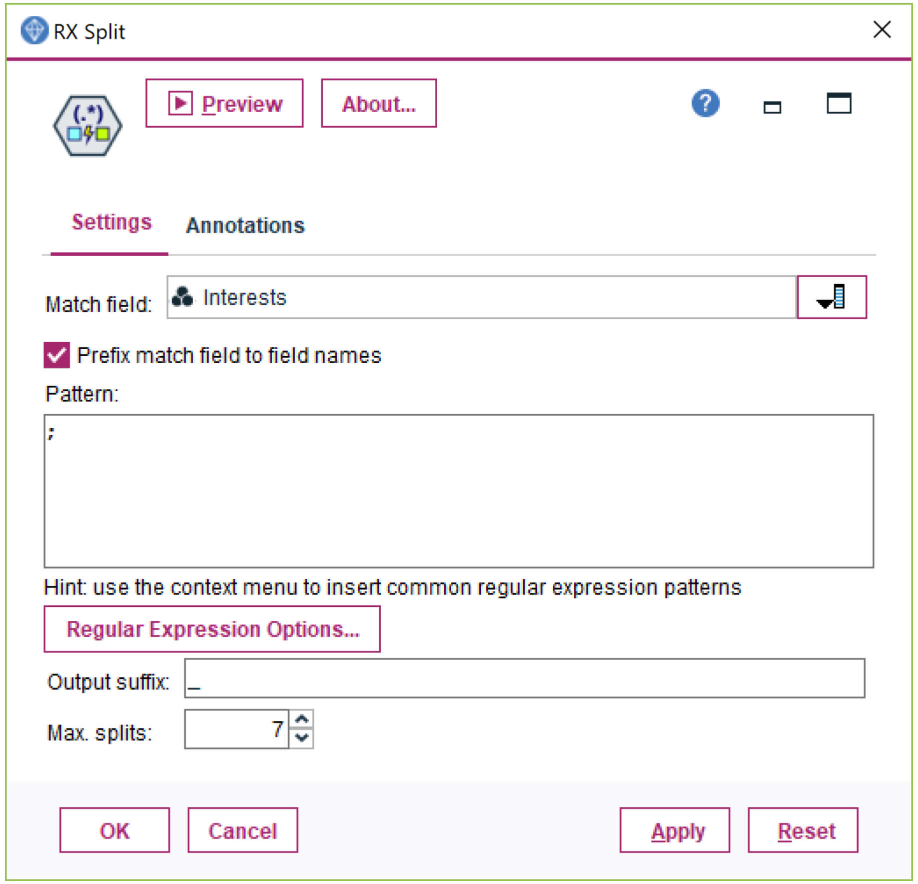


	colsci	intmil	intsci	intenvir	intrhome	agekdbrn	paeduc	Interests	scifrom
1	2	1	1	1	1	20	12	1.000000;1.000000;2.000000;1.000000;1.000000;2.000000;1.000000	2
2	2	1	3	1	1	33	97	1.000000;3.000000;3.000000;1.000000;3.000000;3.000000;1.000000	1
3	2	3	3	3	2	22	97	3.000000;3.000000;3.000000;3.000000;3.000000;3.000000;3.000000	1
4	0	0	0	0	0	26	97	0.000000;0.000000;0.000000;0.000000;0.000000;0.000000;0.000000	0
5	1	3	8	2	2	25	98	1.000000;3.000000;3.000000;3.000000;3.000000;3.000000;9.000000	1
6	1	1	2	2	1	31	6	2.000000;1.000000;3.000000;2.000000;2.000000;2.000000;1.000000	5
7	1	1	2	1	1	0	8	1.000000;2.000000;2.000000;1.000000;3.000000;2.000000;1.000000	3
8	0	0	0	0	0	22	12	0.000000;0.000000;0.000000;0.000000;0.000000;0.000000;0.000000	0
9	1	1	2	2	1	0	16	1.000000;2.000000;2.000000;2.000000;2.000000;1.000000;2.000000	3
10	0	0	0	0	0	0	12	0.000000;0.000000;0.000000;0.000000;0.000000;0.000000;0.000000	0
11	1	2	2	1	1	30	4	2.000000;2.000000;1.000000;1.000000;2.000000;2.000000;1.000000	2
12	0	0	0	0	0	21	98	0.000000;0.000000;0.000000;0.000000;0.000000;0.000000;0.000000	0
13	2	3	3	3	2	25	97	3.000000;1.000000;3.000000;3.000000;3.000000;3.000000;3.000000	5
14	1	1	2	1	1	0	20	2.000000;2.000000;2.000000;2.000000;2.000000;1.000000;2.000000	5
15	0	0	0	0	0	23	97	0.000000;0.000000;0.000000;0.000000;0.000000;0.000000;0.000000	0
16	1	1	1	1	1	0	16	1.000000;3.000000;2.000000;1.000000;3.000000;2.000000;1.000000	3
17	2	3	3	3	2	25	3	3.000000;3.000000;3.000000;3.000000;3.000000;3.000000;3.000000	5
18	2	2	3	3	2	99	12	3.000000;2.000000;3.000000;3.000000;2.000000;3.000000;3.000000	1
19	1	1	1	1	1	31	12	1.000000;1.000000;1.000000;1.000000;2.000000;1.000000;1.000000	3
20	1	2	2	2	2	29	18	1.000000;3.000000;3.000000;2.000000;2.000000;1.000000;2.000000	3

In the middle of our Census data file we find the multi-coded field **Interests** where survey respondents told us their level of interest in 7 topics like politics, the environment, etc.

RX Split

The RX Split Node - Settings



Match field: Interests

☑ Prefix match field to field names

Pattern:

```
;
```

Hint: use the context menu to insert common regular expression patterns

Output suffix: _

Max. splits: 7

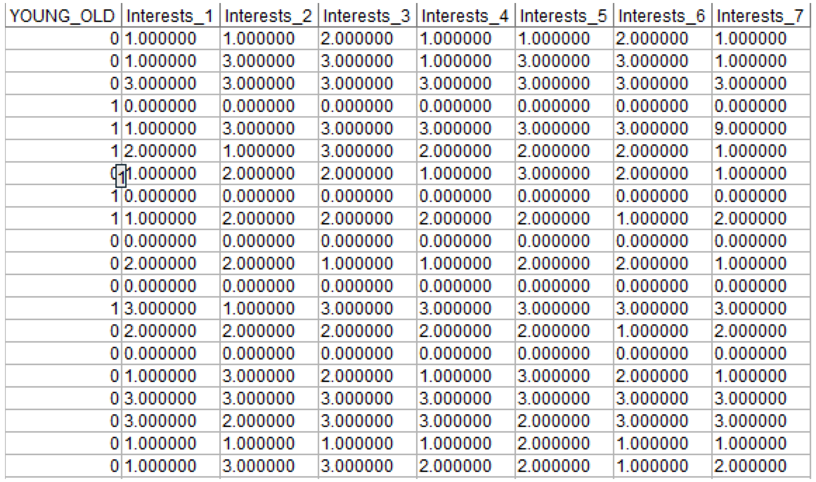
In the pattern box we just specify the delimiter. We could use other regular expression … or a specific character as we do here

In this example we know that there are 7 fields within this single field.
So we set **Max splits:** to 7

RX Split

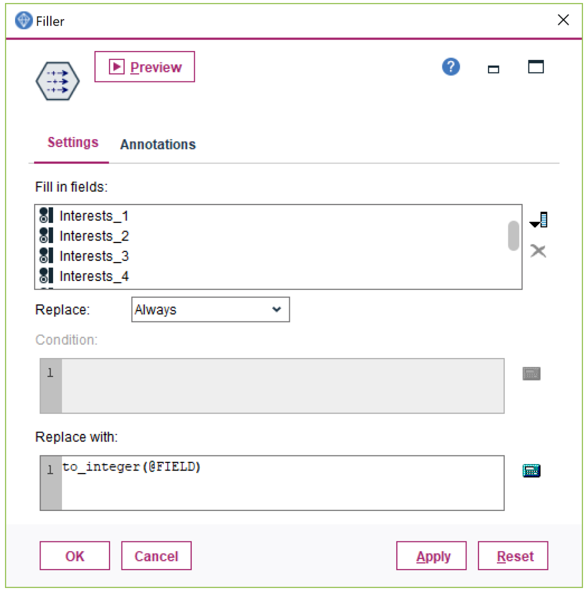
The RX Split Node - Results

The split creates the 7 separate fields ready for analysis.



YOUNG_OLD	Interests_1	Interests_2	Interests_3	Interests_4	Interests_5	Interests_6	Interests_7
0	1.000000	1.000000	2.000000	1.000000	1.000000	2.000000	1.000000
0	1.000000	3.000000	3.000000	1.000000	3.000000	3.000000	1.000000
0	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	1.000000	3.000000	3.000000	3.000000	3.000000	3.000000	9.000000
1	2.000000	1.000000	3.000000	2.000000	2.000000	2.000000	1.000000
0	1.000000	2.000000	2.000000	1.000000	3.000000	2.000000	1.000000
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	1.000000	2.000000	2.000000	2.000000	2.000000	1.000000	2.000000
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0	2.000000	2.000000	1.000000	1.000000	2.000000	2.000000	1.000000
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	3.000000	1.000000	3.000000	3.000000	3.000000	3.000000	3.000000
0	2.000000	2.000000	2.000000	2.000000	2.000000	1.000000	2.000000
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0	1.000000	3.000000	2.000000	1.000000	3.000000	2.000000	1.000000
0	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
0	3.000000	2.000000	3.000000	3.000000	2.000000	3.000000	3.000000
0	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	1.000000
0	1.000000	3.000000	3.000000	2.000000	2.000000	1.000000	2.000000

In their previous format they had decimal places we don't need. We can use a standard filler node to truncate them.



Fill in fields:
- Interests_1
- Interests_2
- Interests_3
- Interests_4

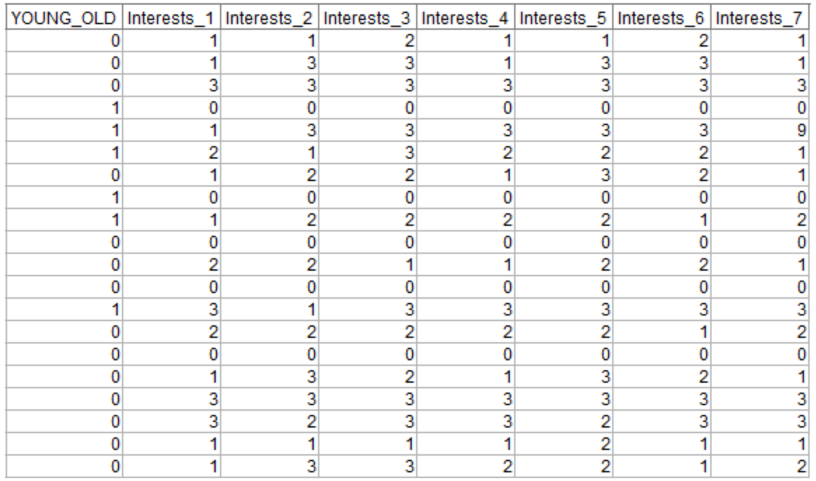
Replace: Always

Condition:

Replace with:

```
to_integer(@FIELD)
```

This leaves us with the integer values ready for analysis.

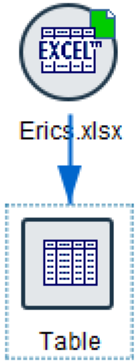


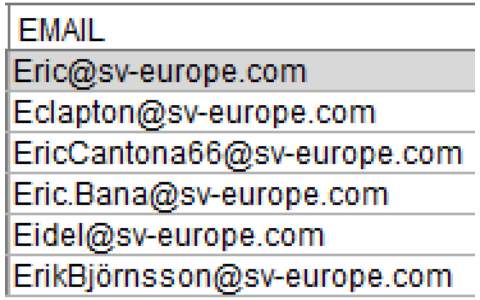
YOUNG_OLD	Interests_1	Interests_2	Interests_3	Interests_4	Interests_5	Interests_6	Interests_7
0	1	1	2	1	1	2	1
0	1	3	3	1	3	3	1
0	3	3	3	3	3	3	3
1	0	0	0	0	0	0	0
1	1	3	3	3	3	3	9
1	2	1	3	2	2	2	1
0	1	2	2	1	3	2	1
1	0	0	0	0	0	0	0
1	1	2	2	2	2	1	2
0	0	0	0	0	0	0	0
0	2	2	1	1	2	2	1
0	0	0	0	0	0	0	0
1	3	1	3	3	3	3	3
0	2	2	2	2	2	1	2
0	0	0	0	0	0	0	0
0	1	3	2	1	3	2	1
0	3	3	3	3	3	3	3
0	3	2	3	3	2	3	3
0	1	1	1	1	2	1	1
0	1	3	3	2	2	1	2

RX Replace

The RX Replace node

- Let’s look to use the **RX Replace** node to redact emails
- Ultimately – in this example - we want to do this in an operational log that contains a mix of text some of which contains **personally identifiable** email names
- But let’s start simple and illustrate how Regex works and how we can build up an expression using the **Context Menu** within the node
- The first part of our stream connects to a short data file of (fictional) Smart Vision Erics …





EMAIL
Eric@sv-europe.com
Eclapton@sv-europe.com
EricCantona66@sv-europe.com
Eric.Bana@sv-europe.com
Eidel@sv-europe.com
ErikBjörnsson@sv-europe.com

Note that Smart Vision don’t have a standard naming convention for emails!

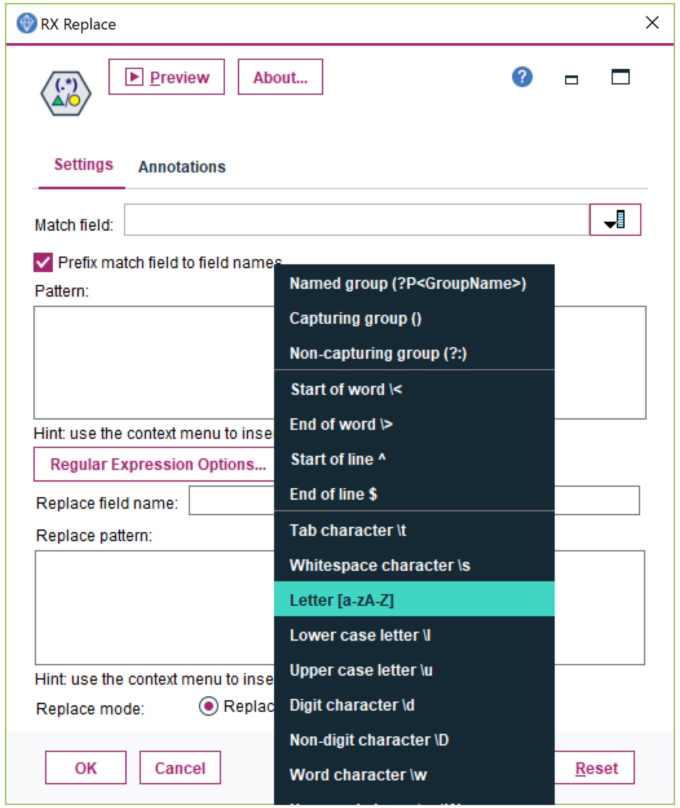
- Let’s say we want to use this view to create some Regex to redact the emails

RX Replace

The RX Replace node – Using the Context Menu

Our identifiable names are mostly made up of letter. We can use the **Context Menu** in the node to automatically select the regex patter that will identify any letter

With that pattern selected we specify the suffix for the field that will contain the adjusted email (_REDACTED) And the text that will replace the Pattern … “REDACTED”



RX Replace

Match field:

☑ Prefix match field to field names

Pattern:

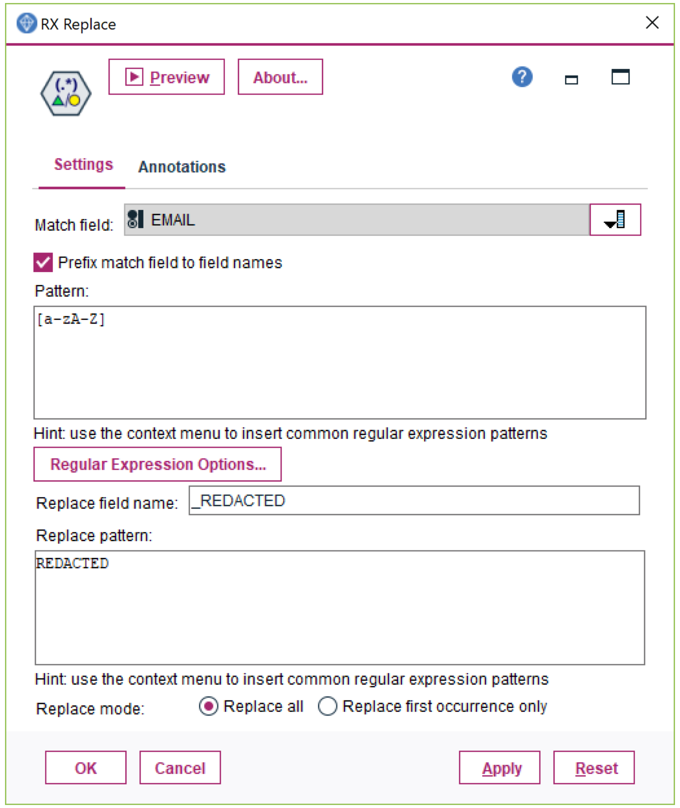
Hint: use the context menu to inse

Replace field name:

Replace pattern:

Hint: use the context menu to inse

Replace mode: ● Replac



RX Replace

Match field: EMAIL

☑ Prefix match field to field names

Pattern:

```
[a-zA-Z]
```

Hint: use the context menu to insert common regular expression patterns

Replace field name: _REDACTED

Replace pattern:

```
REDACTED
```

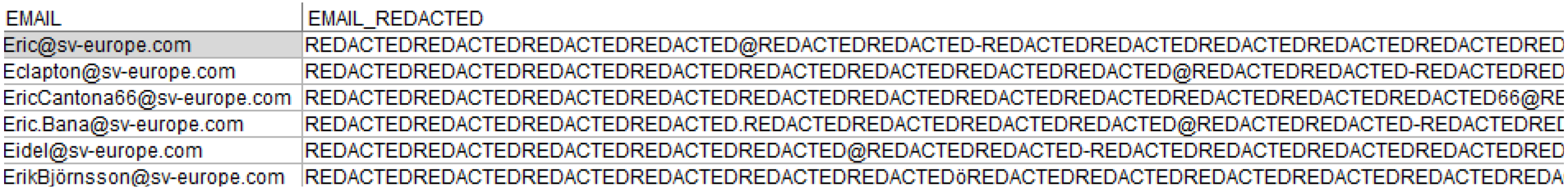
Hint: use the context menu to insert common regular expression patterns

Replace mode: ● Replace all ○ Replace first occurrence only

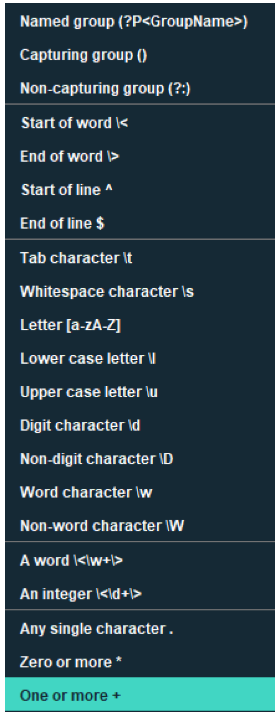
The RX Replace node – Using the Context Menu to improve the Regex

RX Replace

That “sort of” works. We can’t identify the email name any more. But it isn’t very neat. Our current regex has replaced *every* letter with the word redacted

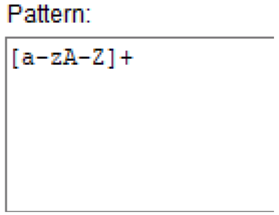


EMAIL	EMAIL_REDACTED
Eric@sv-europe.com	REDACTEDREDACTEDREDACTEDREDACTED@REDACTEDREDACTED-REDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDRED
Eclapton@sv-europe.com	REDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTED@REDACTEDREDACTED-REDACTEDRED
EricCantona66@sv-europe.com	REDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTED66@RE
Eric.Bana@sv-europe.com	REDACTEDREDACTEDREDACTEDREDACTED.REDACTEDREDACTEDREDACTEDREDACTED@REDACTEDREDACTED-REDACTEDRED
Eidel@sv-europe.com	REDACTEDREDACTEDREDACTEDREDACTEDREDACTED@REDACTEDREDACTED-REDACTEDREDACTEDREDACTEDREDACTEDREDACTEDRED
ErikBjörnsson@sv-europe.com	REDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDöREDACTEDREDACTEDREDACTEDREDACTEDREDACTEDREDA



We return to the Context Menu to see that a + sign will replace a pattern of *one or more*

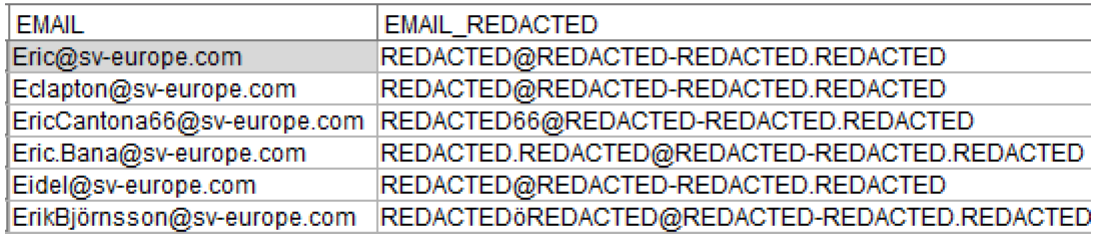
We add that to our current pattern.
The result should now identify one or more letters



Pattern:

```
[a-zA-Z]+
```

This improves matters. But we still have number and the @ sign in the way so we have more “REDACTED”s than we would like…

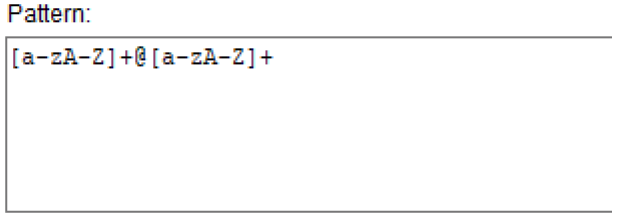


EMAIL	EMAIL_REDACTED
Eric@sv-europe.com	REDACTED@REDACTED-REDACTED.REDACTED
Eclapton@sv-europe.com	REDACTED@REDACTED-REDACTED.REDACTED
EricCantona66@sv-europe.com	REDACTED66@REDACTED-REDACTED.REDACTED
Eric.Bana@sv-europe.com	REDACTED.REDACTED@REDACTED-REDACTED.REDACTED
Eidel@sv-europe.com	REDACTED@REDACTED-REDACTED.REDACTED
ErikBjörnsson@sv-europe.com	REDACTEDöREDACTED@REDACTED-REDACTED.REDACTED

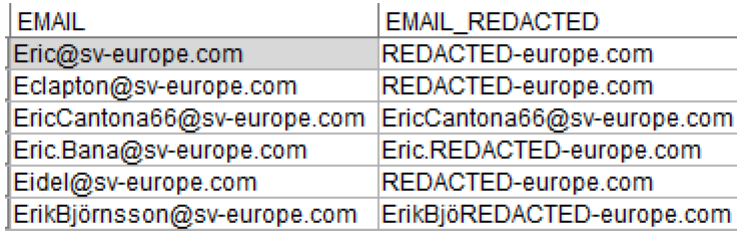
RX Replace

The RX Replace node – Customising the Regex to finalise

To take care of the @ we add it explicitly and repeat the letter pattern on either side of it…

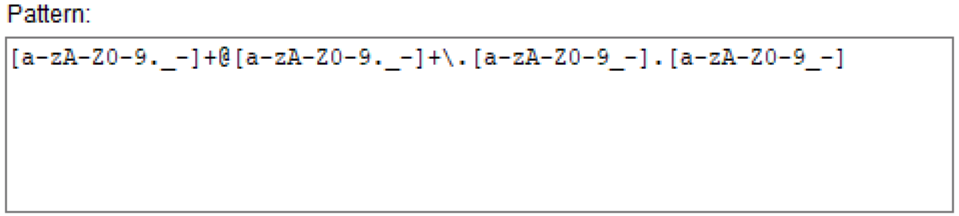

Pattern:

```
[a-zA-Z]+@[a-zA-Z]+
```

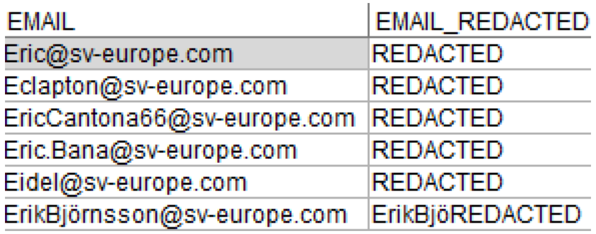


EMAIL	EMAIL_REDACTED
Eric@sv-europe.com	REDACTED-europe.com
Eclapton@sv-europe.com	REDACTED-europe.com
EricCantona66@sv-europe.com	EricCantona66@sv-europe.com
Eric.Bana@sv-europe.com	Eric.REDACTED-europe.com
Eidel@sv-europe.com	REDACTED-europe.com
ErikBjörnsson@sv-europe.com	ErikBjöREDACTED-europe.com

Numbers and hyphens are in the way so we add patterns to take care of those…

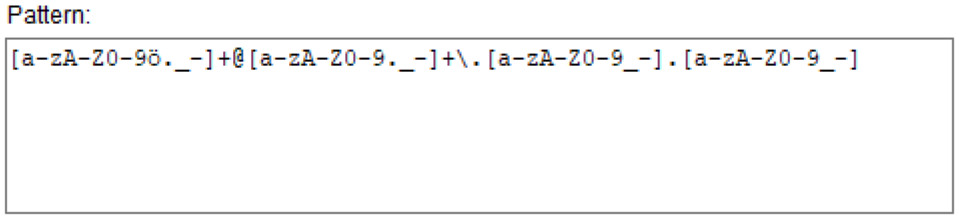

Pattern:

```
[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z0-9_-].[a-zA-Z0-9_-]
```

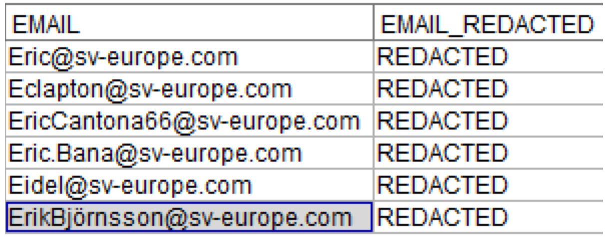


EMAIL	EMAIL_REDACTED
Eric@sv-europe.com	REDACTED
Eclapton@sv-europe.com	REDACTED
EricCantona66@sv-europe.com	REDACTED
Eric.Bana@sv-europe.com	REDACTED
Eidel@sv-europe.com	REDACTED
ErikBjörnsson@sv-europe.com	ErikBjöREDACTED

This just leaves us with the non English/standard ASCII Ö. We can add that explicitly…


Pattern:

```
[a-zA-Z0-9ö._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z0-9_-].[a-zA-Z0-9_-]
```



EMAIL	EMAIL_REDACTED
Eric@sv-europe.com	REDACTED
Eclapton@sv-europe.com	REDACTED
EricCantona66@sv-europe.com	REDACTED
Eric.Bana@sv-europe.com	REDACTED
Eidel@sv-europe.com	REDACTED
ErikBjörnsson@sv-europe.com	REDACTED

RX Replace

The RX Replace node – Deploying to operational data

Here we have an operational log file generated by monitoring equipment and technicians across a mobile telco network.
We want to apply Text Analytics to it to look for themes in the reports that help us to diagnose and predict maintenance events
However the engineers frequently add emails into the middle of main report

To remove these emails se can look to reuse the last version of the email redaction Replace..

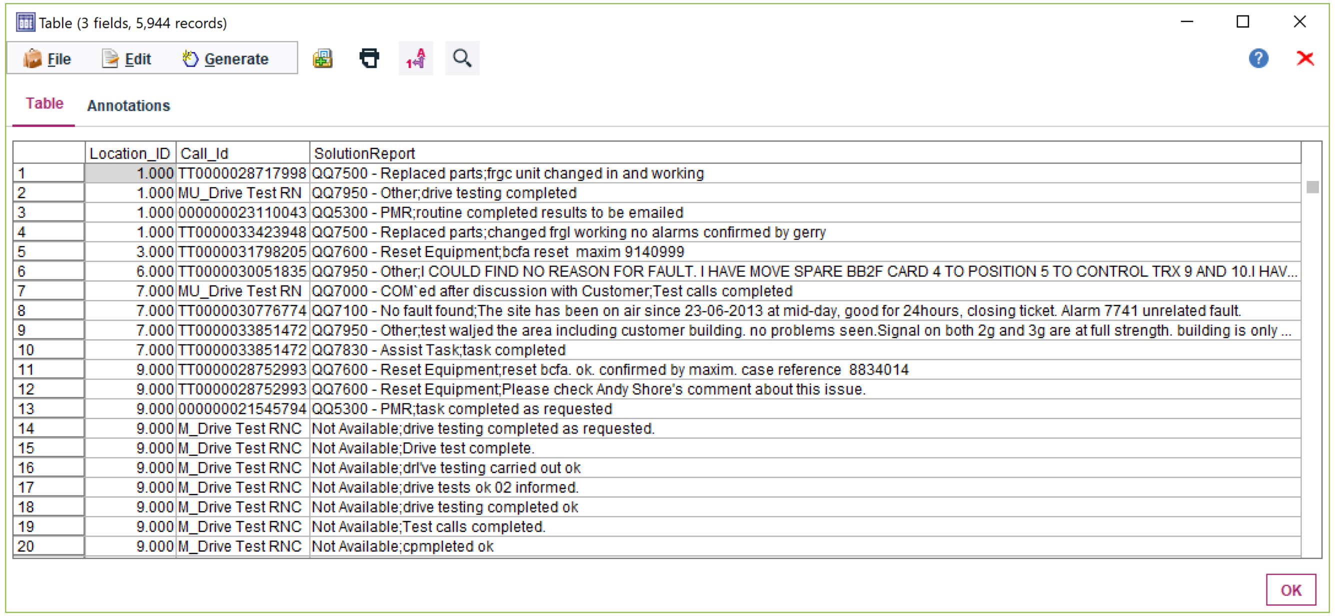
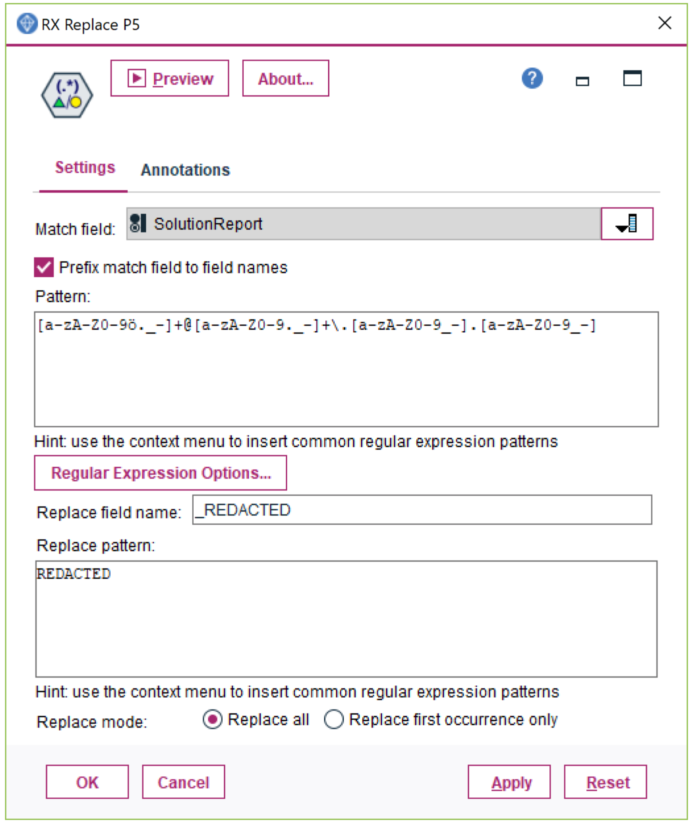


Table (3 fields, 5,944 records)

Table | Annotations

	Location_ID	Call_Id	SolutionReport
1	1.000	TT0000028717998	QQ7500 - Replaced parts;frgc unit changed in and working
2	1.000	MU_Drive Test RN	QQ7950 - Other;drive testing completed
3	1.000	000000023110043	QQ5300 - PMR;routine completed results to be emailed
4	1.000	TT0000033423948	QQ7500 - Replaced parts;changed frgl working no alarms confirmed by gerry
5	3.000	TT0000031798205	QQ7600 - Reset Equipment;bcfa reset maxim 9140999
6	6.000	TT0000030051835	QQ7950 - Other;I COULD FIND NO REASON FOR FAULT. I HAVE MOVE SPARE BB2F CARD 4 TO POSITION 5 TO CONTROL TRX 9 AND 10.I HAV...
7	7.000	MU_Drive Test RN	QQ7000 - COM`ed after discussion with Customer;Test calls completed
8	7.000	TT0000030776774	QQ7100 - No fault found;The site has been on air since 23-06-2013 at mid-day, good for 24hours, closing ticket. Alarm 7741 unrelated fault.
9	7.000	TT0000033851472	QQ7950 - Other;test waljed the area including customer building. no problems seen.Signal on both 2g and 3g are at full strength. building is only ...
10	7.000	TT0000033851472	QQ7830 - Assist Task;task completed
11	9.000	TT0000028752993	QQ7600 - Reset Equipment;reset bcfa. ok. confirmed by maxim. case reference 8834014
12	9.000	TT0000028752993	QQ7600 - Reset Equipment;Please check Andy Shore's comment about this issue.
13	9.000	000000021545794	QQ5300 - PMR;task completed as requested
14	9.000	M_Drive Test RNC	Not Available;drive testing completed as requested.
15	9.000	M_Drive Test RNC	Not Available;Drive test complete.
16	9.000	M_Drive Test RNC	Not Available;drl've testing carried out ok
17	9.000	M_Drive Test RNC	Not Available;drive tests ok 02 informed.
18	9.000	M_Drive Test RNC	Not Available;drive testing completed ok
19	9.000	M_Drive Test RNC	Not Available;Test calls completed.
20	9.000	M_Drive Test RNC	Not Available;cpmpleted ok



RX Replace P5

Settings | Annotations

Match field: SolutionReport

☑ Prefix match field to field names

Pattern:

```
[a-zA-Z0-9ö._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z0-9_-].[a-zA-Z0-9_-]
```

Hint: use the context menu to insert common regular expression patterns

Regular Expression Options...

Replace field name: _REDACTED

Replace pattern:

```
REDACTED
```

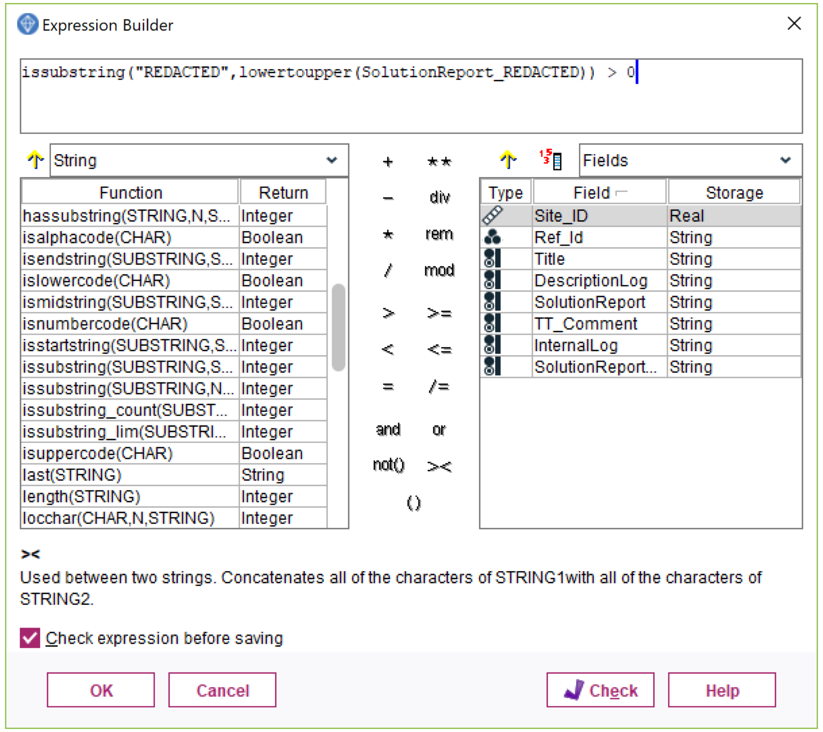
Hint: use the context menu to insert common regular expression patterns

Replace mode: ● Replace all ○ Replace first occurrence only

RX Replace

The RX Replace node – Results

Back to regular Modeler nodes we can create a flag for the Redacted records…



Expression Builder

```
issubstring("REDACTED",lowertoupper(SolutionReport_REDACTED)) > 0
```

String

Function	Return
hassubstring(STRING,N,S...	Integer
isalphacode(CHAR)	Boolean
isendstring(SUBSTRING,S...	Integer
islowercode(CHAR)	Boolean
ismidstring(SUBSTRING,S...	Integer
isnumbercode(CHAR)	Boolean
isstartstring(SUBSTRING,S...	Integer
issubstring(SUBSTRING,S...	Integer
issubstring(SUBSTRING,N...	Integer
issubstring_count(SUBST...	Integer
issubstring_lim(SUBSTRI...	Integer
isuppercode(CHAR)	Boolean
last(STRING)	String
length(STRING)	Integer
locchar(CHAR,N,STRING)	Integer

Fields

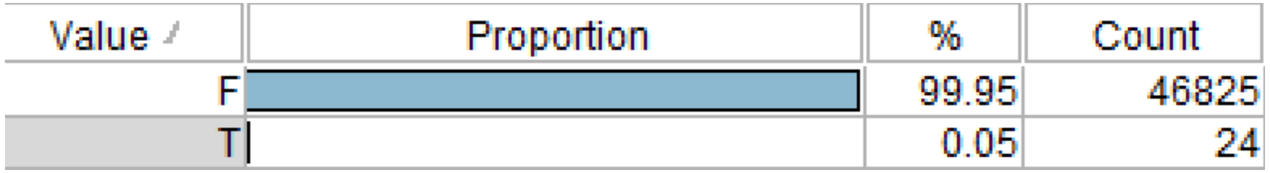
Type	Field	Storage
	Site_ID	Real
	Ref_Id	String
	Title	String
	DescriptionLog	String
	SolutionReport	String
	TT_Comment	String
	InternalLog	String
	SolutionReport...	String

><

Used between two strings. Concatenates all of the characters of STRING1with all of the characters of STRING2.

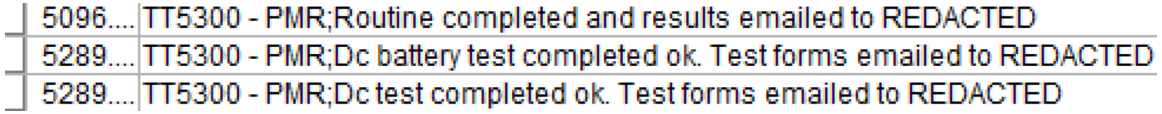
☑ Check expression before saving

And look at how many we have. There aren't many. Needles in a haystack…



Value	Proportion	%	Count
F		99.95	46825
T		0.05	24

And check the way the redaction is added to individual data records…

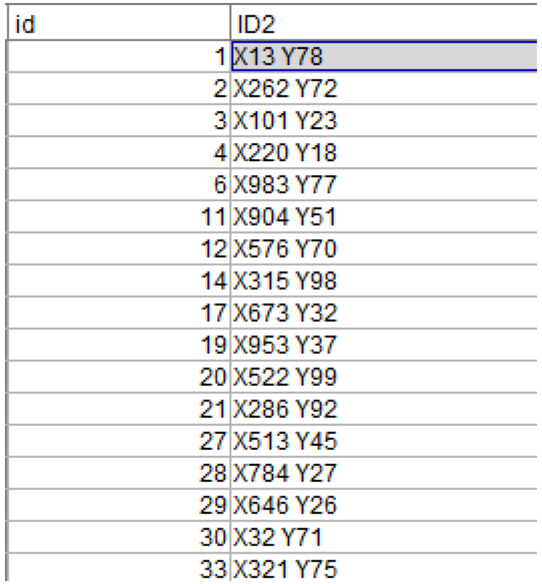


5096....	TT5300 - PMR;Routine completed and results emailed to REDACTED
5289....	TT5300 - PMR;Dc battery test completed ok. Test forms emailed to REDACTED
5289....	TT5300 - PMR;Dc test completed ok. Test forms emailed to REDACTED

String Cleaner

The String Cleaner node

- Compared with the other nodes in the set the String Cleaner is the odd one out
- **To use it we don’t need topick/enter any Regex**
- The Regex happens behind the scenes
- We have an ID field in the census data file that need to be converted to be compatible with another data source that we need to merge (join) with:
  - We need to remove any letters … just leaving the numbers
  - We need to remove any spaces/table in the field

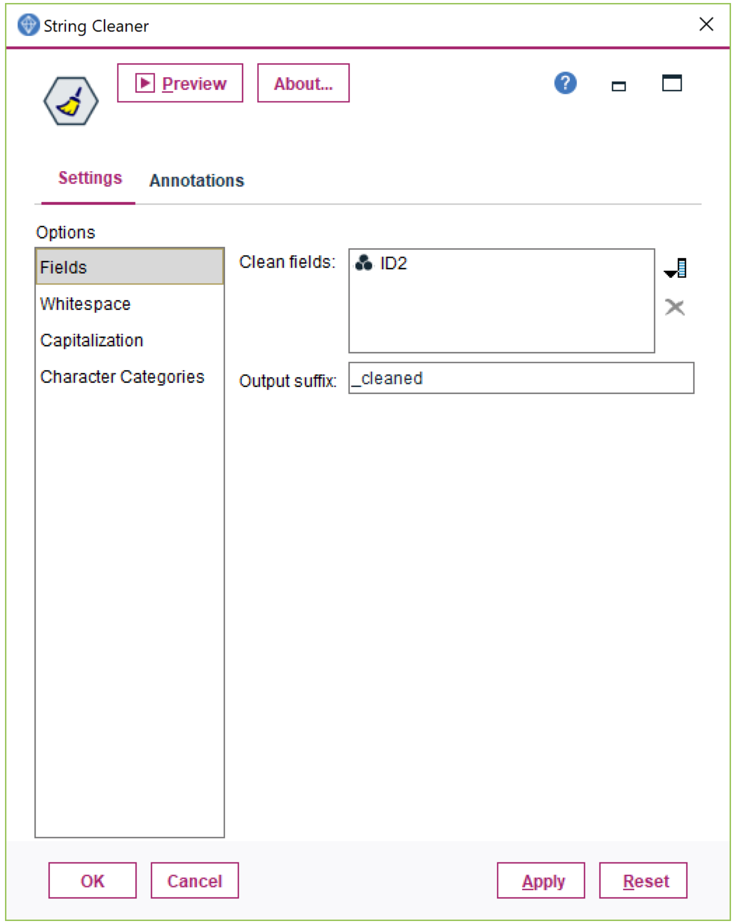


id	ID2
1	X13 Y78
2	X262 Y72
3	X101 Y23
4	X220 Y18
6	X983 Y77
11	X904 Y51
12	X576 Y70
14	X315 Y98
17	X673 Y32
19	X953 Y37
20	X522 Y99
21	X286 Y92
27	X513 Y45
28	X784 Y27
29	X646 Y26
30	X32 Y71
33	X321 Y75

String Cleaner

The String Cleaner node - Specification

Pick the field to clean…



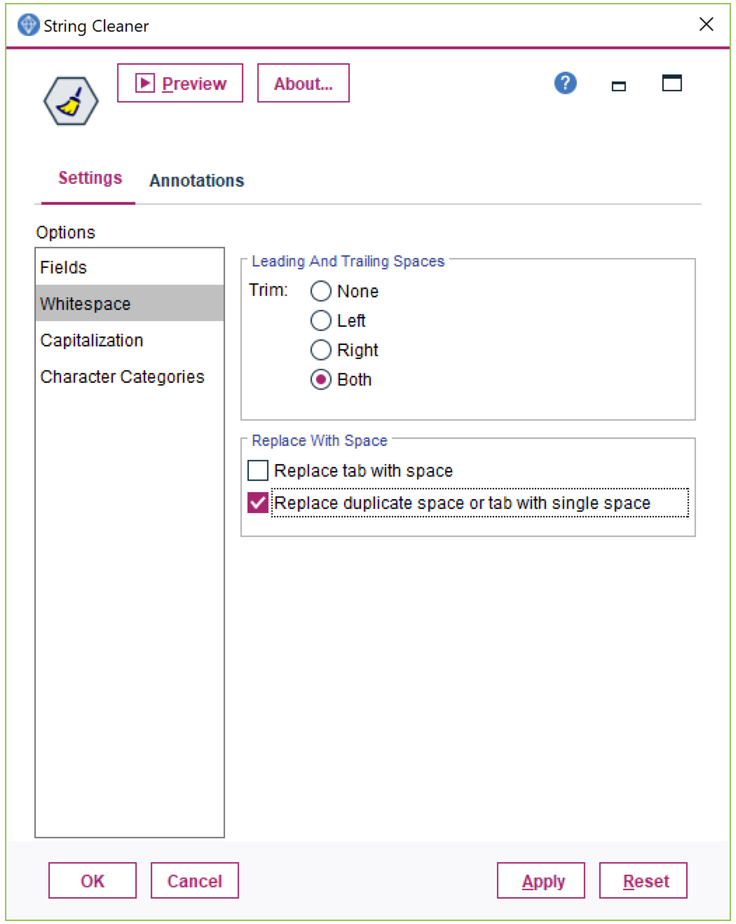
Options

- Fields
- Whitespace
- Capitalization
- Character Categories

Clean fields: ID2

Output suffix: _cleaned

Remove/reduce whitespace…



Options

- Fields
- Whitespace
- Capitalization
- Character Categories

Leading And Trailing Spaces

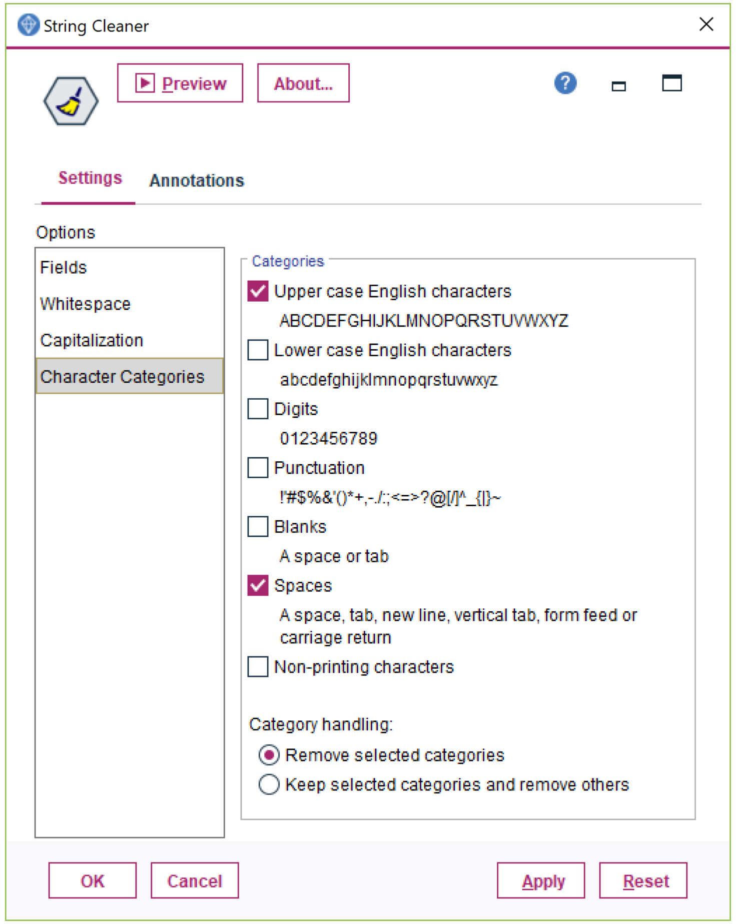
Trim: ○ None ○ Left ○ Right ● Both

Replace With Space

☐ Replace tab with space

☑ Replace duplicate space or tab with single space

Remove text and space…



Options

- Fields
- Whitespace
- Capitalization
- Character Categories

Categories

☑ Upper case English characters
ABCDEFGHIJKLMNOPQRSTUVWXYZ

☐ Lower case English characters
abcdefghijklmnopqrstuvwxyz

☐ Digits
0123456789

☐ Punctuation
!'#$%&'()*+,-./:;<=>?@[/]^_{|}~

☐ Blanks
A space or tab

☑ Spaces
A space, tab, new line, vertical tab, form feed or carriage return

☐ Non-printing characters

Category handling:

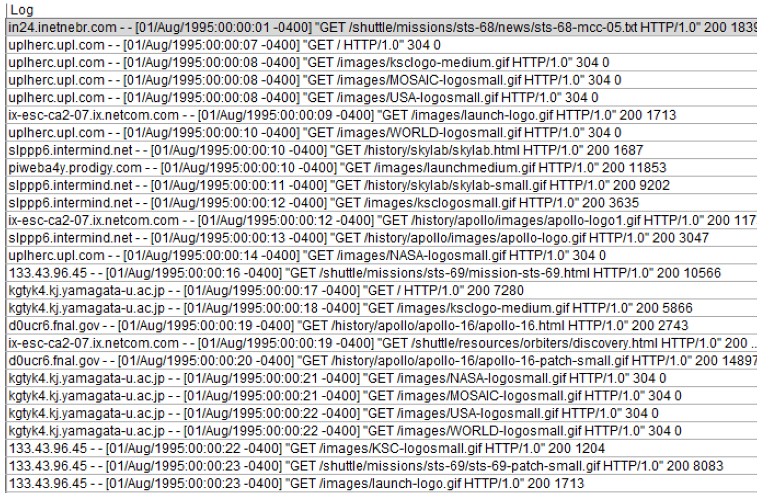
● Remove selected categories

○ Keep selected categories and remove others

RX Groups

The RX Groups node

- We use the RX Group node to parse out pattern groups in longer text fields into new fields based on the patterns
  - Somewhat like the RX Split node but where were we don’t have a consistent delimiter
- In this example we have data from a NASA packet sniffing traces* containing IP addresses, timestamps, URLs, etc.
- Our objective is to separate out the contents of each record into separate fields for further analysis

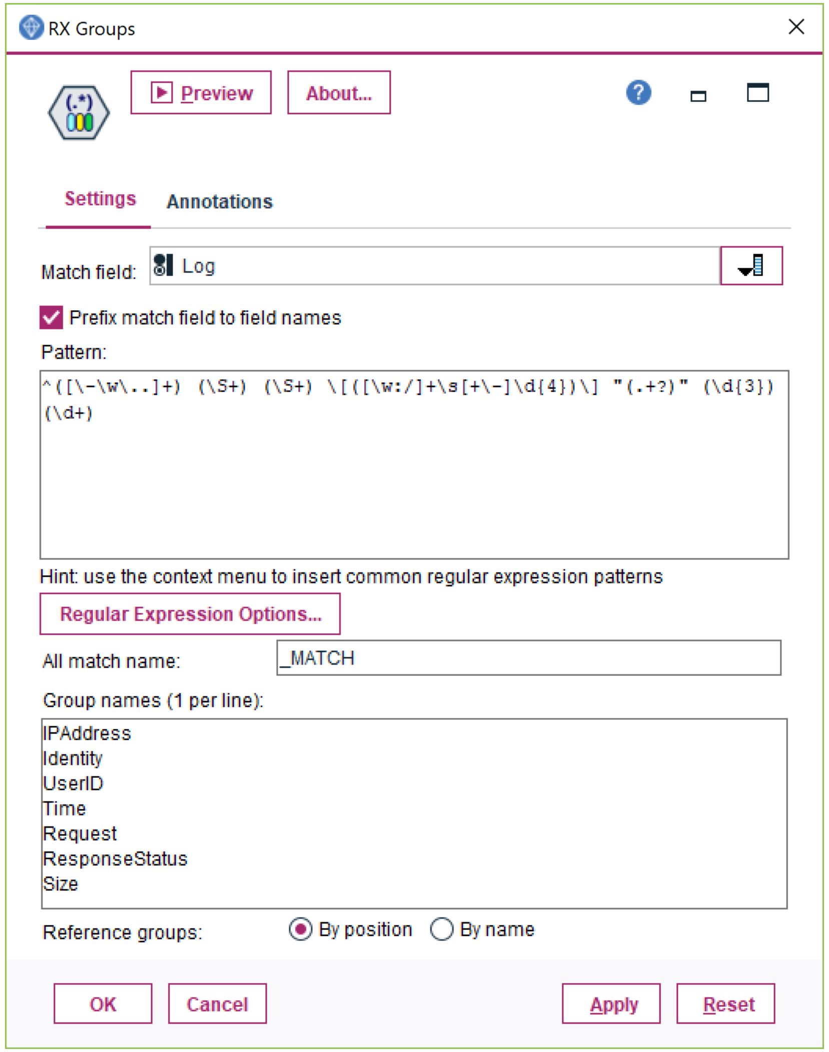


```
Log
in24.inetnebr.com - - [01/Aug/1995:00:00:01 -0400] "GET /shuttle/missions/sts-68/news/sts-68-mcc-05.txt HTTP/1.0" 200 183
uplherc.upl.com - - [01/Aug/1995:00:00:07 -0400] "GET / HTTP/1.0" 304 0
uplherc.upl.com - - [01/Aug/1995:00:00:08 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 304 0
uplherc.upl.com - - [01/Aug/1995:00:00:08 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 304 0
uplherc.upl.com - - [01/Aug/1995:00:00:08 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 304 0
ix-esc-ca2-07.ix.netcom.com - - [01/Aug/1995:00:00:09 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713
uplherc.upl.com - - [01/Aug/1995:00:00:10 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 304 0
slppp6.intermind.net - - [01/Aug/1995:00:00:10 -0400] "GET /history/skylab/skylab.html HTTP/1.0" 200 1687
piweba4y.prodigy.com - - [01/Aug/1995:00:00:10 -0400] "GET /images/launchmedium.gif HTTP/1.0" 200 11853
slppp6.intermind.net - - [01/Aug/1995:00:00:11 -0400] "GET /history/skylab/skylab-small.gif HTTP/1.0" 200 9202
slppp6.intermind.net - - [01/Aug/1995:00:00:12 -0400] "GET /images/ksclogosmall.gif HTTP/1.0" 200 3635
ix-esc-ca2-07.ix.netcom.com - - [01/Aug/1995:00:00:12 -0400] "GET /history/apollo/images/apollo-logo1.gif HTTP/1.0" 200 117
slppp6.intermind.net - - [01/Aug/1995:00:00:13 -0400] "GET /history/apollo/images/apollo-logo.gif HTTP/1.0" 200 3047
uplherc.upl.com - - [01/Aug/1995:00:00:14 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
133.43.96.45 - - [01/Aug/1995:00:00:16 -0400] "GET /shuttle/missions/sts-69/mission-sts-69.html HTTP/1.0" 200 10566
kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:17 -0400] "GET / HTTP/1.0" 200 7280
kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:18 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 200 5866
d0ucr6.fnal.gov - - [01/Aug/1995:00:00:19 -0400] "GET /history/apollo/apollo-16/apollo-16.html HTTP/1.0" 200 2743
ix-esc-ca2-07.ix.netcom.com - - [01/Aug/1995:00:00:19 -0400] "GET /shuttle/resources/orbiters/discovery.html HTTP/1.0" 200 ..
d0ucr6.fnal.gov - - [01/Aug/1995:00:00:20 -0400] "GET /history/apollo/apollo-16/apollo-16-patch-small.gif HTTP/1.0" 200 14897
kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:21 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:21 -0400] "GET /images/MOSAIC-logosmall.gif HTTP/1.0" 304 0
kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:22 -0400] "GET /images/USA-logosmall.gif HTTP/1.0" 304 0
kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:22 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 304 0
133.43.96.45 - - [01/Aug/1995:00:00:22 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204
133.43.96.45 - - [01/Aug/1995:00:00:23 -0400] "GET /shuttle/missions/sts-69/sts-69-patch-small.gif HTTP/1.0" 200 8083
133.43.96.45 - - [01/Aug/1995:00:00:23 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713
```

Source: ftp://ita.ee.lbl.gov/traces

RX Groups

The RX Groups node - Specifying



Match field: Log

☑ Prefix match field to field names

Pattern:

```
^([\-\w\..]+) (\S+) (\S+) \[([\w:/]+\s[+\-]\d{4})\] "(.+?)" (\d{3})
(\d+)
```

Hint: use the context menu to insert common regular expression patterns

All match name: _MATCH

Group names (1 per line):

```
IPAddress
Identity
UserID
Time
Request
ResponseStatus
Size
```

Reference groups: ● By position ○ By name

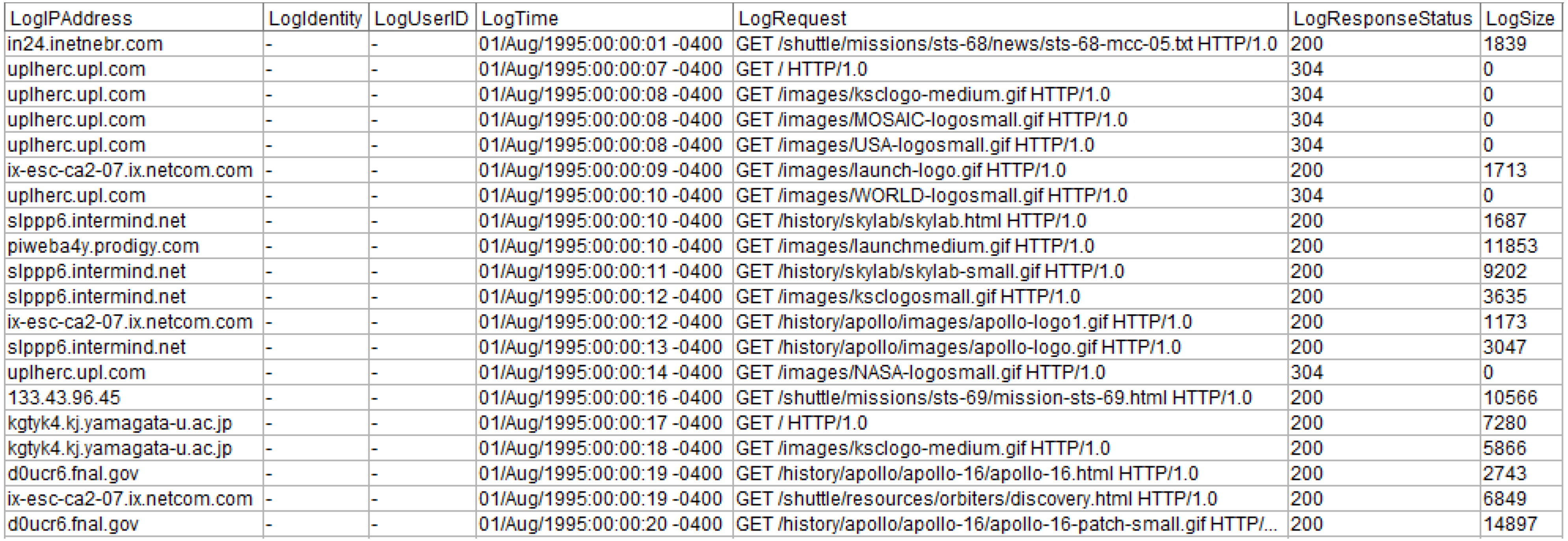
The 7 separate Regex patterns specified here

Are mapped (in order) into the 7 derived fields here

RX Groups

The RX Groups node - Results

The first 20 records with the derived **RX Group** fields are shown below.
The **LogIPAddress** looks read for analysis – or perhaps to use as an ID for modelling e.g. Sequence analysis
We could use RX Split to take it further (splitting on the period to isolate domains)



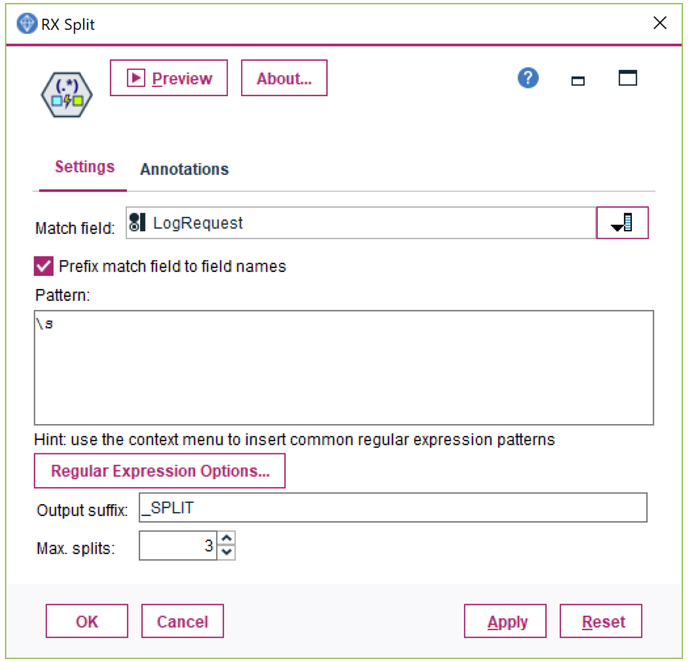
LogIPAddress	LogIdentity	LogUserID	LogTime	LogRequest	LogResponseStatus	LogSize
in24.inetnebr.com	-	-	01/Aug/1995:00:00:01 -0400	GET /shuttle/missions/sts-68/news/sts-68-mcc-05.txt HTTP/1.0	200	1839
uplherc.upl.com	-	-	01/Aug/1995:00:00:07 -0400	GET / HTTP/1.0	304	0
uplherc.upl.com	-	-	01/Aug/1995:00:00:08 -0400	GET /images/ksclogo-medium.gif HTTP/1.0	304	0
uplherc.upl.com	-	-	01/Aug/1995:00:00:08 -0400	GET /images/MOSAIC-logosmall.gif HTTP/1.0	304	0
uplherc.upl.com	-	-	01/Aug/1995:00:00:08 -0400	GET /images/USA-logosmall.gif HTTP/1.0	304	0
ix-esc-ca2-07.ix.netcom.com	-	-	01/Aug/1995:00:00:09 -0400	GET /images/launch-logo.gif HTTP/1.0	200	1713
uplherc.upl.com	-	-	01/Aug/1995:00:00:10 -0400	GET /images/WORLD-logosmall.gif HTTP/1.0	304	0
slppp6.intermind.net	-	-	01/Aug/1995:00:00:10 -0400	GET /history/skylab/skylab.html HTTP/1.0	200	1687
piweba4y.prodigy.com	-	-	01/Aug/1995:00:00:10 -0400	GET /images/launchmedium.gif HTTP/1.0	200	11853
slppp6.intermind.net	-	-	01/Aug/1995:00:00:11 -0400	GET /history/skylab/skylab-small.gif HTTP/1.0	200	9202
slppp6.intermind.net	-	-	01/Aug/1995:00:00:12 -0400	GET /images/ksclogosmall.gif HTTP/1.0	200	3635
ix-esc-ca2-07.ix.netcom.com	-	-	01/Aug/1995:00:00:12 -0400	GET /history/apollo/images/apollo-logo1.gif HTTP/1.0	200	1173
slppp6.intermind.net	-	-	01/Aug/1995:00:00:13 -0400	GET /history/apollo/images/apollo-logo.gif HTTP/1.0	200	3047
uplherc.upl.com	-	-	01/Aug/1995:00:00:14 -0400	GET /images/NASA-logosmall.gif HTTP/1.0	304	0
133.43.96.45	-	-	01/Aug/1995:00:00:16 -0400	GET /shuttle/missions/sts-69/mission-sts-69.html HTTP/1.0	200	10566
kgtyk4.kj.yamagata-u.ac.jp	-	-	01/Aug/1995:00:00:17 -0400	GET / HTTP/1.0	200	7280
kgtyk4.kj.yamagata-u.ac.jp	-	-	01/Aug/1995:00:00:18 -0400	GET /images/ksclogo-medium.gif HTTP/1.0	200	5866
d0ucr6.fnal.gov	-	-	01/Aug/1995:00:00:19 -0400	GET /history/apollo/apollo-16/apollo-16.html HTTP/1.0	200	2743
ix-esc-ca2-07.ix.netcom.com	-	-	01/Aug/1995:00:00:19 -0400	GET /shuttle/resources/orbiters/discovery.html HTTP/1.0	200	6849
d0ucr6.fnal.gov	-	-	01/Aug/1995:00:00:20 -0400	GET /history/apollo/apollo-16/apollo-16-patch-small.gif HTTP/...	200	14897

But let's focus on the **LogRequest**. Let's RX Split that further …

RX Split

RX Splitting our Log Request

Our **LogRequest** has 3 parts … with the interesting part (the page/content seen) in the middle. This is space delimited so we can return to our RX Split node to isolate the interesting text into a separate field (we will call it URL)



Match field: LogRequest

☑ Prefix match field to field names

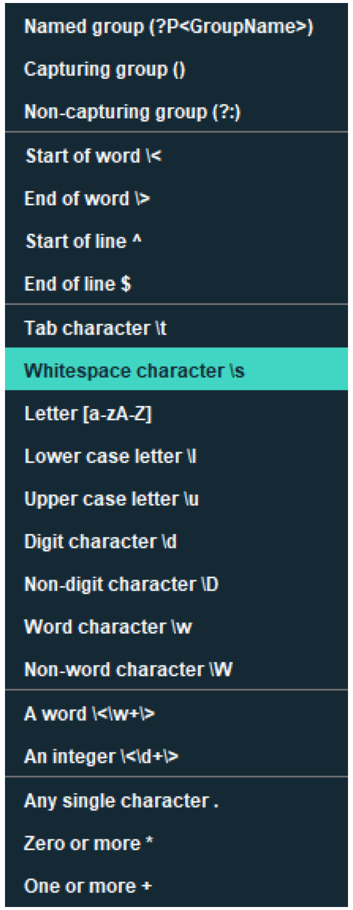
Pattern:

```
\s
```

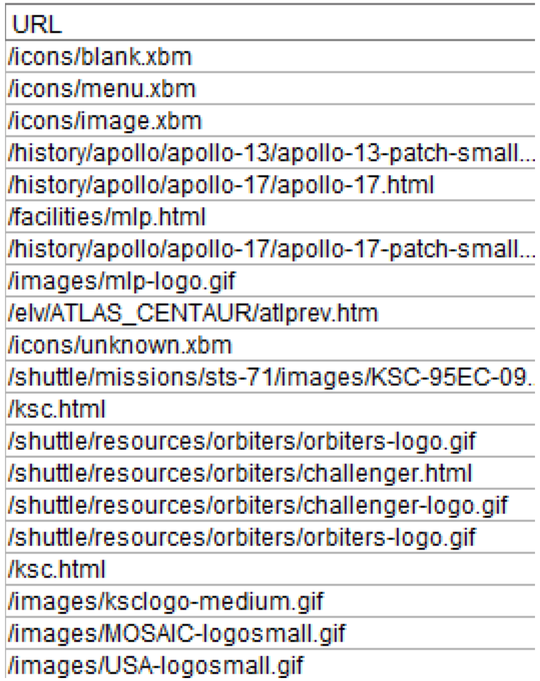
Hint: use the context menu to insert common regular expression patterns

Output suffix: _SPLIT

Max. splits: 3



It looks like the **LogRequest** is space delimited so we can use the **Context Menu** to specify the **Whitespace character** (\s) as our delimiter. This gives us the following field as a URL…

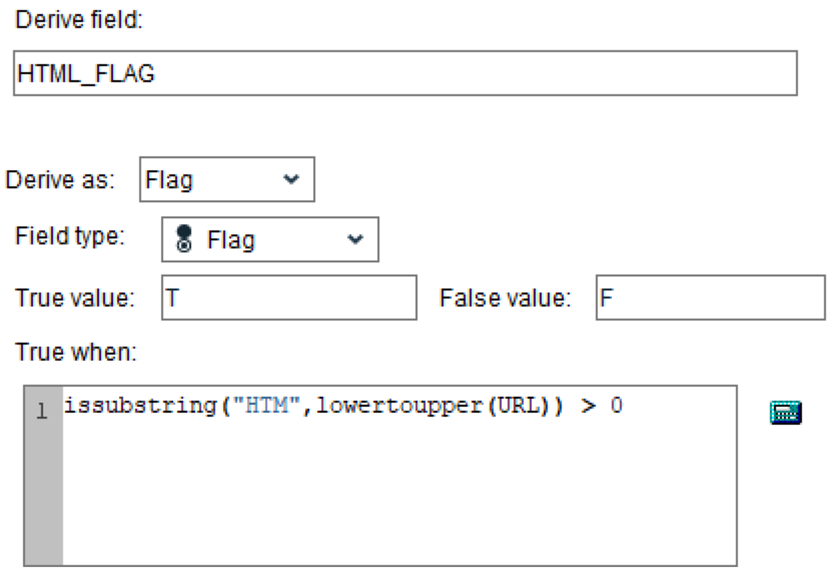


URL
/icons/blank.xbm
/icons/menu.xbm
/icons/image.xbm
/history/apollo/apollo-13/apollo-13-patch-small...
/history/apollo/apollo-17/apollo-17.html
/facilities/mlp.html
/history/apollo/apollo-17/apollo-17-patch-small...
/images/mlp-logo.gif
/elv/ATLAS_CENTAUR/atlprev.htm
/icons/unknown.xbm
/shuttle/missions/sts-71/images/KSC-95EC-09.
/ksc.html
/shuttle/resources/orbiters/orbiters-logo.gif
/shuttle/resources/orbiters/challenger.html
/shuttle/resources/orbiters/challenger-logo.gif
/shuttle/resources/orbiters/orbiters-logo.gif
/ksc.html
/images/ksclogo-medium.gif
/images/MOSAIC-logosmall.gif
/images/USA-logosmall.gif

RX Groups

Isolating HTML URLs for further analysis

To finish off we **Derive** a flag field to tag URLs that have identify interesting content viewed by the visitor (HTML content)…



Derive field: HTML_FLAG

Derive as: Flag

Field type: Flag

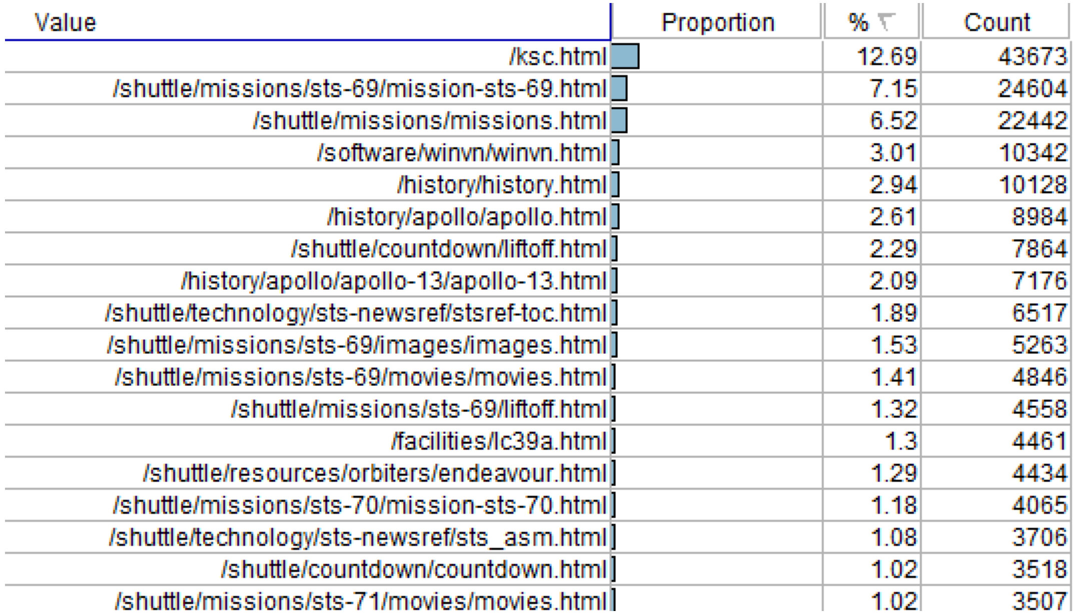
True value: T

False value: F

True when:

```
issubstring("HTM",lowertoupper(URL)) > 0
```

We can select on this flag and can see which (HTML) content is most consumed by our visitors



Value	Proportion	%	Count
/ksc.html		12.69	43673
/shuttle/missions/sts-69/mission-sts-69.html		7.15	24604
/shuttle/missions/missions.html		6.52	22442
/software/winvn/winvn.html		3.01	10342
/history/history.html		2.94	10128
/history/apollo/apollo.html		2.61	8984
/shuttle/countdown/liftoff.html		2.29	7864
/history/apollo/apollo-13/apollo-13.html		2.09	7176
/shuttle/technology/sts-newsref/stsref-toc.html		1.89	6517
/shuttle/missions/sts-69/images/images.html		1.53	5263
/shuttle/missions/sts-69/movies/movies.html		1.41	4846
/shuttle/missions/sts-69/liftoff.html		1.32	4558
/facilities/lc39a.html		1.3	4461
/shuttle/resources/orbiters/endeavour.html		1.29	4434
/shuttle/missions/sts-70/mission-sts-70.html		1.18	4065
/shuttle/technology/sts-newsref/sts_asm.html		1.08	3706
/shuttle/countdown/countdown.html		1.02	3518
/shuttle/missions/sts-71/movies/movies.html		1.02	3507

And from here we can analyse visitor behaviour in a more algorithmic way. Looking at common **sequences** and common **clusters** of content (which would reveal segments of visitor behaviour)

String Cleaner

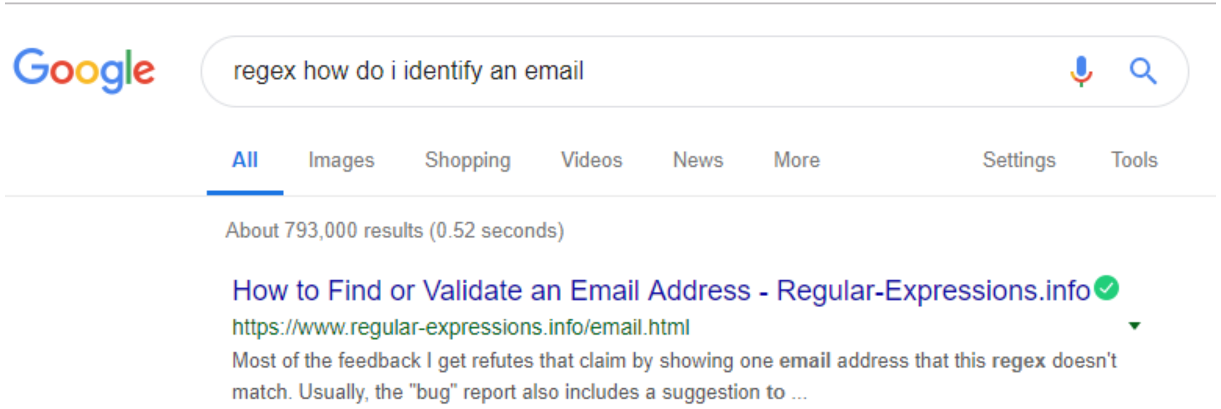
All RX nodes are scriptable

The RX nodes can be (Python) scripted/automated in the familiar style…

```
node = modeler.script.stream().createAt("regexp_cleaner", u"String Cleaner", 512, 192)
node.setPropertyValue("clean_fields", [u"HomePhone", u"MobilePhone"])
node.setPropertyValue("output_suffix", u"_processed")
node.setPropertyValue("trim_mode", u"both")
node.setPropertyValue("replace_tabs", True)
node.setPropertyValue("replace_duplicate_blanks", True)
node.setPropertyValue("capitalize_mode", u"none")
node.setPropertyValue("find_upper_english_chars", False)
node.setPropertyValue("find_lower_english_chars", False)
node.setPropertyValue("find_digits", True)
node.setPropertyValue("find_punctuation", False)
node.setPropertyValue("find_blanks", False)
node.setPropertyValue("find_spaces", False)
node.setPropertyValue("find_non_printing_chars", False)
node.setPropertyValue("categories_mode", u"keep")
```

Getting help

- Regex examples are very easy to find …



regex how do i identify an email

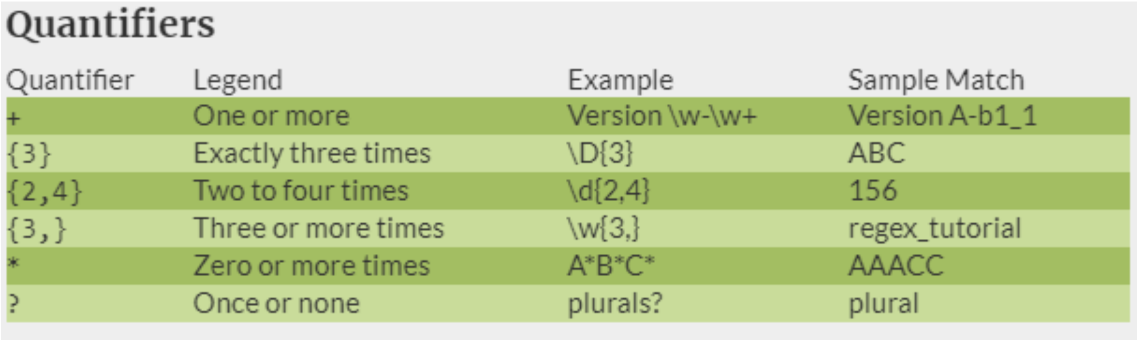
About 793,000 results (0.52 seconds)

How to Find or Validate an Email Address - Regular-Expressions.info

https://www.regular-expressions.info/email.html

Most of the feedback I get refutes that claim by showing one **email** address that this **regex** doesn't match. Usually, the "bug" report also includes a suggestion to ...

- https://www.rexegg.com/regex-quickstart.html

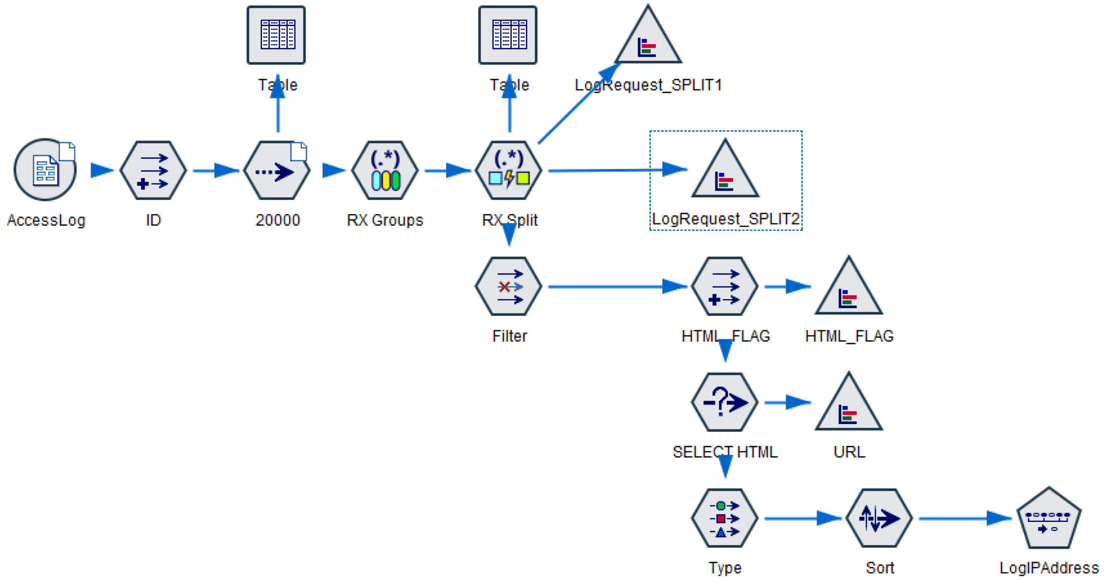


Quantifiers

Quantifier	Legend	Example	Sample Match
+	One or more	Version \w-\w+	Version A-b1_1
{3}	Exactly three times	\D{3}	ABC
{2,4}	Two to four times	\d{2,4}	156
{3,}	Three or more times	\w{3,}	regex_tutorial
*	Zero or more times	A*B*C*	AACC
?	Once or none	plurals?	plural

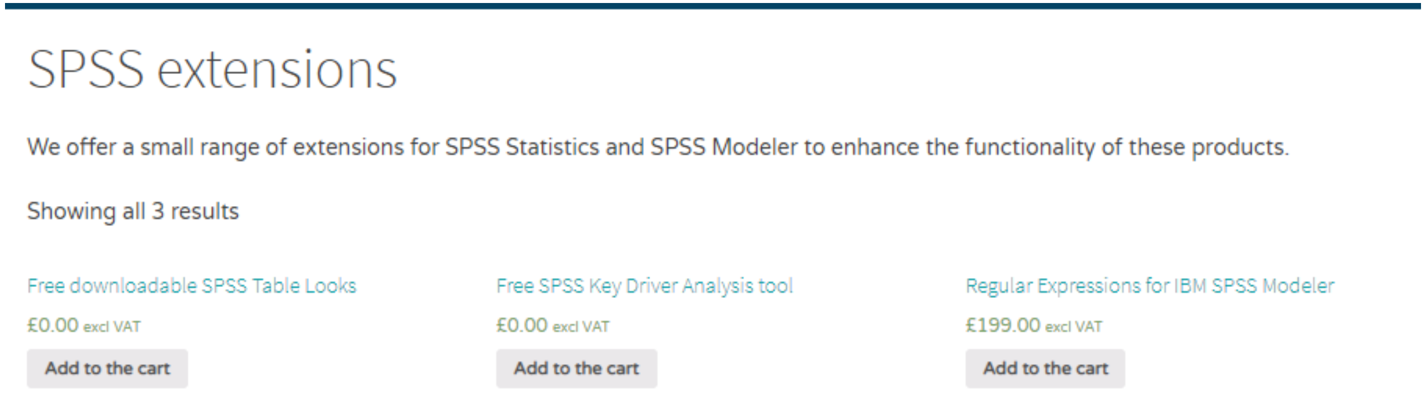
In Summary

- The Regex nodes significantly extend the string handling capability *within* Modeler
- True to the Modeler ethos they are accessible to users who have little or no experience of regular expressions
- And even for those who do they integrate advanced text manipulation in the flow of a Modeler stream



Extensions extending

- Smart Vision is planning to develop more extensions for both SPSS Modeler and SPSS Statistics
  - We have a v1.2 update to the RX nodes to extend UNICODE support and some fixes
  - A new, enhanced, **Metadata node**
  - Check in to https://www.sv-europe.com/product-category/spss-extensions/ to see more
- Please do mail us to suggest/request other extensions
  - info@sv-europe.com



SPSS extensions

We offer a small range of extensions for SPSS Statistics and SPSS Modeler to enhance the functionality of these products.

Showing all 3 results

Free downloadable SPSS Table Looks	Free SPSS Key Driver Analysis tool	Regular Expressions for IBM SPSS Modeler
£0.00 excl VAT	£0.00 excl VAT	£199.00 excl VAT
Add to the cart	Add to the cart	Add to the cart

Working with Smart Vision Europe Ltd.

- **Sourcing Software**
  - You can buy your analytical software from us often with discounts
  - Assist with selection, pilot, implementation & support of analytical tools
  - http://www.sv-europe.com/buy-spss-online/
- **Training and Consulting Services**
  - Guided consulting & training to develop in house skills
  - Delivery of classroom training courses / side by side training support
  - Identification & recruitment of analytical skills into your organisation
- **Advice and Support**
  - offer ‘no strings attached’ technical and business advice relating to analytical activities
  - Technical support services around SPSS

What next?

The RX nodes are available at:

https://www.sv-europe.com/product/regular-expressions-ibm-spss-modeler/

The free, enhanced, Metadata node at:

https://www.sv-europe.com/product/enhanced-metadata-node-for-ibm-spss-modeler/

If you are new to IBM/SPSS Modeler we have an Intro Course at:

https://www.sv-europe.com/smart-vision-spss-courses/introduction-spss-modeler/

SMARTVISION
Europe

Contact us:

+44 (0)207 786 3568
info@sv-europe.com
Twitter: @sveurope
Follow us on LinkedIn
Sign up for our Newsletter

Thank you