# Introduction to Spatial Data Processing using FME and Python

www.inser.ch

# Hello

**Eulalie** – ETH engineer
FME Certified Professional

**David** – Computer engineer
FME Certified Professional
FME Server certified Professional

**Régis** – ETH engineer
FME Certified Professional
FME Certified Trainer

**Marion** – Student in Topography
Intern

# Workshop structure

1. Brief FME introduction
2. FME – Python theory
3. Exercises :
    1. ACLED exercise
    2. PTP exercise
    3. Syria exercise
    4. DEM exercise

Geopol.ch

Get your Remote desktop credentials at :

**https://www.geopol.ch/#/en/workspaces/74**

Or download the data :

**https://goo.gl/Y5TyhT**

On your desktop :

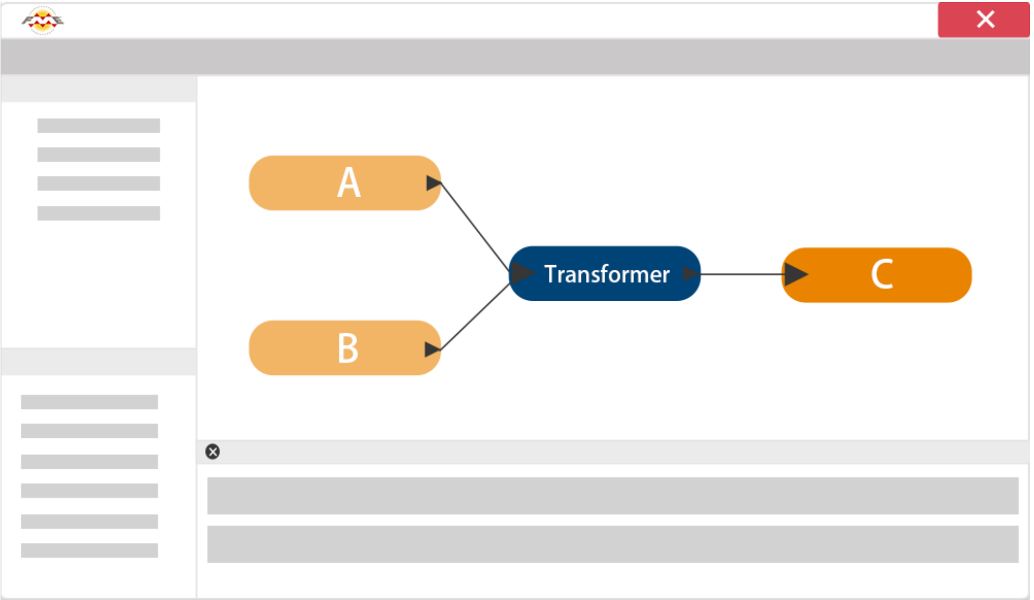
# Workshop structure

To begin, just a few quick questions :

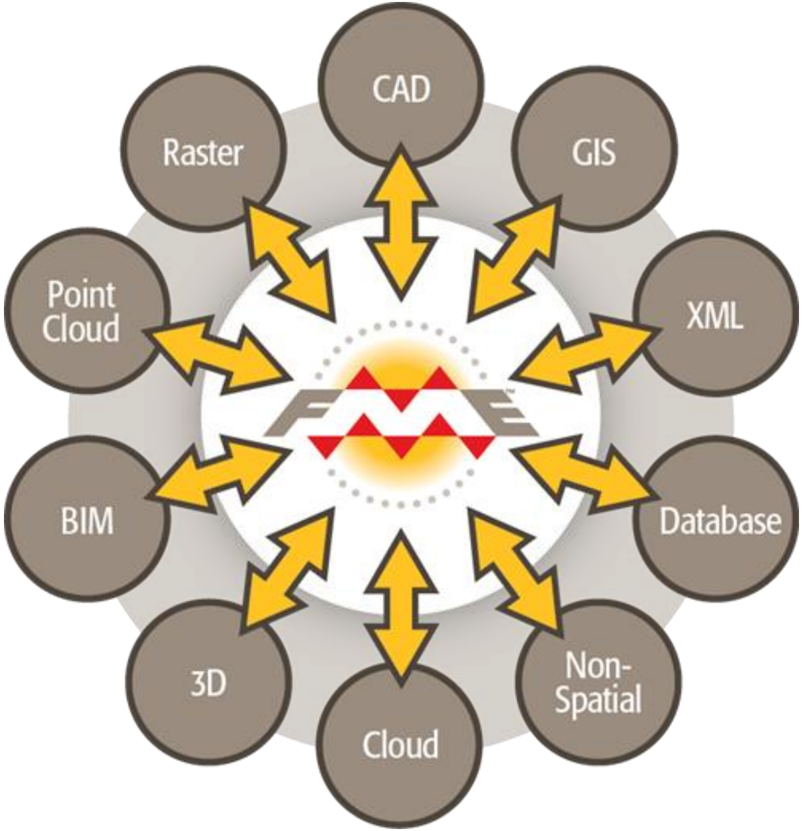
- Who has never used FME and/or Python before ?
- Who has some basic knowledge of FME

# FME, very briefly...

- Data Interoperability Solution
- Extract, Transform and Load

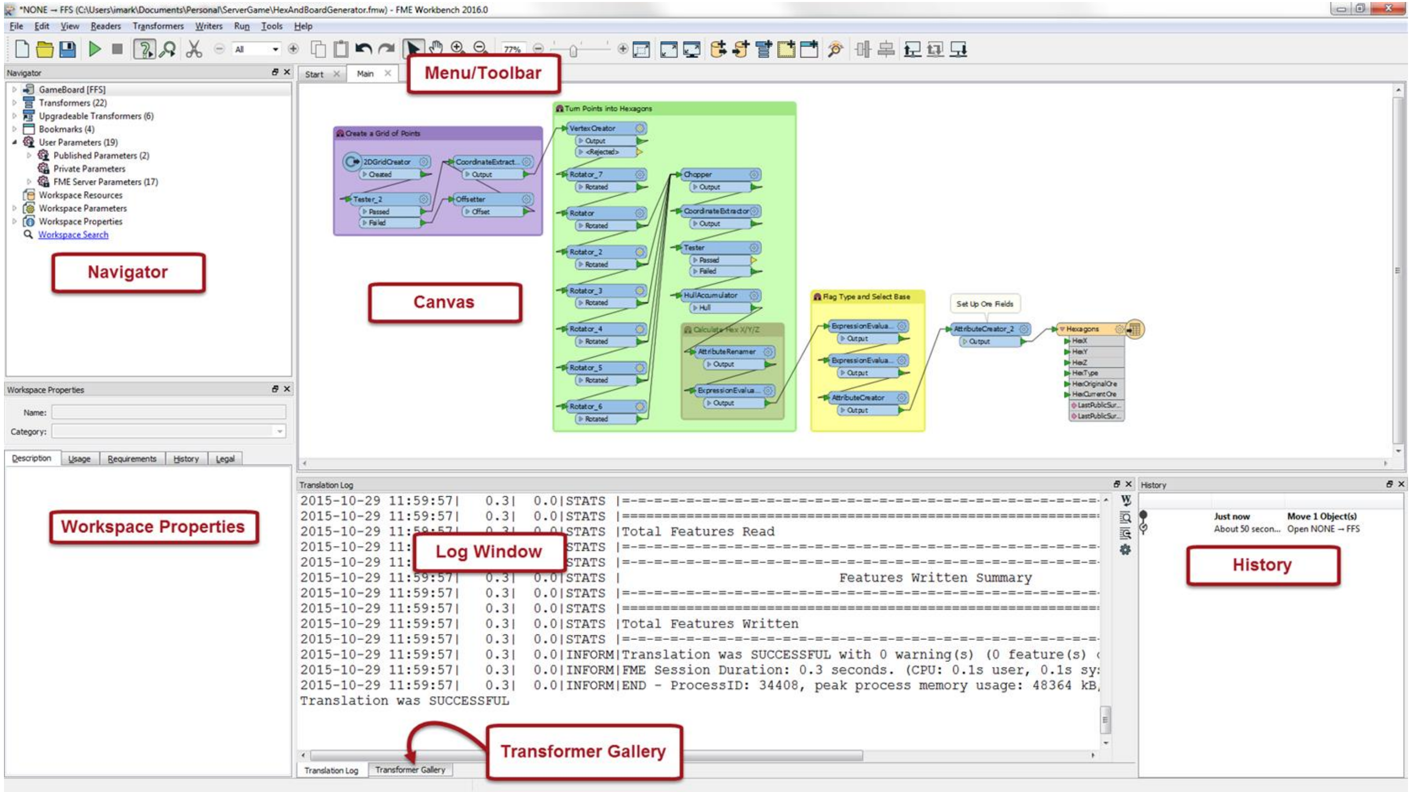


# FME, very briefly…



- Rich Data Model

# FME, very briefly…



```
2015-10-29 11:59:57|   0.3|   0.0|STATS |=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
2015-10-29 11:59:57|   0.3|   0.0|STATS |=========================================================
2015-10-29 11:59:57|   0.3|   0.0|STATS |Total Features Read
2015-10-29 11:                    STATS |=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
2015-10-29 11:                    STATS |=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
2015-10-29 11:59:57|   0.3|   0.0|STATS |                           Features Written Summary
2015-10-29 11:59:57|   0.3|   0.0|STATS |=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
2015-10-29 11:59:57|   0.3|   0.0|STATS |=========================================================
2015-10-29 11:59:57|   0.3|   0.0|STATS |Total Features Written
2015-10-29 11:59:57|   0.3|   0.0|STATS |=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
2015-10-29 11:59:57|   0.3|   0.0|INFORM|Translation was SUCCESSFUL with 0 warning(s) (0 feature(s) 
2015-10-29 11:59:57|   0.3|   0.0|INFORM|FME Session Duration: 0.3 seconds. (CPU: 0.1s user, 0.1s sy
2015-10-29 11:59:57|   0.3|   0.0|INFORM|END - ProcessID: 34408, peak process memory usage: 48364 kB,
Translation was SUCCESSFUL
```

# Python

A friendly message from Safe

From the help text of the PythonCaller:

*“Using Python to perform arbitrary operations on features is a powerful aspect of Workbench.*

*However, the logic introduced into a workspace is less visible and can therefore be more difficult to maintain than logic built using Workbench’s built-in transformers.*

*It is recommended that other transformers be used when possible instead of Python scripts.”*

# Where to write python script ?

## At the beginning

Scripted parameters

## Before the process

Startup Python script

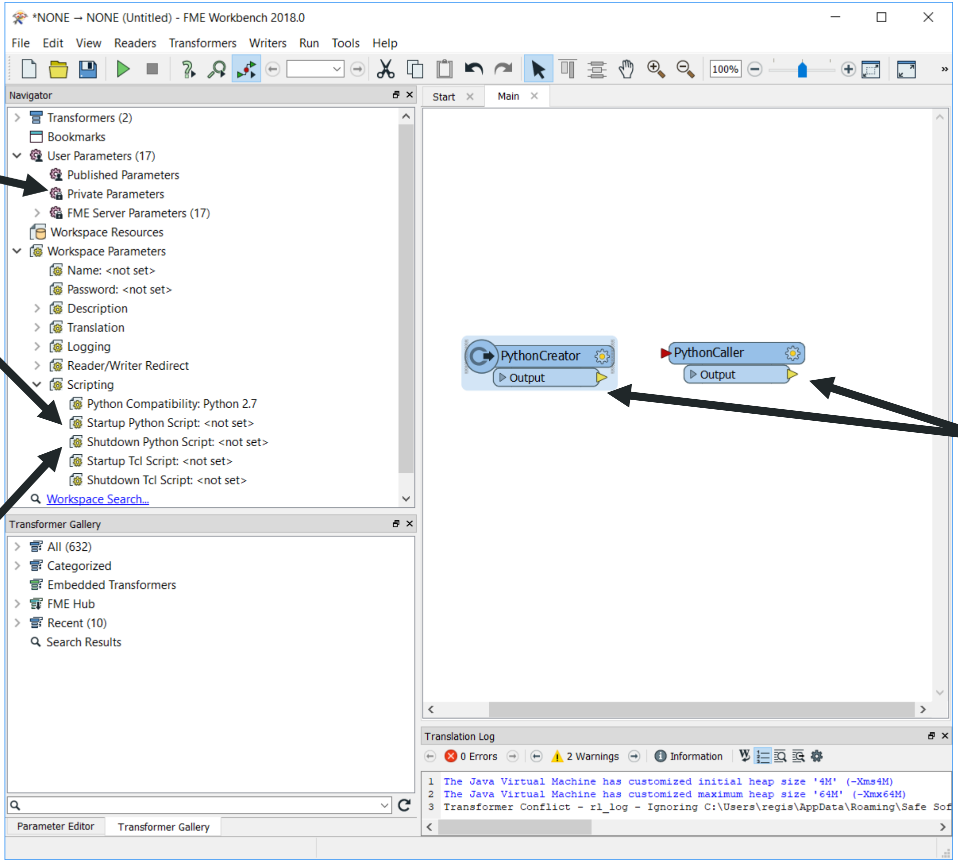
## After the process

Shutdown Python script

## During the process

PythonCreator

PythonCaller



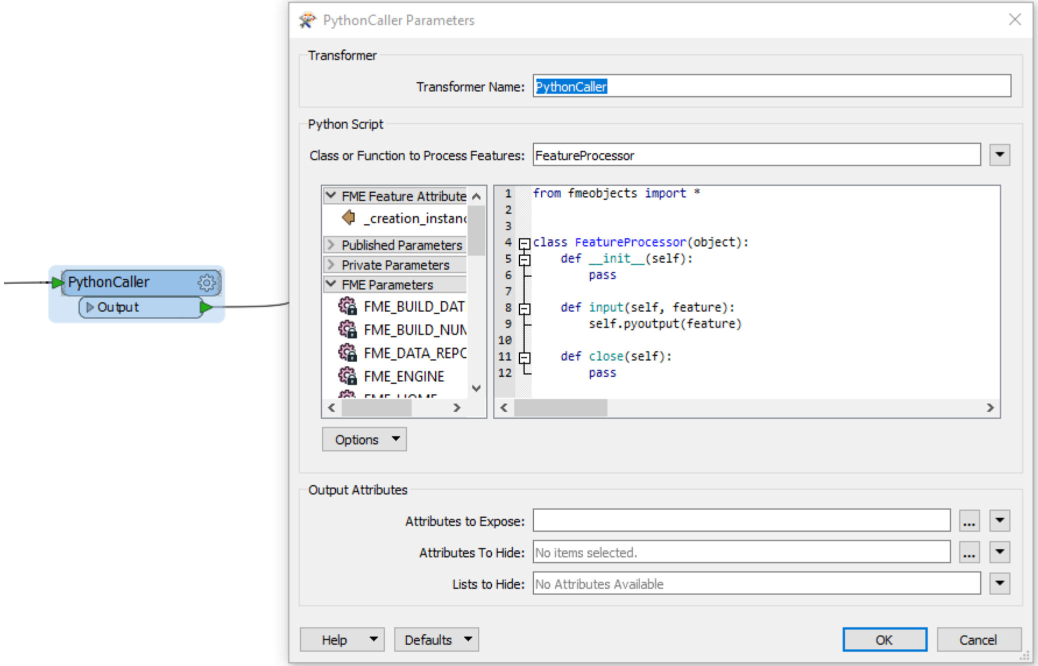
Transformers (2)
Bookmarks
User Parameters (17)
Published Parameters
Private Parameters
FME Server Parameters (17)
Workspace Resources
Workspace Parameters
Name: <not set>
Password: <not set>
Description
Translation
Logging
Reader/Writer Redirect
Scripting
Python Compatibility: Python 2.7
Startup Python Script: <not set>
Shutdown Python Script: <not set>
Startup Tcl Script: <not set>
Shutdown Tcl Script: <not set>
Workspace Search...

```
1 The Java Virtual Machine has customized initial heap size '4M' (-Xms4M)
2 The Java Virtual Machine has customized maximum heap size '64M' (-Xmx64M)
3 Transformer Conflict - rl_log - Ignoring C:\Users\regis\AppData\Roaming\Safe Sof
```

# PythonCaller

# ACLED data

# Ex1 – PythonCaller - ACLED



Transformer Name: PythonCaller

Class or Function to Process Features: FeatureProcessor

```
from fmeobjects import *


class FeatureProcessor(object):
    def __init__(self):
        pass

    def input(self, feature):
        self.pyoutput(feature)

    def close(self):
        pass
```

Attributes to Expose:

Attributes To Hide: No items selected.

Lists to Hide: No Attributes Available

- Manipulate features using a Python script
- Called once for each feature that passes through
- Supports two models:
  1. Procedure interface: quick and simple, for processing one feature at a time.
  2. Class interface: powerful and flexible, includes startup and shutdown.
- Example: using the Python module textwrap to word-wrap a
long string attribute into an FME list attribute

# Ex1 – PythonCaller - ACLED

ACLED

Bringing clarity to crisis

**Armed Conflict Location & Event Data Project (ACLED) is a disaggregated conflict collection, analysis and crisis mapping project.**

**ACLED collects the dates, actors, types of violence, locations, and fatalities of all reported political violence and protest events across Africa, South Asia, South East Asia and the Middle East.**

# Ex1 – PythonCaller - ACLED

ACLED data can be accessed via an API

**https://api.acleddata.com/acled/read?year=@Value(year)**

However, it only downloads 500 features. To access the whole dataset, the page argument must be added

**https://api.acleddata.com/acled/read?year=@Value(year)&page=@Value(page)**

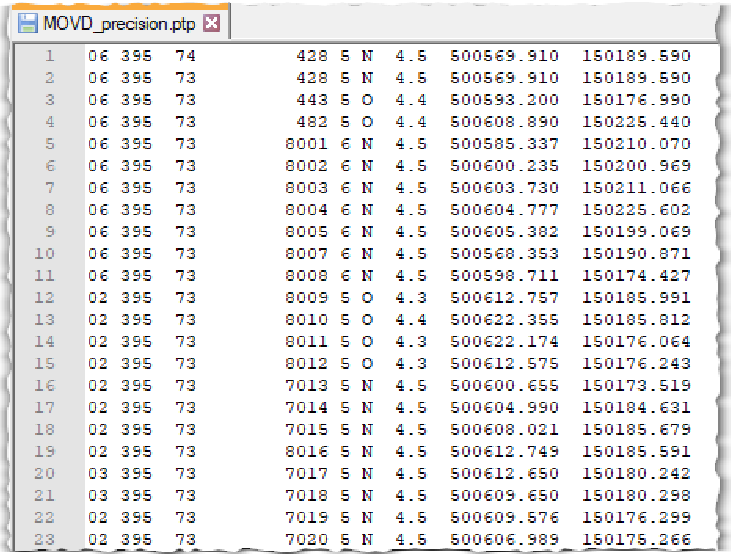
# Ex1 – PythonCaller - ACLED

Log into your computer → Read the doc for exercise 1 → Explore the FME workspace → Follow the steps to finish the exercise

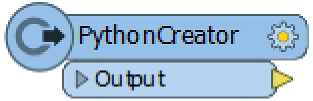
PythonCreator

# Custom ptp reader

# Ex2 – Custom ptp reader



```
06 395  74         428 5 N  4.5  500569.910  150189.590
06 395  73         428 5 N  4.5  500569.910  150189.590
06 395  73         443 5 O  4.4  500593.200  150176.990
06 395  73         482 5 O  4.4  500608.890  150225.440
06 395  73        8001 6 N  4.5  500585.337  150210.070
06 395  73        8002 6 N  4.5  500600.235  150200.969
06 395  73        8003 6 N  4.5  500603.730  150211.066
06 395  73        8004 6 N  4.5  500604.777  150225.602
06 395  73        8005 6 N  4.5  500605.382  150199.069
06 395  73        8007 6 N  4.5  500568.353  150190.871
06 395  73        8008 6 N  4.5  500598.711  150174.427
02 395  73        8009 5 O  4.3  500612.757  150185.991
02 395  73        8010 5 O  4.4  500622.355  150185.812
02 395  73        8011 5 O  4.3  500622.174  150176.064
02 395  73        8012 5 O  4.3  500612.575  150176.243
02 395  73        7013 5 N  4.5  500600.655  150173.519
02 395  73        7014 5 N  4.5  500604.990  150184.631
02 395  73        7015 5 N  4.5  500608.021  150185.679
02 395  73        8016 5 N  4.5  500612.749  150185.591
03 395  73        7017 5 N  4.5  500612.650  150180.242
03 395  73        7018 5 N  4.5  500609.650  150180.298
02 395  73        7019 5 N  4.5  500609.576  150176.299
02 395  73        7020 5 N  4.5  500606.989  150175.266
```



- Creation of FME features from a Python script
- Useful for e.g. formats not natively supported by FME or needing advanced pre-processing
- Script is called only once, but may return an arbitrary number of features
- Expects method close() to output all features using method pyoutput()

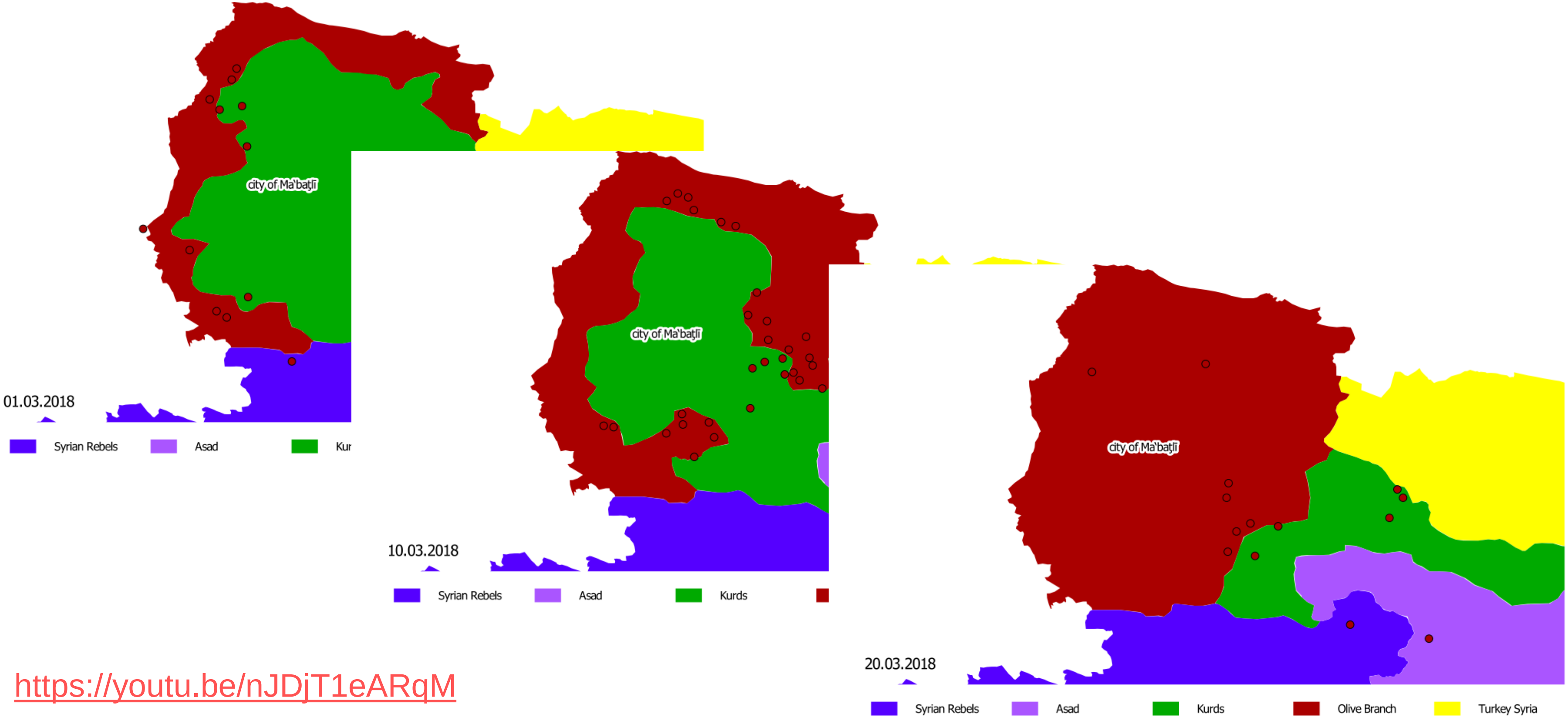
# Ex2 – Custom ptp reader

Log into your computer → Read the doc for exercise 2 → Follow the steps to finish the exercise

# Shutdown scripted

# Syria

# Ex3 – Shutdown python script - Syria



https://youtu.be/nJDjT1eARqM

# Ex3 – Shutdown python script - Syria

**FME installs its own Python interpreter, but you may wish to use a different one.**

**Why would I want to choose a different Python Interpreter?**

- **Arcpy** - When creating a Python script for use with both FME and ArcGIS (for example) you could point FME to use the Python Interpreter installed by ArcGIS, to ensure both applications work.

- When you want to integrate FME with a **3rd-party Python package**.

- When you need to run a script in a **different version of Python** to that which FME installs, you will need to install that version separately and direct FME to use it.

# Ex3 – Shutdown python script - Syria

Log into your computer → Read the doc for exercise 3 → Explore the FME workspace → Follow the steps to finish the exercise

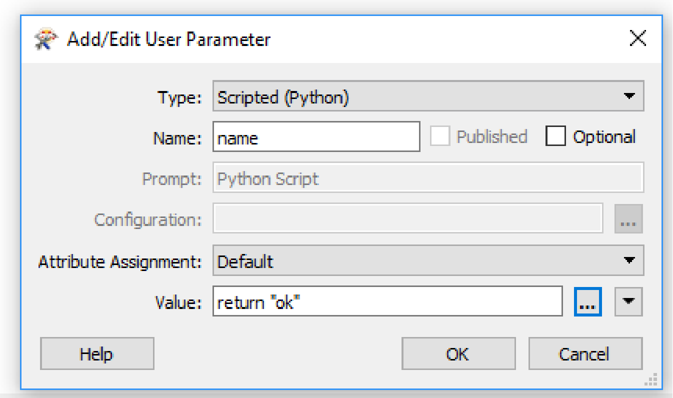
# Scripted parameters

# DEM Digital elevation model

# Ex4 – Scripted parameters - DEM

Scripted parameters are extremely useful when we want to set a parameter in FME based on something we derived or calculated from another parameter or parameters.

For example you may want users to select themes or groups of layers and have your script set the individual feature types to read within these groups.



Add/Edit User Parameter

Type: Scripted (Python)

Name: name □ Published □ Optional

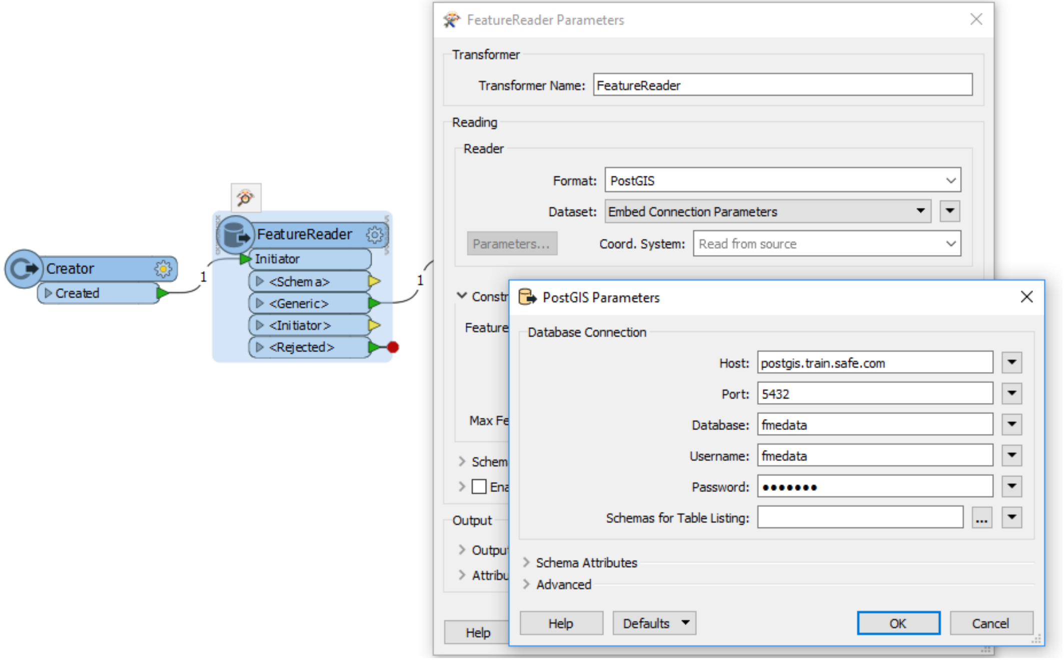
Prompt: Python Script

Configuration:

Attribute Assignment: Default

Value: return "ok"

# Ex4 – Scripted parameters - DEM



Transformer Name: FeatureReader

Format: PostGIS

Dataset: Embed Connection Parameters

Coord. System: Read from source

Host: postgis.train.safe.com

Port: 5432

Database: fmedata

Username: fmedata

Password: •••••••

Schemas for Table Listing:

Working with parameters is mainly usefully when deploying one workspace in different environments (development env., test env. , User acceptance testing env. production env.).

Paths and credentials can be easily adapted to the environment without changing the workspace.

# Ex4 – Scripted parameters - DEM

Log into your computer → Read the doc for exercise 4 → Explore the FME workspace → Follow the steps to finish the exercise