Python in FME

Simple and powerful scripting
Python

- Great for beginners and superb for experts
- Object-oriented
- Big user community
- Independent of OS
- Wealth of 3rd party packages
FME and Python

- **Startup Python script**
  - Tasks before a workspace runs

- **Scripted Parameters**
  - Hide complex functionality or set parameters based on calculated parameters

- **PythonCaller and PythonCreator Transformers**
  - Manipulate and create features or groups of data

- **Shutdown Python script**
  - Tasks after a workspace runs
Startup script example

- Start up scripts that uses arcpy to check SDE database connection.

```python
import fmeobjects, arcpy, sys

# Set variables
logFile = fmeobjects.FMELogFile()
ft = FME_MacroValues["FEATURE_TYPES"]

# Make connection to SDE
arcpy.env.workspace = r"C: \Users\arloit\AppData\Roaming\ESRI\Desktop10.0\ArcCatalog\TNE.sde"
fcList = arcpy.ListFeatureClasses()

# Check if the selected feature type exists
try:
    if ft in fcList:
        logFile.logMessageString("Connection tested. Found feature type: " + ft,fmeobjects.FME_INFORM)
    else:
        logFile.logMessageString("No feature type selected. Please set a feature type to read.",fmeobjects.FME_WARN)
except:
    logFile.logMessageString("No connection or Feature Type not existing: " + ft,fmeobjects.FME_WARN)
raise sys.exit("Could not find feature type or connect to database")
```
PythonCaller and PythonCreator

- Can execute scripts to manipulate and create features
- Includes a template for getting you started

```python
import fmeobjects

# Template Function interface:
def processFeature(feature):
    pass

# Template Class Interface:
class FeatureProcessor(object):
    def __init__(self):
        pass

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

    def close(self):
        pass
```
PythonCaller and PythonCreator parameters

- Connection between parameters and script

```python
import fmeobjects

def CoordinateFetch(feature):
    # Define variables
    alist = []
    blist = []
    # Get list of all coordinates for a feature
    num_coords = feature.numCoords()
    i = 0  # Set counter
    # Loop through all coordinates
    while i <= num_coords:
        First = " ".join(map(str,feature.getCoordinate(i)))
        alist.append(First)
        # Save the first set of coordinates
        if i == 0:
            First = " ".join(map(str,feature.getCoordinate(i)))
            alist.append(First)
        # Save the last set of coordinates
        elif i == (num_coords - 1):
            Last = " ".join(map(str,feature.getCoordinate(i)))
            blist.append(Last)
        # Update the counter
        i = i + 1
    # Concatenate the two sets of coordinate to a list
    clist = First + " "+ Last
    # Send back results
    feature.setAttribute("_CoordList", clist)
```
Function or Class

- **Function**
  - Process one feature at a time

- **Class**
  - Can in addition process group of features
  - Filter out features
PythonCaller Transformer example

- Get the first and last coordinates written to a list

```python
import fmeobjects

def CoordinateFetch(feature):
    # Define variables
    alist = []
    blist = []
    # Get list of all coordinates for a feature
    num_coords = feature.numCoords()

    i = 0 # Set counter
    # Loop through all coordinates
    while i <= num_coords:
        # Save the first set of coordinates
        if i == 0:
            First = " ".join(map(str, feature.getCoordinate(i)))
            alist.append(First)
        # Save the last set of coordinates
        elif i == (num_coords - 1):
            Last = " ".join(map(str, feature.getCoordinate(i)))
            blist.append(Last)
        i = i + 1 # Update the counter

    # Concatenate the two sets of coordinate to a list
    clist = First + ";" + Last
    # Send back result to FME
    feature.setAttribute("_coordlist", clist)
```
Shutdown script example

Send a logmessage in a sms

```python
# Importer library
import fmeobjects
from twill.commands import go, showforms, formclear, fv, submit

# Define a class
class MySMS(object):
    def __init__(self):
        self.countFeatures = 0  # set the counter
    def input(self, feature):
        self.countFeatures += 1  # Count objects
        self.id = feature.getAttribute('_count')  # Get the _count attribute from FME
    def close(self):
        # Make a string for the message
        self.count = ("Counter is complete. Total number of features: ' + str(self.countFeatures))
        self.lastReadID = ('. Last read ID is: ' + str(self.id))
        self.msg = (self.count + self.lastReadID)

        # Twill - This is all it takes to send an sms from FME with Python
        # go ('url') defines which url to go to
        go('http://online.telefonkatalogen.no/online/editsms.php?dummy=1&msg=&emailaddr=&sendsmsemail=
        # Show webform
        showforms()
        # fv is short for formvalue.
        fv("1", "msg", "melding")
        # submit form id # 12 for "send now"
        submit('12')
```
Questions?

arne.loitegaard@vegvesen.no

Havøysund, Finnmark
Photo: Nasjonale Turistveger / Statens vegvesen