Article

← FME Desktop (/S/Topic/0TO4Q000000QL9uWAG/Fme-...

Writing JSON

Aug 4, 2022 • Knowledge

Product Type
FME Desktop

FME Version
2022.0

Tutorial: Tutorial: Getting Started with JSON (/s/article/tutorial-getting-started-with-json) | **Previous:** Advanced JSON Reading (/s/article/json-reading-advanced) | **Next:** Writing JSON with JSONTemplater (/s/article/json-writing-with-jsontemplater)

Introduction

FME offers a number of ways to write JSON data. This table provides an overview of methods and their common use-cases.

Method	Common Scenario	Tutorial
JSON Writer	Writing simple, unnested JSON	This page
JSONTemplater + Text File Writer	Writing nested JSON	Writing JSON with the JSONTemplater (/s/article/json-writing-with-jsontemplater)
FeatureWriter	Writing JSON in the middle of a translation	The FeatureWriter Transformer (/s/article/basic-featurewriter-example)

HTTPCaller	Sending a JSON message via an HTTP POST request	Tutorial: Getting Started with APIs (/s/article/getting-started-with-apis-1)
GeoJSON Writer	You want the destination JSON data to follow the GeoJSON specification	GeoJSON Reader/Writer Documentation (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/geojson/geojson.htm)
Esri JSON Writer	You want the destination JSON data to follow the Esri JSON specification	Esri-JSON Reader/Writer Documentation (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/esrijson/esrijson.htm)
IMDF Writer	You want the destination JSON data to follow the IMDF GeoJSON specification	IMDF Reader/Writer Documentation (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/imdf/imdf.htm)

This article shows a simple example of writing using the JSON writer. The JSON writer provides a quick and easy way to get the data into JSON format. However, it is only suited to writing flat JSON without nesting. If you want to write nested JSON, you can use the JSONTemplater with a Text File writer. That workflow is covered in the next tutorial (https://community.safe.com/s/article/json-writing-with-jsontemplater).

Content Overview

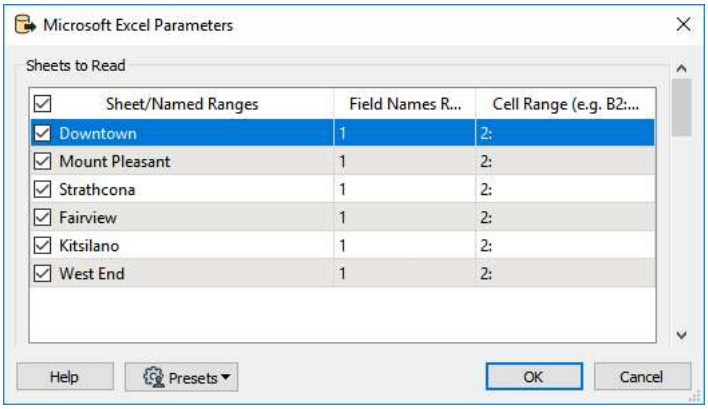
Step-by-step Instructions

Part 1: Writing JSON

1. Start FME Workbench and Generate Workspace

Start FME Workbench and click Generate Workspace. For Reader Format, select Microsoft Excel. The dataset is PublicArt.xlsx, which can be downloaded from the Files section of this article.

Click the Parameters button to preview the Excel data. You'll see the Excel Workbook is comprised of six sheets, each of which represents a neighborhood in Vancouver.



Click Cancel, as we don't need to modify our reader parameters.

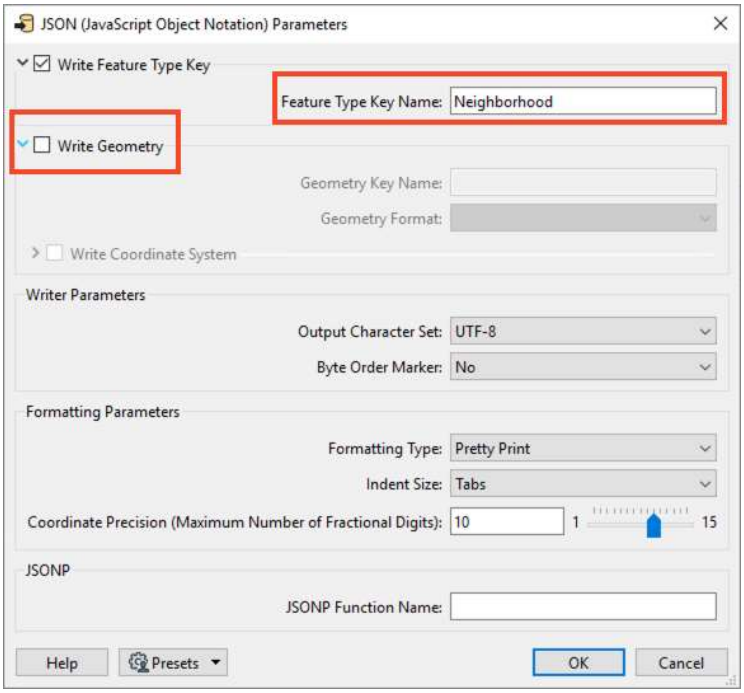
For Writer Format, select JSON (JavaScript Object Notation). For Dataset, write out a local JSON file called PublicArt.json wherever you wish.

Click the writer Parameters button. By default, when writing to JSON, each FME feature becomes one element in the JSON array, with attributes nested under it. The FME feature type is included as an attribute, json_featuretype. By default, Write Geometry is checked, which lets you include FME geometry in the destination JSON file in a number of formats.

For this article, we will be writing each public art installation as an element in the JSON array, but we want to include the neighborhood as an attribute. Because each sheet in the source Excel file corresponds to a neighborhood, we can simply rename the default json_featuretype attribute to Neighborhood.

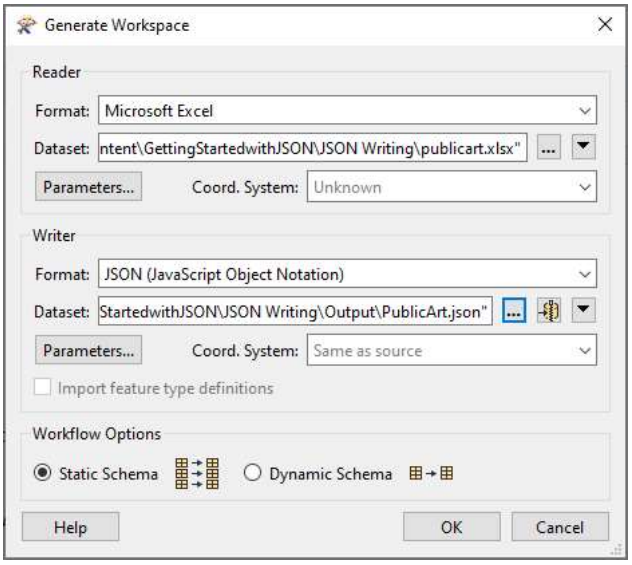
Uncheck Write Geometry, because we don't want to include geometry in our simple JSON output.

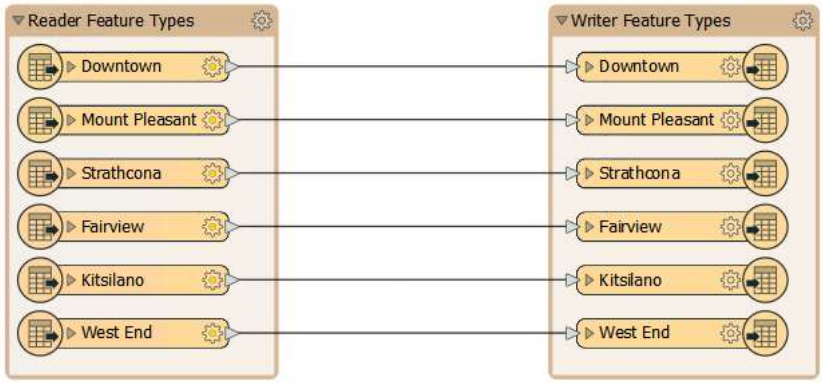
We can leave the remaining writer parameters as default. The writer parameters dialog should look like this:



Click OK.

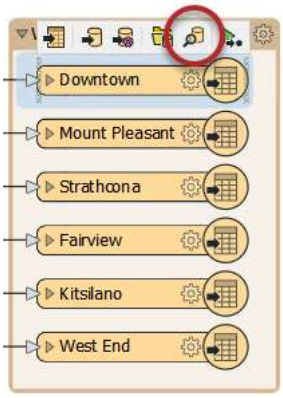
The Generate Workspace dialog should look like this:



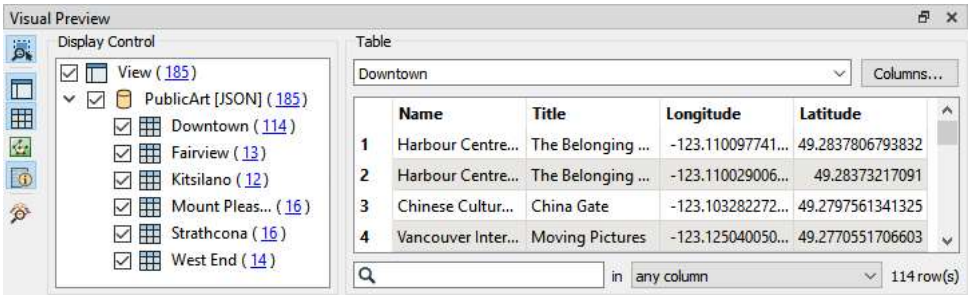


2. Run the Workspace

Click the Run button to run the workspace. Once the workspace has run successfully, click the Downtown feature type to select it. Then click the Inspect Written Data button to open the output dataset in Visual Preview:



You should see the Downtown feature type displayed in Table View, with all its original attributes. You can verify that the feature type attribute Neighborhood is being used to split the data into feature types by opening the Display Control window to see that each neighborhood is a feature type within the JSON file.



3. Inspect the Result

If you inspect the JSON file in a text editor like Notepad++, you'll notice that each feature has a Neighborhood attribute:

```
[
    {
        "Neighborhood" : "Downtown",
        "Name" : "Harbour Centre Parkade",
        "Title" : "The Belonging Action",
        "Longitude" : -123.110097741722,
        "Latitude" : 49.2837806793832
    },
…
]
```

However, you'll also notice that this is not a very efficient way to organize a JSON file, as it does not take advantage of JSON's nested structure at all. It could be structured with Neighborhood as the top-level of the hierarchy, with each art installation underneath the relevant neighborhood, e.g.:

```
{
    "Downtown": [{
            "Name": "Harbour Centre Parkade",
            "Title": "The Belonging Action",
            "Longitude": -123.110097741722,
            "Latitude": 49.2837806793832
        },
        {
            "Name": "Chinese Cultural Centre",
            "Title": "China Gate",
            "Longitude": -123.103282272368,
            "Latitude": 49.2797561341325
        },
        ...
    ],
    "Strathcona": [{
            "Name": "National  Works Yard",
            "Title": "Roller",
            "Longitude": -123.092675,
            "Latitude": 49.2736209999959
        },
        {
            "Name": "Jim Green Residence",
            "Title": "Entranceway",
            "Longitude": -123.095131,
            "Latitude": 49.2842699999959
        },
        ...
    ],
    ...
}
```

This point illustrates that FME's JSON writer is best used for cases where nesting is not required. For cases where you wish to nest the JSON data, refer to the Writing JSON with the JSONTemplater (/s/article/json-writing-with-jsontemplater) and the JSONTemplater transformer (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/jsontemplater.htm).

Part 2: Writing GeoJSON

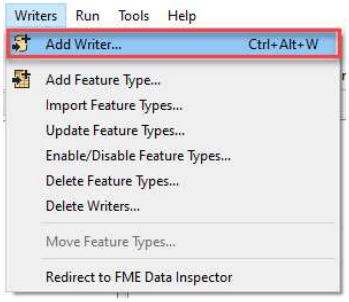
In part 1, you learned how to write JSON. In part 2, we'll write data to GeoJSON. We're going to use the same workspace to create a GeoJSON dataset. GeoJSON is a JSON specification that handles geometry encoding for geographic data.

1. Open Workspace

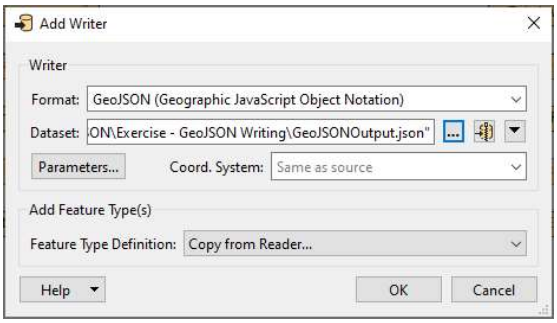
Continue with the workspace from Part 1 or open writing-json.fmw

2. Add GeoJSON Writer

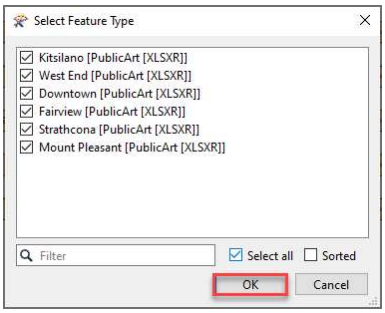
Navigate to the Writers drop-down menu above the toolbar and select Add Writer.



An Add Writer dialog will appear; change the format to GeoJSON (Geographic JavaScript Object Notation). Select a destination location. Ensure the Feature Type Definition is set to Copy from Reader. Click OK.

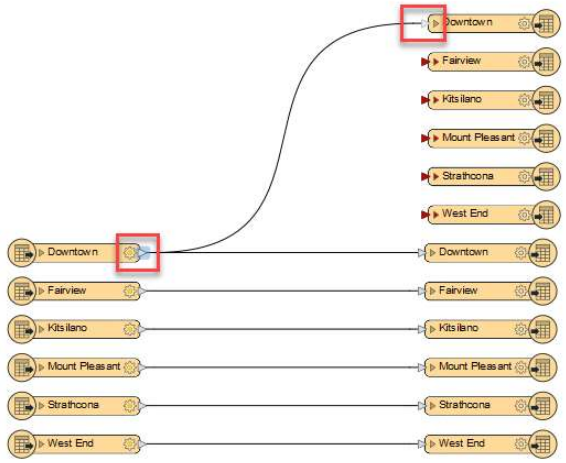


After clicking OK, a Select Feature Types dialog will appear. Select All and click OK.



3. Connect the Reader and Writer Feature Types

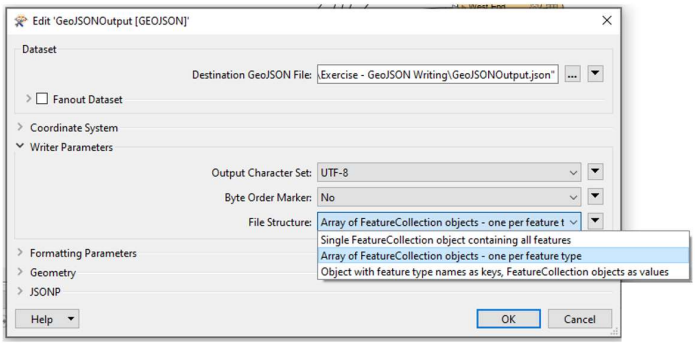
Six new writer feature types will be added to the canvas. Connect the new writer feature types to the corresponding reader. As a keyboard shortcut, you can hold the Shift key, then click.



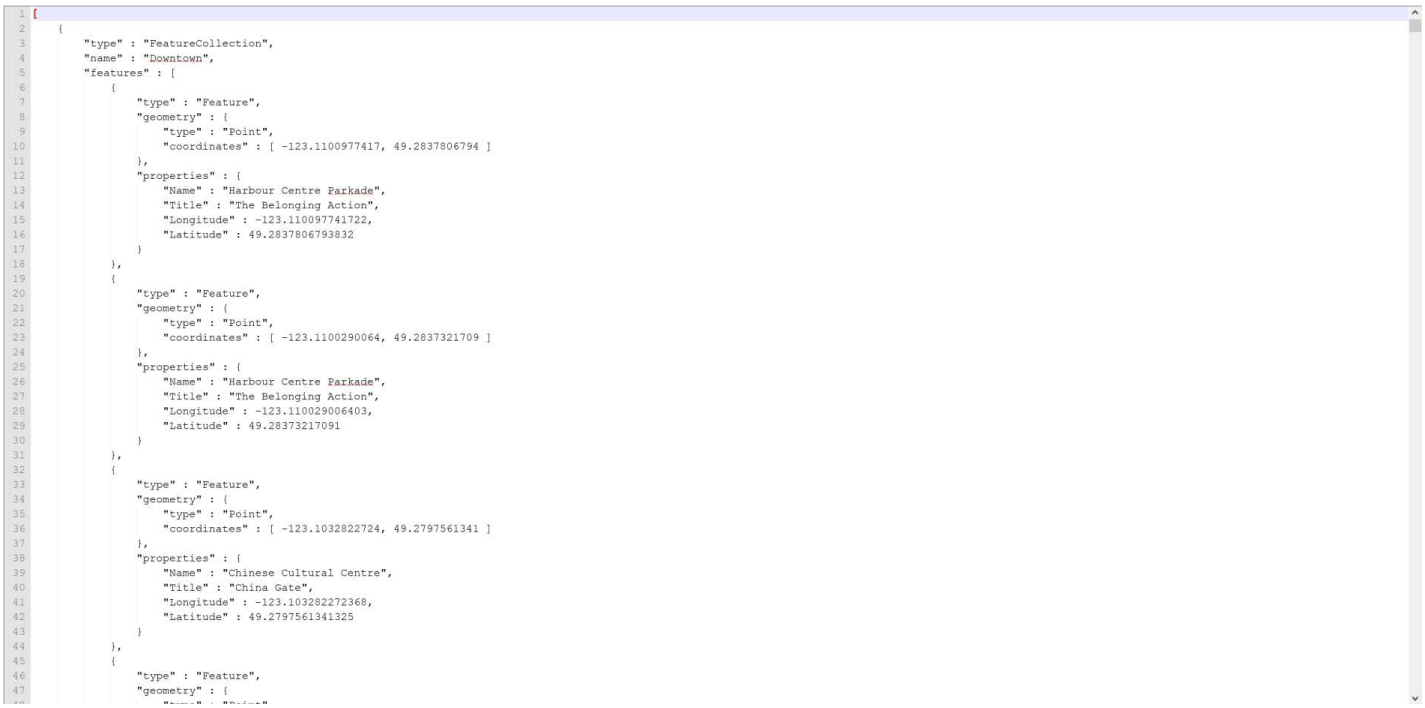
4. Consider the File Structure Parameter

If you have more than one feature type, you might want to consider changing the File Structure Parameter. The value of this parameter controls the structure of

the GeoJSON file produced by the writer. This parameter can be accessed from the Navigator window, by double clicking on the GeoJSON writer destination. You can find the options explained in the GeoJSON Writer documentation (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/geojson/GEOJSON_writer.htm).



Run the workspace and inspect the output. Below is a screenshot of our sample output; the output may look different depending on what File Structure option was used.



```
[
    {
        "type" : "FeatureCollection",
        "name" : "Downtown",
        "features" : [
            {
                "type" : "Feature",
                "geometry" : {
                    "type" : "Point",
                    "coordinates" : [ -123.1100977417, 49.2837806794 ]
                },
                "properties" : {
                    "Name" : "Harbour Centre Parkade",
                    "Title" : "The Belonging Action",
                    "Longitude" : -123.110097741722,
                    "Latitude" : 49.2837806793832
                }
            },
            {
                "type" : "Feature",
                "geometry" : {
                    "type" : "Point",
                    "coordinates" : [ -123.1100290064, 49.2837321709 ]
                },
                "properties" : {
                    "Name" : "Harbour Centre Parkade",
                    "Title" : "The Belonging Action",
                    "Longitude" : -123.110029006403,
                    "Latitude" : 49.28373217091
                }
            },
            {
                "type" : "Feature",
                "geometry" : {
                    "type" : "Point",
                    "coordinates" : [ -123.1032822724, 49.2797561341 ]
                },
                "properties" : {
                    "Name" : "Chinese Cultural Centre",
                    "Title" : "China Gate",
                    "Longitude" : -123.103282272368,
                    "Latitude" : 49.2797561341325
                }
            },
            {
                "type" : "Feature",
                "geometry" : {
```

Continue to the next article: Writing JSON with the JSONTemplater (/s/article/json-writing-with-jsontemplater)

Data Attribution

The data used here originates from open data made available by the City of Vancouver (https://opendata.vancouver.ca/pages/home/), British Columbia. It contains information licensed under the Open Government License - Vancouver.

First Published Date
7/29/2020, 12:18 AM

Last Published Date
8/4/2022, 1:59 AM

Transformation (/s/topic/0TO4Q000000...

FME Desktop (/s/topic/0TO4Q000000...

Sort by:

Latest Posts

LizAtSafe (/s/profile/0050c00000CeGV0AAN) (Employee) published a new version of this Knowledge.
August 4, 2022 at 1:59 AM (/s/feed/0D54Q00009ggWCiSAM)

1 view

Like Comment

Log In to Comment

kailinatsafe (/s/profile/0054Q00000EwiZmQAJ) (Employee) published a new version of this Knowledge.
July 14, 2022 at 7:39 PM (/s/feed/0D54Q00009e2qchSAA)

1 view

Like Comment

Log In to Comment

kailinatsafe (/s/profile/0054Q00000EwiZmQAJ) (Employee) published a new version of this Knowledge.
June 30, 2022 at 7:44 PM (/s/feed/0D54Q00009cP2qLSAS)

1 view

Like Comment

Log In to Comment

kailinatsafe (/s/profile/0054Q00000EwiZmQAJ) (Employee) published a new version of this Knowledge.
June 30, 2022 at 7:42 PM (/s/feed/0D54Q00009cP2BZSA0)

1 view

Like Comment

Log In to Comment

Follow

Files (3) (/s/relatedlist/ka14Q000001DWySQAW/AttachedContentDocuments)

writing-json-2022
Aug 3, 2022 • 72KB • fmw

publicart
Jul 28, 2020 • 13KB • xlsx

publicart
Jul 28, 2020 • 31KB • json

View All

(/s/relatedlist/ka14Q000001DWySQAW/AttachedContentDocuments)

Related Articles

Writing JSON with the JSONTemplater (/s/article/json-writing-with-jsontemplater)

Tutorial: Getting Started with JSON (/s/article/tutorial-getting-started-with-json)

Transforming JSON using the JSONExtractor, JSONFlattener, and JSONFragmenter (/s/article/json-transformations)

Advanced JSON Reading (/s/article/json-reading-advanced)

Extracting Location from JSON (/s/article/converting-from-json-to-a-spatial-format-gis)