con·terra

Webinar Python Scripting in FME

Ken Bragg
European Services Manager
Safe Software Inc.

Tino Miegel
Software Engineer
con terra GmbH

Stefan Offermann
Software Engineer
con terra GmbH

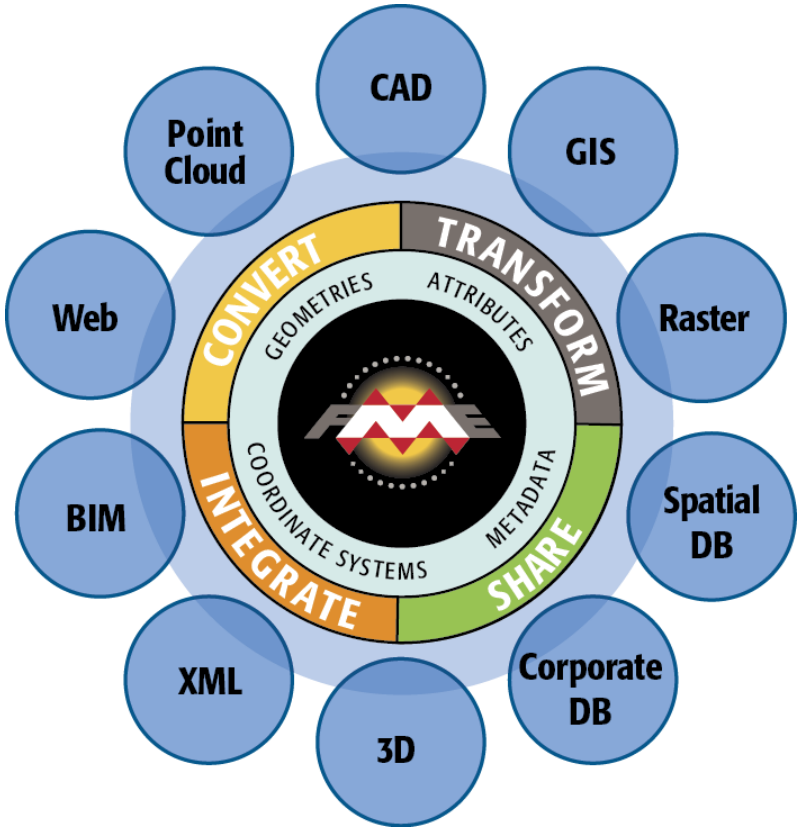
Christian Dahmen
Consultant
con terra GmbH

Poll: About You #1

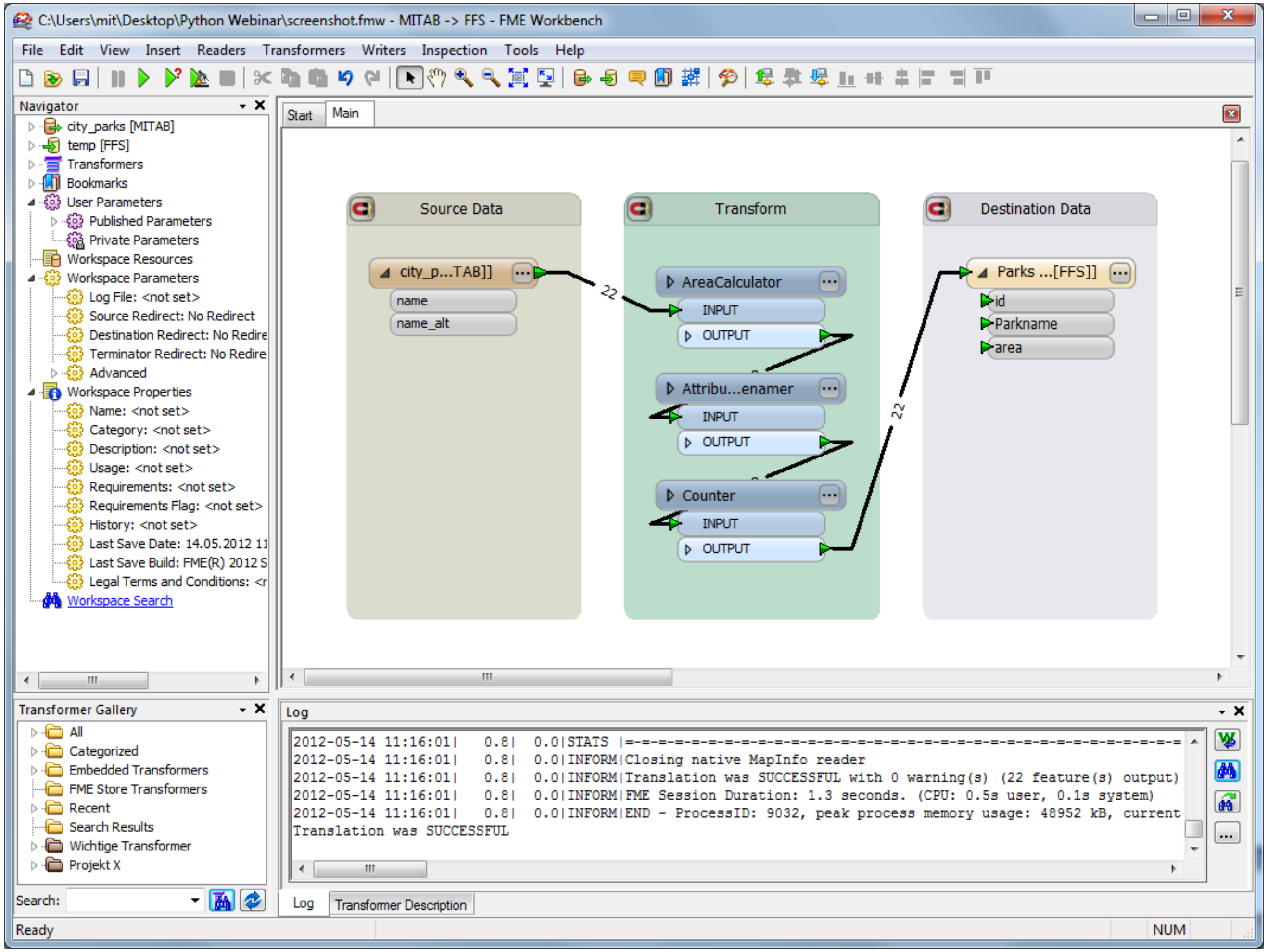
- **How long have you been using FME?**

  > Never
  > Less than 1 year
  > 1-2 years
  > More than 3 years

Powering the Flow of Spatial Data



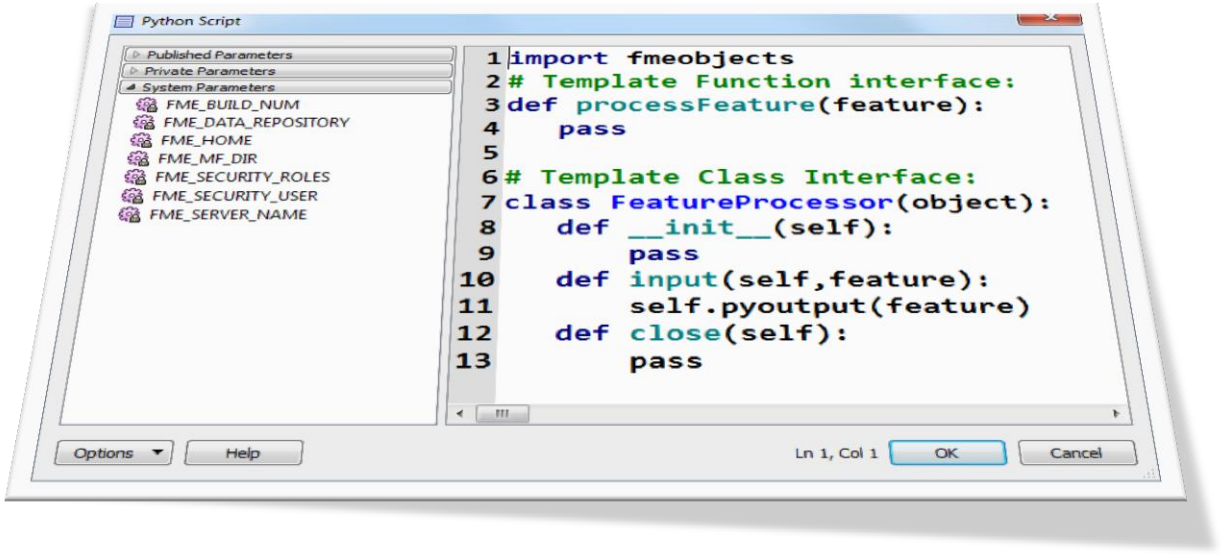
FME Workbench



con terra

Agenda

- What is Python?
- FME and Python
- Best Practice



Poll: About You #2

- **Do you have any experience with Python?**

  > Yes, long time Python user
  > Yes, also within FME
  > Some
  > None

What is Python?

- Programming/Scripting Language
- Easy to learn and use, well documented
- Great user community
- Platform independent
- Great for GIS automatization tasks
- It‘s free!

- More details on www.python.org

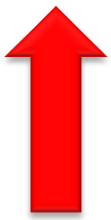
Python Basics

- **Variables (data-types)**
  > String, Integer, Float, List, Dictionary, Tuple
  > Dynamic typing
- **Built-In methods**
  > len(), max(), min(), …
- **Modules**
  > Thematic-grouped extensions
  > e.g. math, os, numpy, zip, re
- **IDEs**
  > IDLE, PyWin, PyDev for Eclipse, PyScripter, …

```
a=1 #integer
b='Hello, World!' #string
```

Sample #1

```
# Sample #1
# print even numbers from 1 to 9
a = range(1,10)                 # list from…to [excluded]
# for-loop
for i in a:
    if i%2 == 0:                # condition
        print 'number %i is even' %i
    else:
        print 'number %i is odd' %i
```



Line indentation is very important in Python!

FME and Python

- **Why and when should one make use of Python with FME Workbench?**

- **Want to do automatization tasks?**

  > Call Workspace from your Python script

- **Missing the right transformer?**

  > Write a few lines of python code in PythonCaller

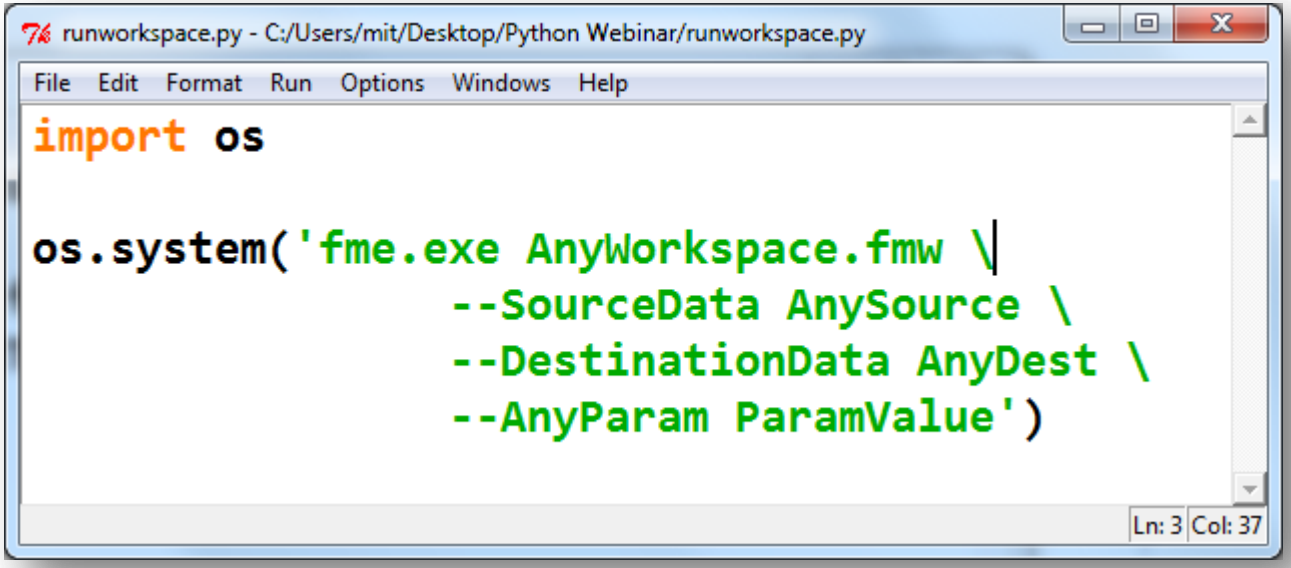
- **Want to carry out tasks before or after translation automatically?**

  > Use Python Startup or Shutdown Scripts

- **ALWAYS look for an existing FME Transformer or functionality first!**

Sample #2

- **Run FME Workspaces from a Python script (IDLE)**



Python in FME Workbench

- **Startup Script**
- **Shutdown Script**
- **Transformer PythonCreator & PythonCaller**
- **Private Scripted Parameter**

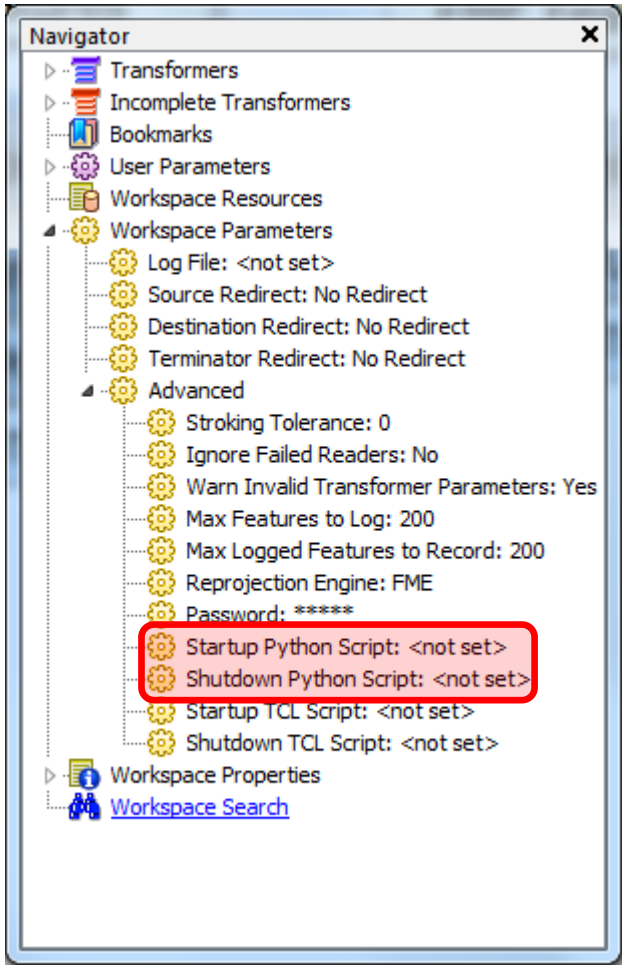
- ***Short Introduction into FME Objects API***

Startup Script

- **Executed prior the actual FME translation process**

- **Potential uses**
  - > Define your own Python functions
  - > Check database connectivity
  - > Move data or copy a template file
  - > Add your custom messages to Logfile
  - > Access any FME Macro Values:

```
FME_MacroValues['SourceDataset_ACAD']
```



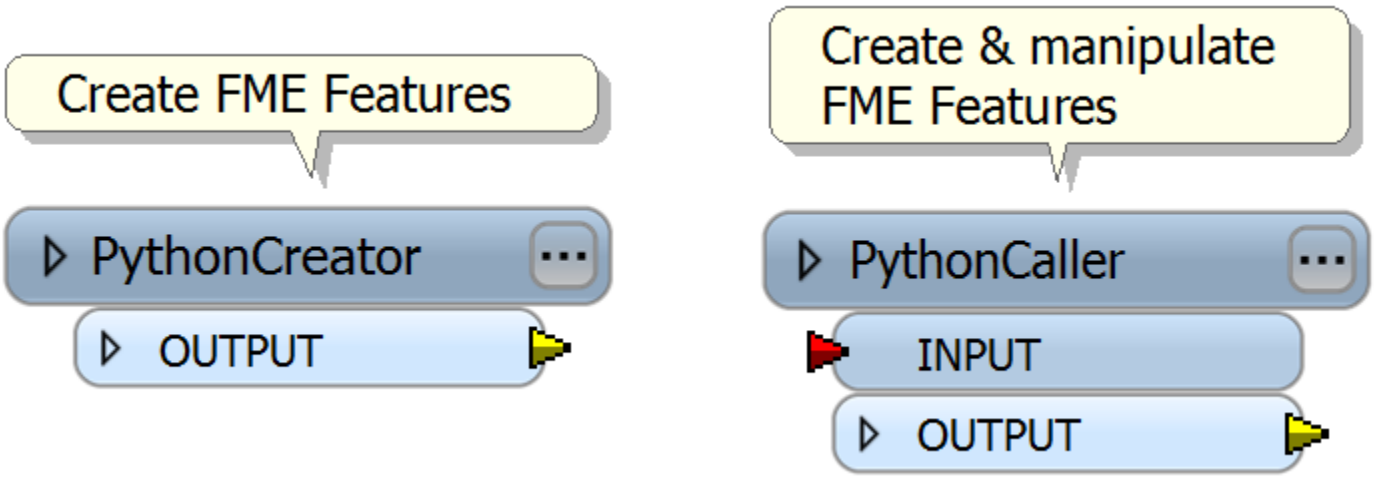
Shutdown Script

- **Executed after all Reader/ Writer work is done and process is finished either with SUCCESS or FAILURE**
- **Potential use**
  - > Any kind of post processing (e.g. calling ArcGIS scripting through ArcPy)
  - > Accessing statistical information about translation
  - > Copy result files
  - > Send an Success/ Failure email

- **Sample #3**

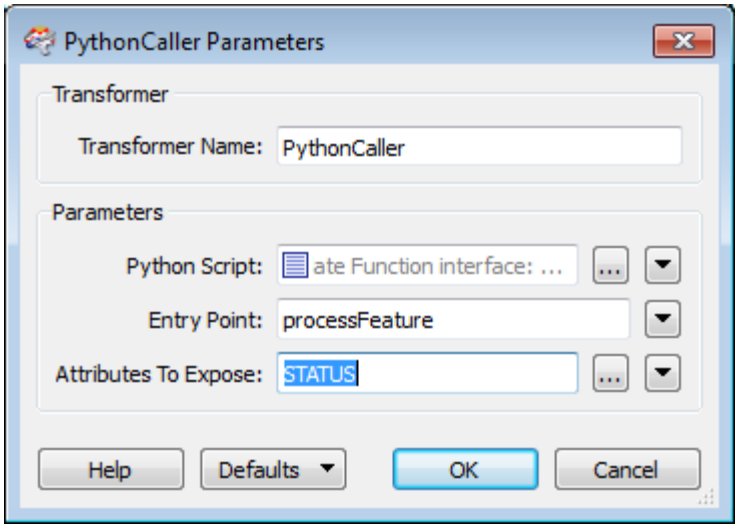
PythonCreator & PythonCaller

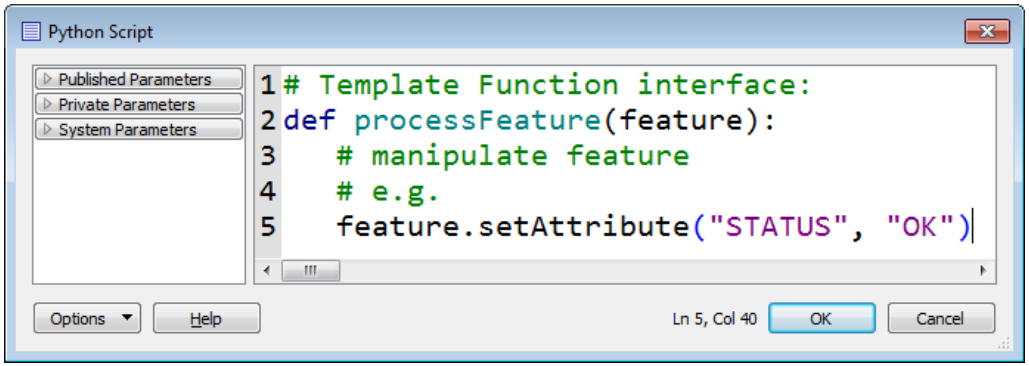
- **Both Transformers can execute Python scripts to manipulate or create FME Features.**



PythonCreator & PythonCaller

- **Parameters**
  - > Python Script
  - > Entry Point
  - > Attributes To Expose (opt)





Function

- **Using a Function (per-feature-manipulation)**

  > The function gets called for every feature passing the Transformer.
- **PythonCaller only**

```
# Template Function interface:
def processFeature(feature):
      # manipulate feature
      feature.buffer(10.0)
      feature.setAttribute("STATUS", "Buffered")
```

Class

- **Class: Allows to manipulate a group of features and create new features.**
- **PythonCaller and PythonCreator**

```
class MyClass(object):
    def __init__(self):
        # Constructor is called only once
    def input(self,feature):
        # input() is called for every feature passing
        # self.pyoutput() can be called in input() to
        # output the processed feature
        self.pyoutput(feature)
    def close(self):
        # close is called only once when the last feature has been
        # processed in input().
        # self.pyoutput() can be called to output new features.
        self.pyoutput(newFeature)
```

FMEObjects module

- **Python API to access FME functionality**
  - > FMEFeature()
  - > FMEGeometry()
  - > FMELogFile()
  - > ...
- **Documentation**
  - > %FME_Home%\help\python\apidoc
  - > %FME_Home%\fmeobjects\python\apidoc (more detailed) -> Requires installation of SDK

Sample #4

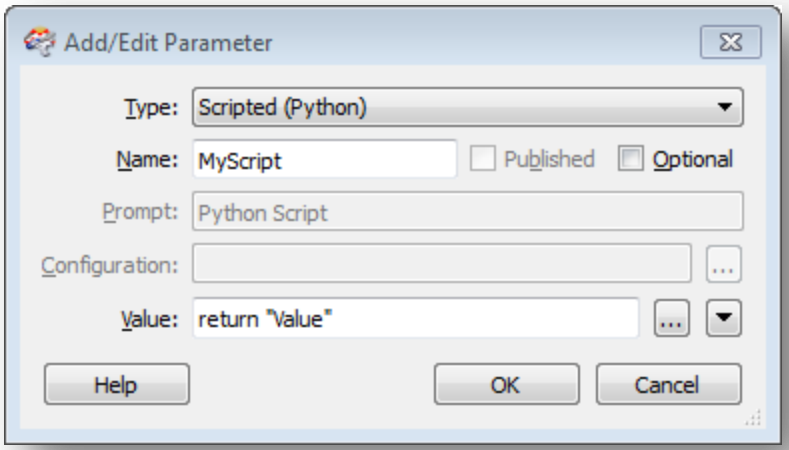
- **Python Olympics**

Private Scripted Parameter

- **Python script for assigning the value of a parameter to the workspace at runtime.**
- **An additional type of User Parameter**
- **Usage**

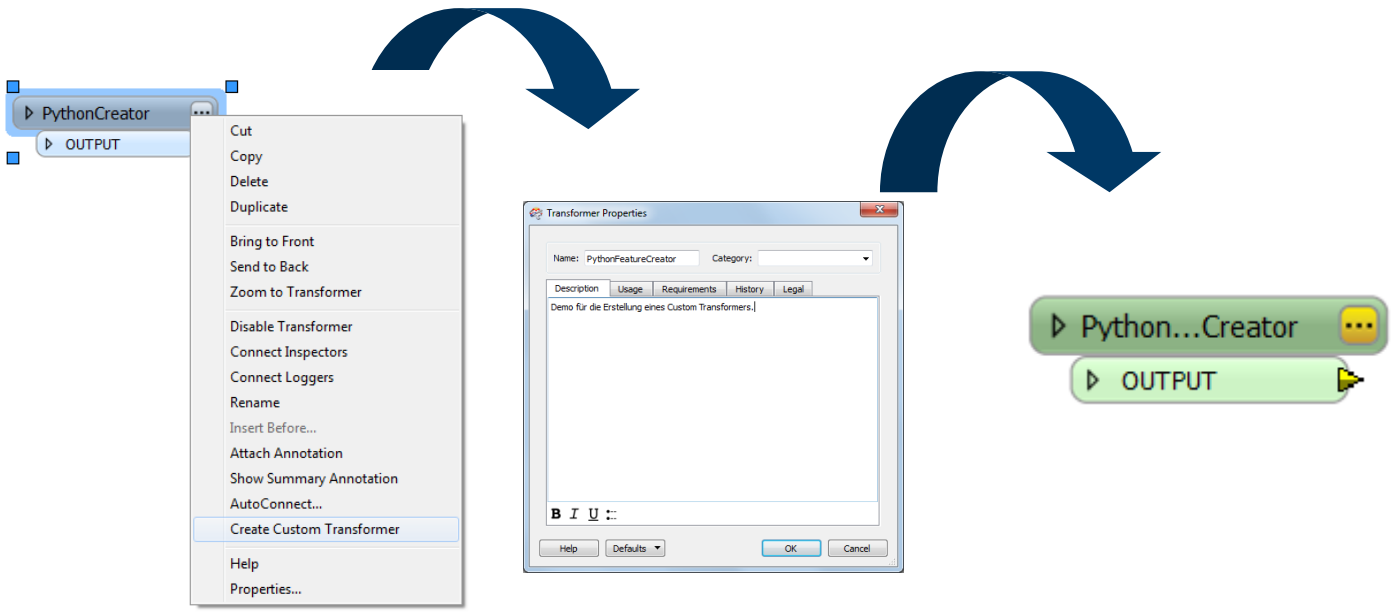
  > Hide (complex) functionality

- **Sample #5**



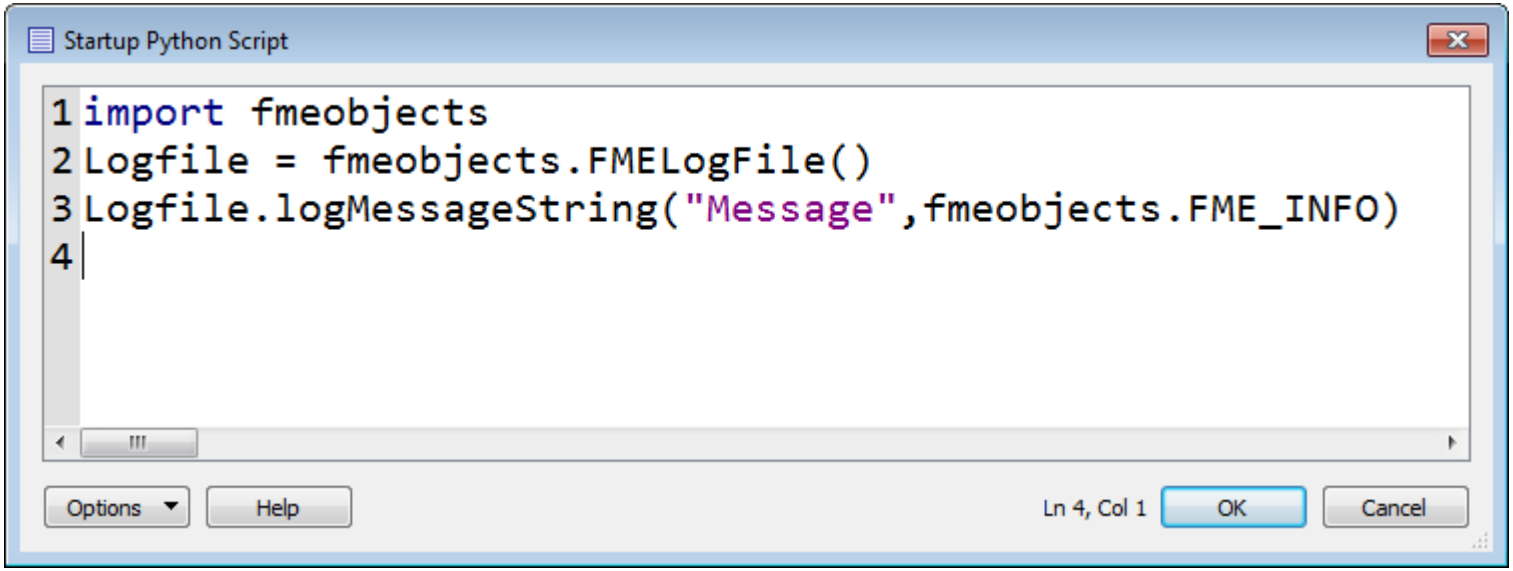
Best Practice I

- **Custom Transformer**
  > Create Custom Transformer from PythonCreator or PythonCaller to extend your Transformer Gallery



Best Practice II

- Use FME logging instead of print() statements
- Choose Severity Level (Info, Warn, Error, …)
- Messages are included into written Logfile



Documentation

- **FME Workbench Transformer Description**
- **Help -> Workbench Help**
- **FME Store (e.g. FuzzyStringComparer)**
- **FMEpedia**
  > Example-scripts-for-deleting-Excel-files-prior-to-writing
  > Python-and-FME-Basics
  > What-is-Python-and-How-Can-I-Install-It

The Road ahead

- **Python begin and end transformers**
- **Allow creation of input and output ports with PythonCaller transformer**
- **Using Python in place of TCL in FME transformers where performance would benefit**
- **Return more than one parameter from a Scripted Parameter**
- **Python Plug-In SDK**

  > allow ability to create transformers using Python plugin SDK

  > more samples and documentation for Python Plugin SDK

Priority on YOUR request and feedback!

Poll #3

- **Are you interested in an online FME & Python training course?**

  > Yes, absolutely
  > Maybe
  > No, thanks

What's Next?

- See FME 2012 on the FME World Tour: http://fme.ly/2012tour
- Read our latest newsletter www.safe.com/newsletter
- Download FME 2012: www.safe.com/downloads

con terra

Share Today's Webinar

- Today's webinar was recorded

Thank you!

- **Questions about Python and FME?**
- **Interested in FME Training?**
  - > Basic and Advanced trainings
  - > On-site training
  - > FME Certified Trainers + Professionals

- **Send an email to fme@conterra.de**

con terra GmbH
European FME Service Center
Martin-Luther-King-Weg 24
48155 Muenster, Germany
+49 251 74745 0
www.conterra.de

con·terra