
Open Data Automation Documentation

Release 1.0.1

Open Chattanooga

Dec 18, 2018

1	Index	3
1.1	Workflow Organization	3
1.2	Common Workflows	4
1.3	Workflow Scheduling	5
1.4	Best Practices	6

Here we describe how we are approaching data automation with [Safe FME](#). We are reading from a variety of data sources and writing into [Socrata](#).

1.1 Workflow Organization

1.1.1 A Single Workflow Project

A single workflow project is usually a part of a large number of other workflows. It is good for them to be organized in a consistent manner.

Workflow Runner

This is an FME workflow that runs the project's main workflow. This is the actual workflow that is scheduled to run by the master workflow or placed directly on a scheduler.

The name of this file for each workflow project is called `runner.fmw`.

Dataset Info

This is a CSV file that includes information about the main workflow including the full filepath to the workflow. The workflow runner reads this file and executes the main workflow from the full filepath included in the dataset info file.

The name of this file for each workflow project is called `dataset_info.csv`.

The following is an example of the contents of a dataset info file:

```
Job Order,Job Name,Description,Path to fmw file
1,Some Dataset,This is a dataset.,C:\Workflows\Some Category\Some Dataset\somedataset.
↪fmw
```

Main Workflow

This is the project's main FME data extraction workflow. These workflows can be run independently if needed. So, it is best to give the filename of the workflow something close to the name of the actual dataset instead of something generic.

1.1.2 Grouping Multiple Workflows

The following directory organization has been followed: All Workflows → Entity Name → Dataset Name

All Workflows is a directory that contains all of our workflows. Entity name is a directory named after the entity/organization that the dataset comes from. Dataset Name is a directory that is named after the actual dataset and contains all of dataflow project's files.

For instance, we have the Building Permit dataset from Economic and Community Development. Our directory that contains all workflows called Workflows. Assuming our Workflow directory is located on the root of some drive, for example G:, our file path might look like: G:\Workflows\Economic and Community Development\Building Permit\.

1.2 Common Workflows

There are four common workflows that we have encountered while automating data: basic, geospatial (point), geospatial (polygon, and etc.), and SQLCreator.

All workflows use the [Socrata writer](#) except for the geospatial (polygon, and etc.) workflow.

1.2.1 Basic

This is the most basic workflow. It requires, at the very least, a basic [reader](#) and Socrata writer.

1.2.2 Geospatial (Point Data)

This workflow requires use of a geospatial reader. You will also need, at least, a [CoordinateExtractor](#) for extracting latitude and longitude, [AttributeReprojector](#) to project the coordinates into EPSG:4326/LL84, a [StringConcatenator](#) for making a Location column, and a Socrata writer.

Your point columns should be represented by the following names and types:

- Latitude
 - Type: Number
- Longitude
 - Type: Number
- Location
 - Type: Location

1.2.3 Geospatial (Polygons, and etc.)

TODO

1.2.4 SQLCreator

This workflow uses the SQLCreator “reader” and a Socrata writer at the minimum. It is different from a normal reader such that you have more control over your data source via SQL for complex joins and the like. You should use this if you have existing workflows already utilizing SQL.

1.3 Workflow Scheduling

If you have several workflows that need to run periodically, you will need to schedule them with some sort of mechanism. You could schedule each workflow to run individually. However, there are other ways of scheduling workflows that have been found to be very beneficial.

1.3.1 Master Workflow

Safe FME has the ability to do batch processing. You can take advantage of this by creating a master workflow that reads in an index of all workflows, and runs each one.

You can use a CSV reader to read in a CSV where each record represents a workflow you want automated. Each record in the CSV will need to contain the full path to a workflow.

The CSV will then be piped into a `WorkspaceRunner` that will run each workflow.

From here if you wish, you can use the Socrata writer to write to a dataset that keeps record of all of your workflow failures and successes as well as for how long it took to run them.

1.3.2 Job Management

Depending on your operating system, there are many choices for job management. Out of the box, Windows offers the task scheduler. OS X and Linux both use cron. You will need to schedule jobs, monitor them, and potentially stop jobs that may not have stopped for whatever reason.

Windows Task Scheduler and Task Manager

The task scheduler can be accessed by opening the Control Panel, then going to Administrative Tools. The most comprehensive documentation for Windows’ Task Scheduler is located [here](#). Reading that will answer any question you will have about the Task Scheduler, but we’ll go over common use cases below.

Creating A Task

The easiest way to create a task is to use the “Create Basic Task Wizard” by going to `Action` in the file menu then clicking `Create Basic Task`. From here, you can specify when the task should run. It will also ask you what program you want to run and the parameters you want to use. The best place to find this information is at the top of the FME workspace log window where it lists the command-line info needed to run the workspace (see the example below).

```
Command-line to run this workspace:
"C:\Program Files\FME\fme.exe" \\some\filepath\to\your\workflow\opendata_
↪daily.fmw
--SourceDataset_CSV "\\some\filepath\to\your\parameter\opendata_daily_
↪index.csv"
```

For automating a Safe FME workflow using the above example, you will want to enter something like the following into the Action section of the Task Scheduler:

Program/script:

```
"C:\Program Files\FME\fme.exe"
```

Add Arguments (optional):

```
\\some\filepath\to\your\workflow\opendata_daily.fmw --SourceDataset_CSV  
"\\some\filepath\to\your\parameter\opendata_daily_index.csv"
```

Start In (optional):

```
\\some\filepath\to\your\workflow\
```

Stopping A Task

There will be occasions where a workflow may not stop running. You should first check if the scheduled task is still running by opening up the scheduler. Check the “Last Run Result” of your job. If it’s finished, it should say something like “The operation completed successfully.” Otherwise, try right clicking on the task and then left click “End”.

If that doesn’t work. Open up the Windows Task Manager. You can get to the Windows Task Manager by pressing Ctrl+Shift+Esc at the same time.

When you have the Task Manager open, look in the “Processes” tab for any instances of `FME.exe` or `fme.exe`. If you don’t have Safe FME open, those are likely workflows that are still trying to run. To end a task, right click on it then left click “End task”. Use this to close any instances of `FME.exe` or `fme.exe`.

1.4 Best Practices

These are good guidelines to go by when setting up a workflow for automation. Following these guidelines will generally prevent you from falling into certain pitfalls mentioned below.

1.4.1 Readers

Readers are generally fairly straightforward, but sometimes you will run into issues with them. You should follow these practices to avoid hitting a snag.

If Reading From A Geospatial Database Choose The Geospatial Reader

Many times a database, like SQL Server, will have a non-spatial and spatial reader. If you are reading geospatial data, choose the spatial reader. If your data is non-spatial, choose the non-spatial reader.

This is the most confusing where the non-spatial and spatial reader name, like for SQL Server, are very similar.

Set A Timeout

If you set a timeout of zero for the reader, the reader will attempt to read from the data source forever. It is generally ok if you do this while you are working inside FME. However, if you run this workflow from the master workflow, the potential is there for the workflow to not ever stop running.

Setting your timeout to 30 seconds is generally good. If you find that your reader needs more time, set a higher timeout.

1.4.2 Dates And Times

Convert Dates and Times To ISO 8601

Use the [DateFormatter](#) transformer to convert date/time columns into a standard date/time format. Socrata understands several [date/time formats](#), but it is best to adhere to ISO 8601 as much as possible.

1.4.3 Geospatial Data

Ensure Geospatial Data is Projected To EPSG:4326/LL84

Many times geospatial will be stored in a regionally relevant coordinate reference system. In our case: EPSG:2274. You should always re-project to EPSG:4326/LL84 unless there is a very good reason not to do so.

1.4.4 Scripting

The ability to write code in your workflows is very powerful. However, you should heed to these recommendations.

Use A Transformer When Possible

It is very likely there already exists a transformer for something that you want to do. You should search the [FME knowledgebase](#) and double check before you start writing code.

If a transformer for your problem doesn't exist, then go ahead. A notable example we've run into where we've needed to write code is pulling data from SFTP servers. FME does not support the SFTP protocol out of the box.

Use Python, Not TCL

There is nothing wrong with TCL. Both Python and TCL both seem to be well supported. However, if you get stuck, you will more likely find help with Python than with TCL.

Also, there are a myriad of libraries for Python. There is likely already a library for what you want to do.

Follow Standard Python Style Convention

Follow the [PEP8](#) Python style guide for writing clear code.