THE PEAK OF DATA INTEGRATION
2022 UC

Attribute Alchemy

Advanced Attribute and List Handling

SAFE SOFTWARE

Mark Stoakes

Manager, FME Application experts

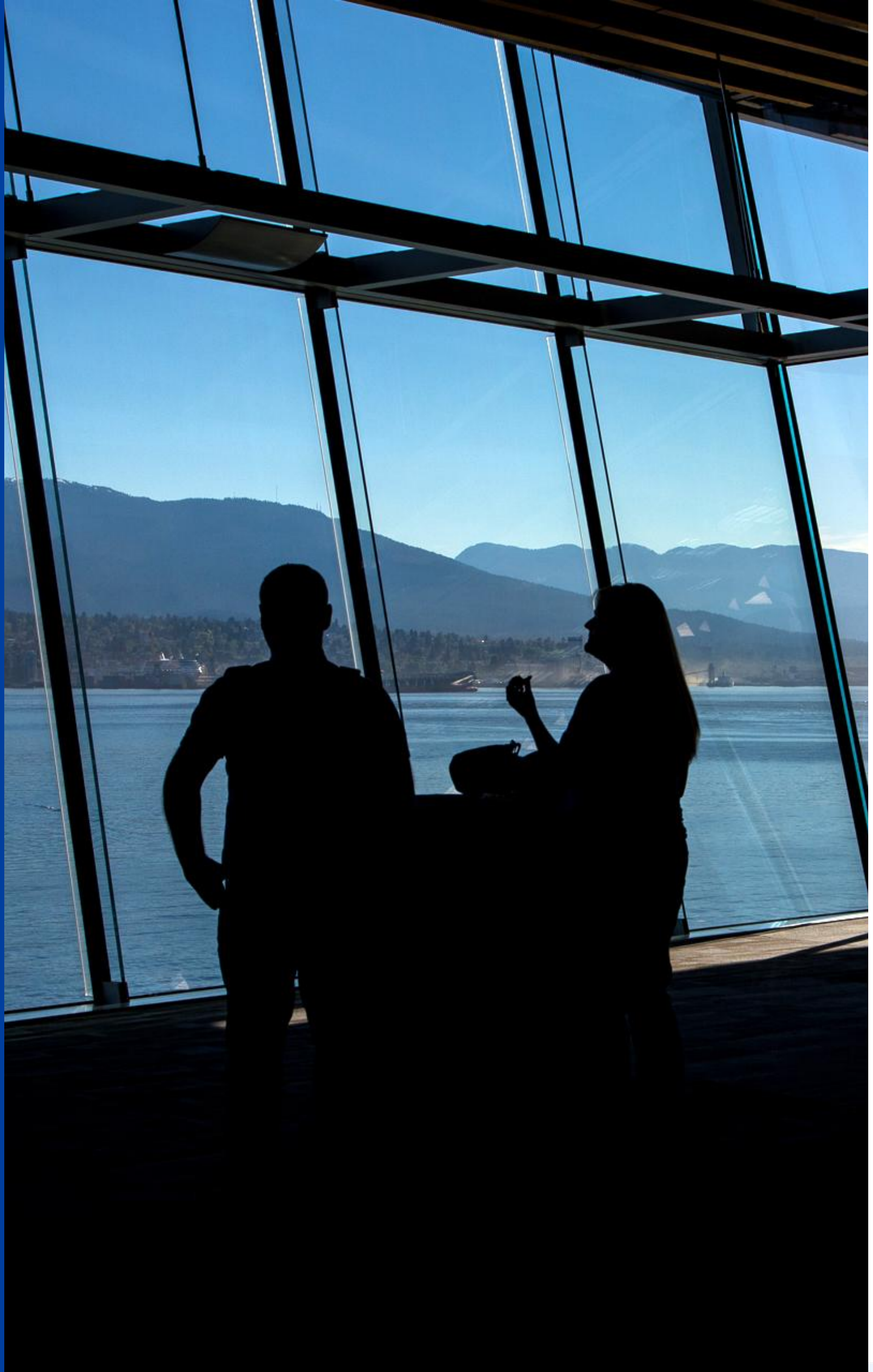
Andrea Eisma

Technical Support Team Lead, FME Desktop

Start Your Journey

Please request a virtual machine:

HTTP: http://fme.ly/UC22Training
User: Administrator
Pass: FMElearnings#1



Agenda

We're going to cover two broad topics:

- Part I: Advanced Attribute Manipulation
  - Date/Time in FME
  - Finding Patterns with Regex
  - Comparing Adjacent Features
  - Name / Value Pairs
  - The Hidden String Functions
  - Bulk Transformers
- Part II: FME List Attributes
  - What are List Attributes
  - Creating List Attributes
  - Working with List Attributes
  - A little bit of Python

Training Resources

- Virtual machine
- Training Data
  - `C:\FMEData-2022\Resources\FMEUC22\Advanced Attribute and List Handling`
- These slides (as PDF)
- Tutorial: Getting Started with List Attributes
  - Creating Lists Using Transformers
  - Transformers for Working with List Attributes

20
22

FME User Conference

Part I: Advanced Attribute Manipulation

Dates & Times in FME

Dates and Times are Complicated
Formatting?

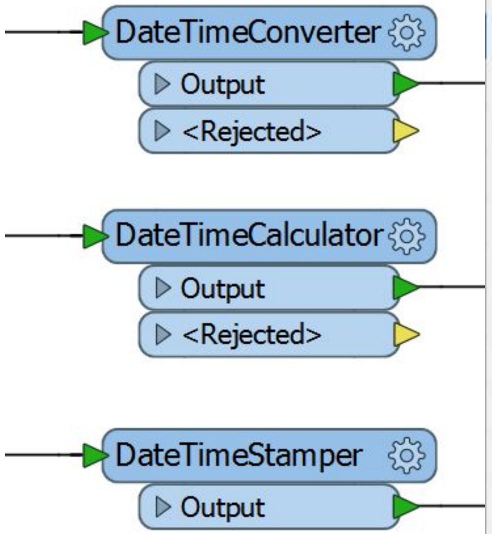
- <d v="2016-12-13T00:00:00"/> *(Excel)*
- YYMMDD, HHMMSS, UTC
- Jun 2016
- 'on Saturday, Jan 9th 2016, 01:00 am' & '+0530'
- 2016-12-07 12:20:07.785403-05
- YYYY-MM-DD hh:mm:ss[.nnnnnnn] *(SQL Server 'datetime2' value)*
- **YYYYMMDDHHMMSS.SS *(FME)***

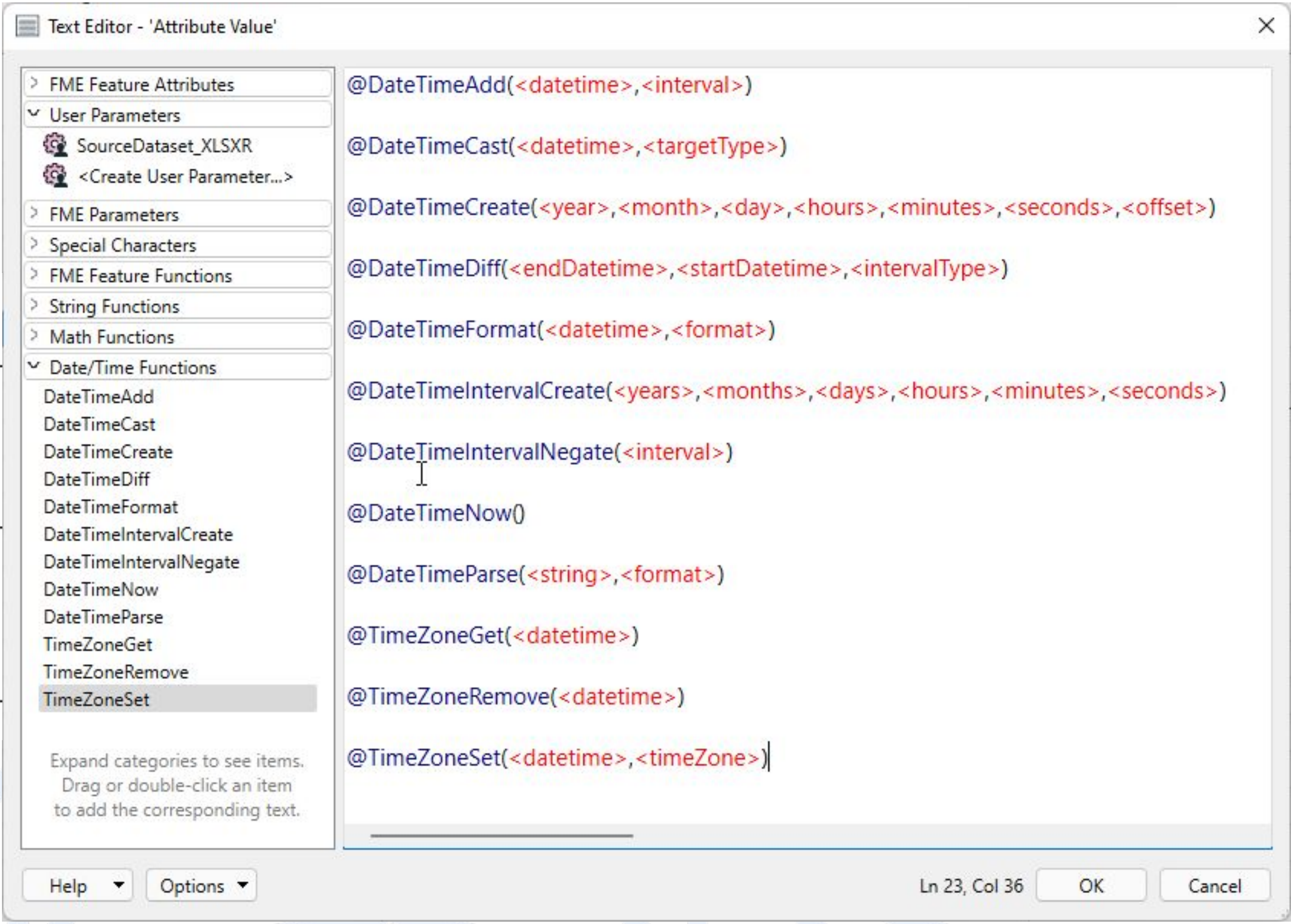
Calculations?

- FME Date / Time functions
  - Date2-Date1 = How many days?

Date Time Tools

Transformers & Functions



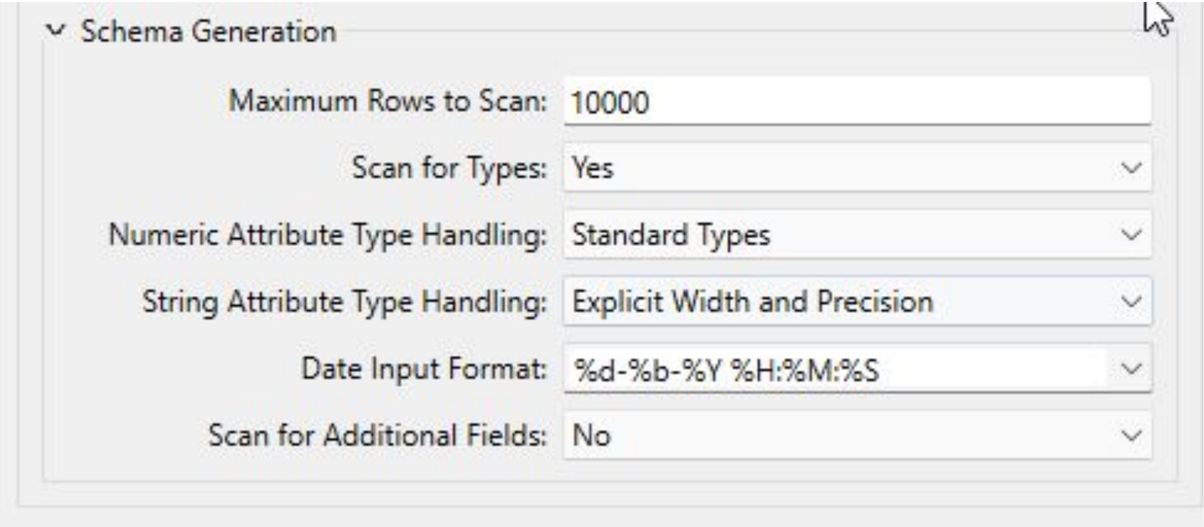


Expand categories to see items. Drag or double-click an item to add the corresponding text.

```
@DateTimeAdd(<datetime>,<interval>)

@DateTimeCast(<datetime>,<targetType>)

@DateTimeCreate(<year>,<month>,<day>,<hours>,<minutes>,<seconds>,<offset>)

@DateTimeDiff(<endDatetime>,<startDatetime>,<intervalType>)

@DateTimeFormat(<datetime>,<format>)

@DateTimeIntervalCreate(<years>,<months>,<days>,<hours>,<minutes>,<seconds>)

@DateTimeIntervalNegate(<interval>)

@DateTimeNow()

@DateTimeParse(<string>,<format>)

@TimeZoneGet(<datetime>)

@TimeZoneRemove(<datetime>)

@TimeZoneSet(<datetime>,<timeZone>)
```

Date Time in CSV

CSV Reader can parse dates:



Schema Generation

Maximum Rows to Scan: 10000

Scan for Types: Yes

Numeric Attribute Type Handling: Standard Types

String Attribute Type Handling: Explicit Width and Precision

Date Input Format: %d-%b-%Y %H:%M:%S

Scan for Additional Fields: No

... to create the FME formatted date and a date/time data type

Why might it be better to sort out the dates in the CSV reader rather then, say, DateTimeConverter?

Gotcha: FME won’t recognise the FME date time in a SQL query - use the ***native*** date time format, i.e.: "2021-12-01"

POP TEST

Date & Time

Finding patterns: regex

Most regex patterns have been solved by someone!

Problem: extract the first three letters:

Code ABD3705337067

Regular Expression: ([A-Z]{3})([0-9]+)

Use the FME Regex editor

Replacement text: **\1** - replaces with the sub-expression

regex is great - but there are many FME string function alternatives if you're not familiar with regex…

String Functions:

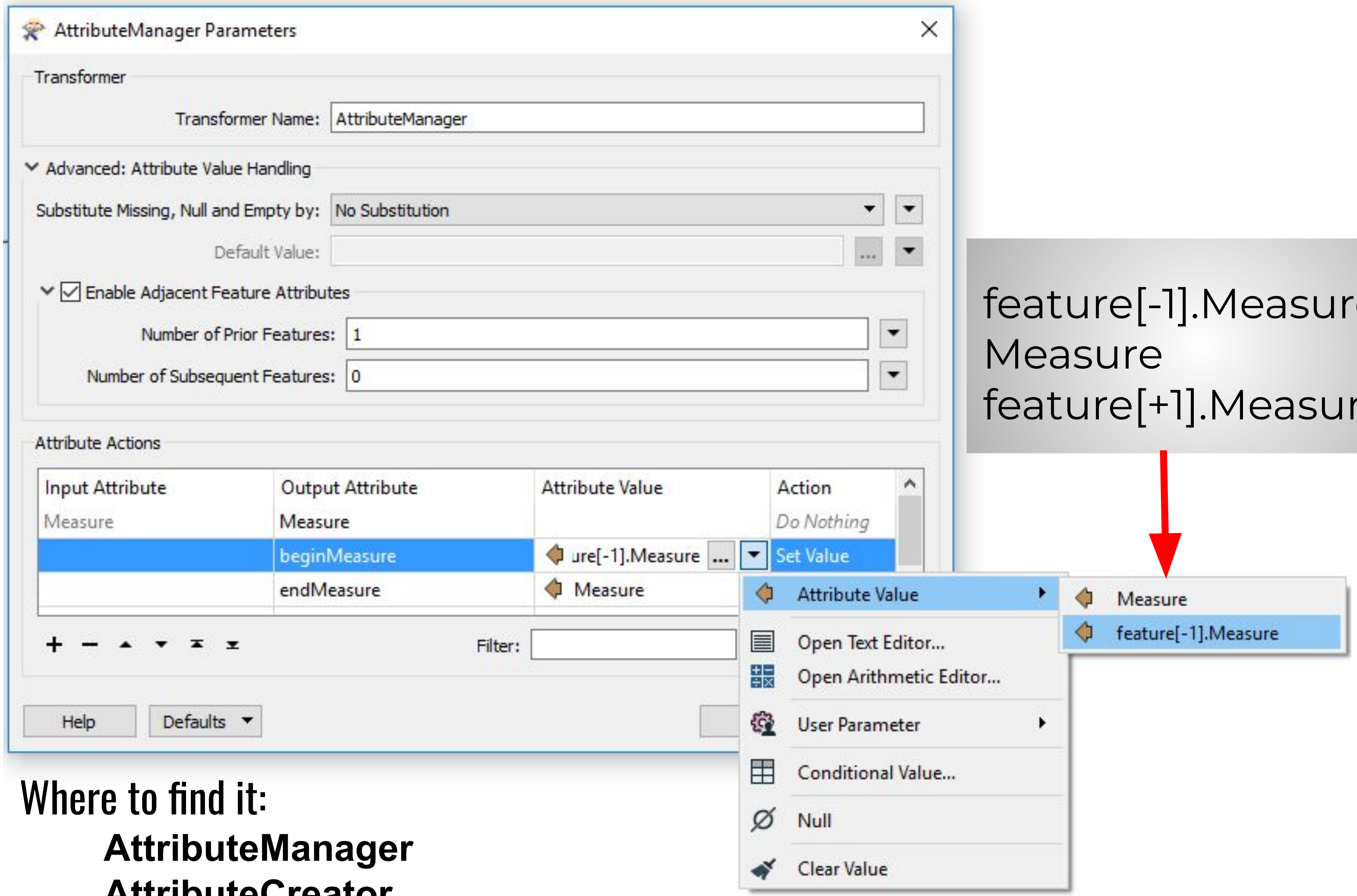
Attribute	String Function
alpha	@Left(@Value(Code),3)
beta	@Substring(@Value(Code),3,-1)

Fun regex learning website: https://regexr.com/

Past, Present & Future Attributes

Adjacent Attributes

- Great for sequenced or ordered data



AttributeManager Parameters

Transformer

Transformer Name: AttributeManager

Advanced: Attribute Value Handling

Substitute Missing, Null and Empty by: No Substitution

Default Value:

☑ Enable Adjacent Feature Attributes

Number of Prior Features: 1

Number of Subsequent Features: 0

Attribute Actions

Input Attribute	Output Attribute	Attribute Value	Action
Measure	Measure		Do Nothing
	beginMeasure	ure[-1].Measure	Set Value
	endMeasure	Measure	

Filter:

feature[-1].Measure
Measure
feature[+1].Measure

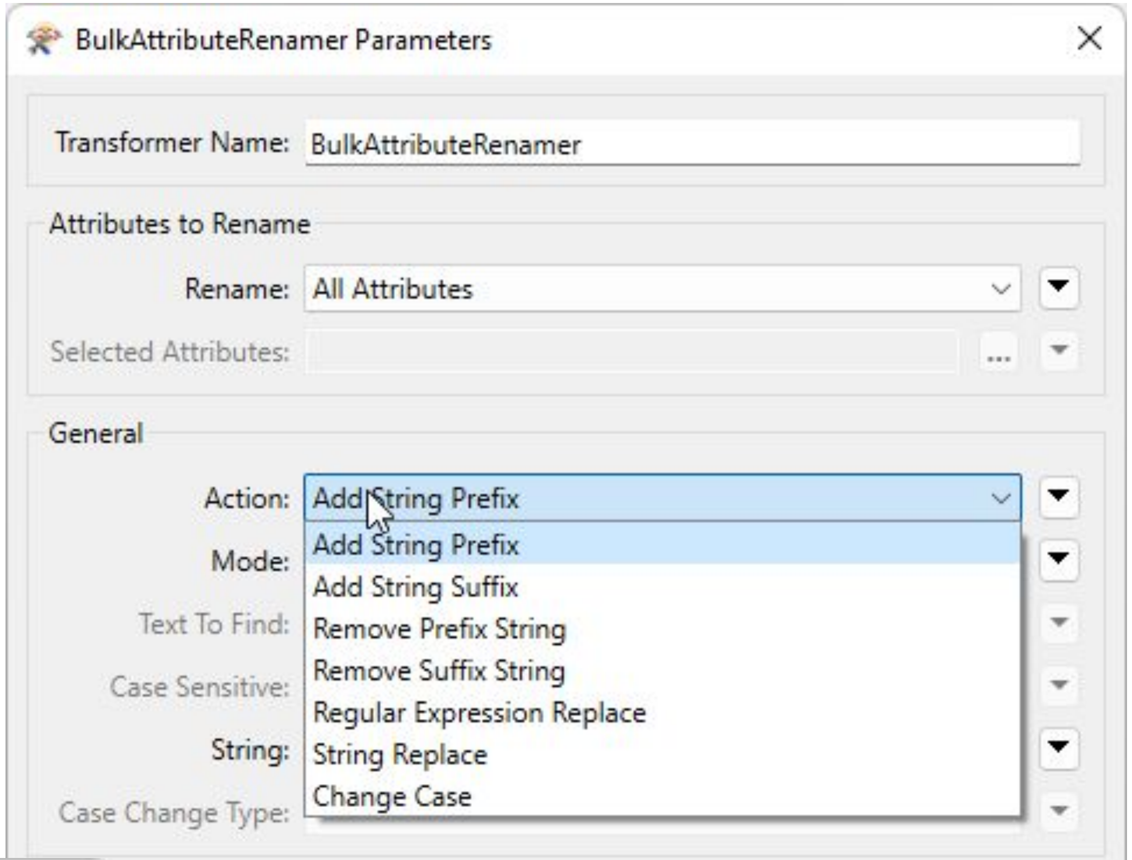
Where to find it:

AttributeManager
AttributeCreator

Bulk-up

Manipulate Attribute Names

- BulkAttributeRenamer
  - Prefix,Suffix
  - Case Change
  - Regex
- BulkAttributeRemover
  - Regex for attribute cleanup



BulkAttributeRenamer Parameters

Transformer Name: BulkAttributeRenamer

Attributes to Rename

Rename: All Attributes

Selected Attributes:

General

Action: Add String Prefix

- Add String Prefix
- Add String Suffix
- Remove Prefix String
- Remove Suffix String
- Regular Expression Replace
- String Replace
- Change Case

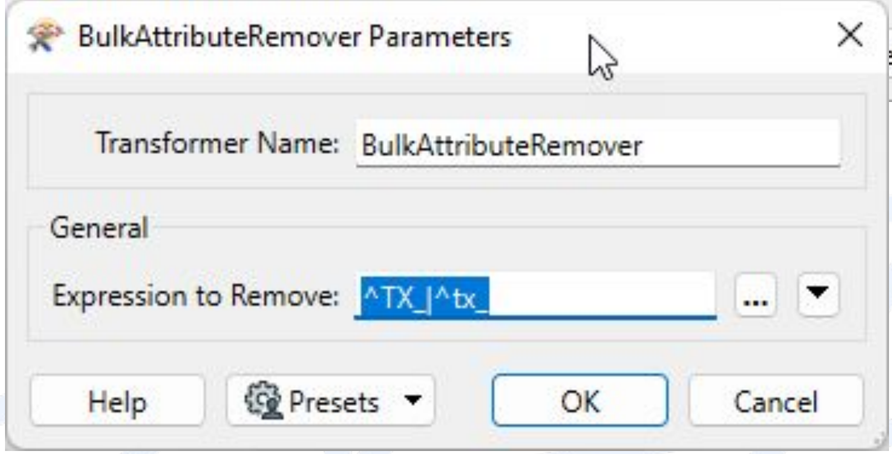
Mode:

Text To Find:

Case Sensitive:

String:

Case Change Type:



BulkAttributeRemover Parameters

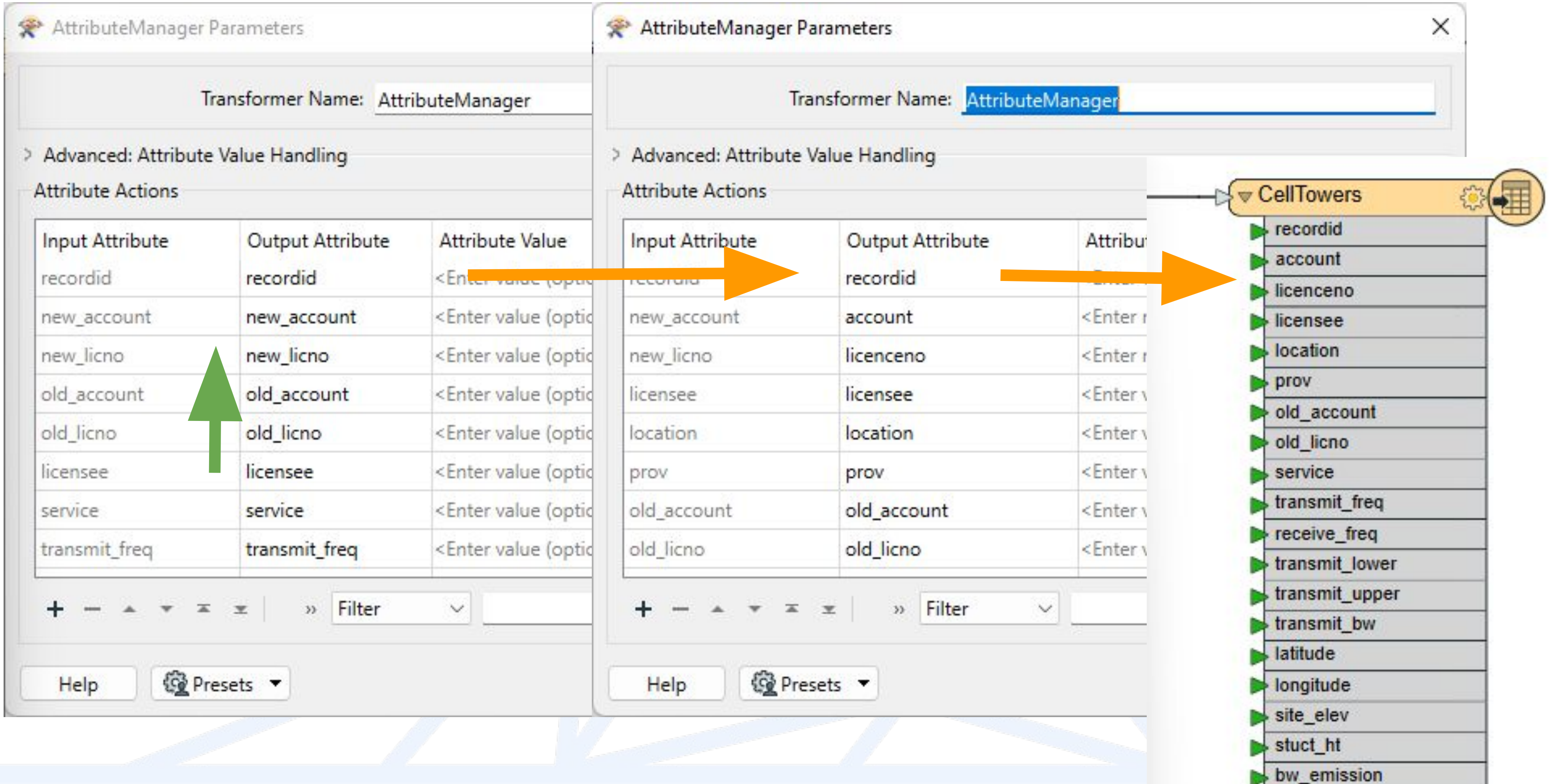
Transformer Name: BulkAttributeRemover

General

Expression to Remove: ^TX_|^tx_

AttributeManager

AttributeManager can reorder attributes



AttributeManager Parameters

Transformer Name: AttributeManager

› Advanced: Attribute Value Handling

Attribute Actions

Input Attribute	Output Attribute	Attribute Value
recordid	recordid	<Enter value (optic
new_account	new_account	<Enter value (optic
new_licno	new_licno	<Enter value (optic
old_account	old_account	<Enter value (optic
old_licno	old_licno	<Enter value (optic
licensee	licensee	<Enter value (optic
service	service	<Enter value (optic
transmit_freq	transmit_freq	<Enter value (optic

AttributeManager Parameters

Transformer Name: AttributeManager

› Advanced: Attribute Value Handling

Attribute Actions

Input Attribute	Output Attribute	Attribu
recordid	recordid	<Enter
new_account	account	<Enter
new_licno	licenceno	<Enter
licensee	licensee	<Enter
location	location	<Enter
prov	prov	<Enter
old_account	old_account	<Enter
old_licno	old_licno	<Enter

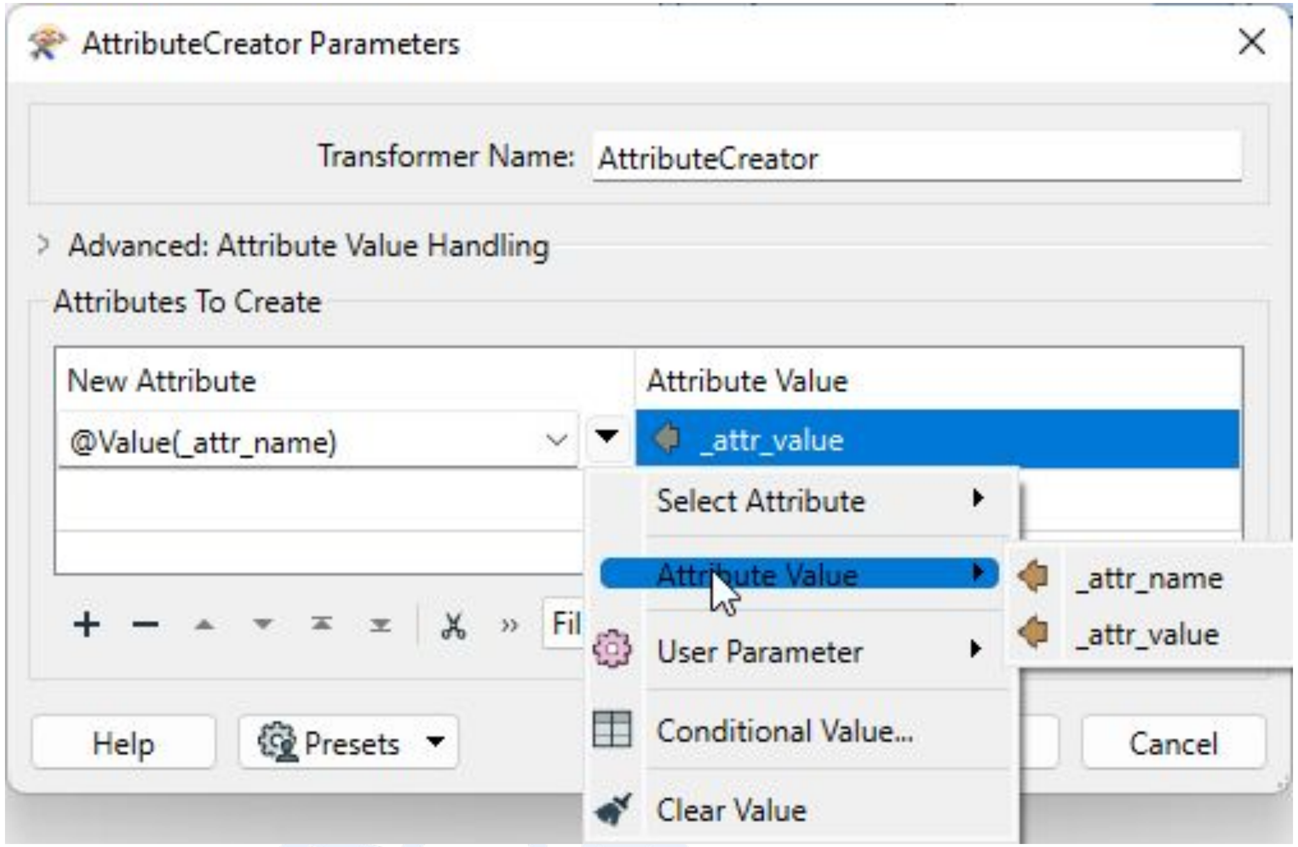
CellTowers

- recordid
- account
- licenceno
- licensee
- location
- prov
- old_account
- old_licno
- service
- transmit_freq
- receive_freq
- transmit_lower
- transmit_upper
- transmit_bw
- latitude
- longitude
- site_elev
- stuct_ht
- bw_emission

Attribute Value

The ***value*** of an attribute has the ***name*** of the attribute you want to reference…

```
    attribute_name = UserName
```



AttributeCreator Parameters

Transformer Name: AttributeCreator

Advanced: Attribute Value Handling

Attributes To Create

New Attribute	Attribute Value
@Value(_attr_name)	_attr_value

- often used after AttributeExploder

EXERCISE

Attribute Handling

Continue along the road of learning

Learning Resources:

- Trailhead: Transform Attributes
- World Tour: Time Machines & Attribute Alchemy
- Performance: Comparative Performance Test
- Tutorial: Advanced Attribute and List Handling

Part II: List Attributes

List Attributes

List attributes allow an attribute to have more than one value.

- `FRUITS = “Apple, Orange, Pear”`

  vs. simple list:

- `Fruits{0} = Apple`

  `Fruits{1} = Orange`

  `Fruits{2} = Pear`

  OR complex list:

- `Fruits{0}.name           = Apple`
- `Fruits{0}.scientificName = Malus domestica`
- `Fruits{1}.name           = Orange`
- `Fruits{1}.scientificName = Citrus X sinensis`

Tutorial: [Getting Started with List Attributes]

List Attributes

Tutorial: Getting Started with List Attributes

- Creating Lists Using Transformers
  - example workspace: **park_trees_final.fmw**
- Transformers for Working with List Attributes
  - example workspace: **ListTransformerExamples.fmw**

EXERCISE

List Handling

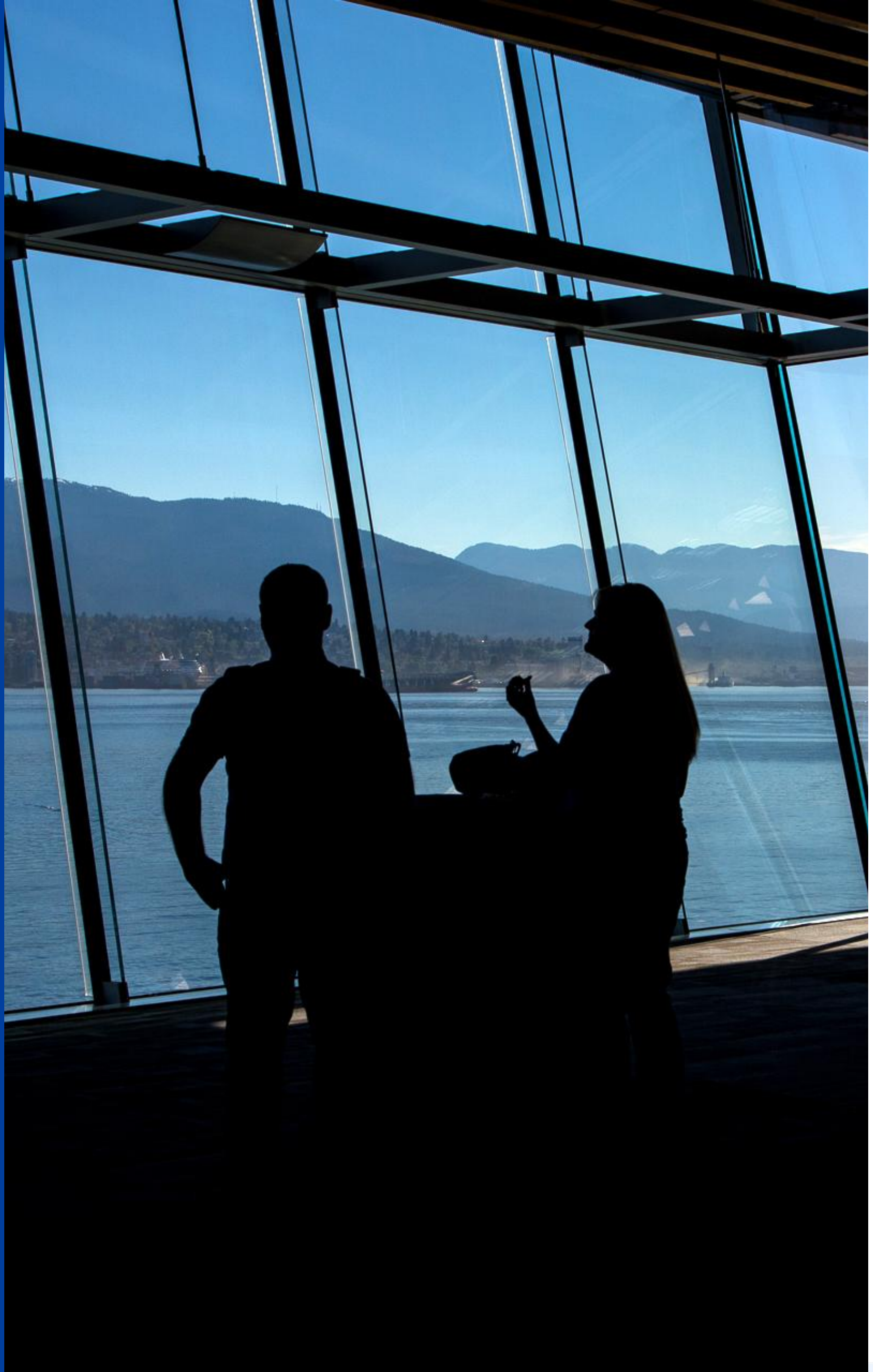
Continue along the road of learning

Learning Resources:

- Tutorial: Getting Started with List Attributes
- Trailhead: Working with Multiple Data Models Using Lists
- Documentation: About List Attributes
- FME UC 2017: List Manipulation in FME

  by Joanna Hobbins, CAE
- Python: Python List Tutorial

Wrap up



Summary

We covered two broad topics:

- Advanced Attribute Manipulation
  - Date/Time in FME
  - Finding Patterns with Regex
  - Comparing Adjacent Features
  - Name / Value Pairs
  - The Hidden String Functions
  - Bulk Transformers
- FME List Attributes
  - What are List Attributes
  - Creating List Attributes
  - Working with List Attributes
  - A little bit of Python

Next Steps

- Review some of the resources we listed
- Continue to expand you FME knowledge with the Trailhead learning resource

THE PEAK OF DATA INTEGRATION
2022 UC

Thank You!

andrea.eisma@safe.com

mark.stoakes@safe.com

SAFE SOFTWARE®