

# Article

← [FME Desktop \(/S/Topic/0TO4Q000000QL9uWAG/Fme-...\)](#)

## Dynamic Workflows: Components

🕒 Aug 4, 2022 • Knowledge

### Product Type

FME Desktop

### FME Version

2022.0

**Previous:** [Dynamic Workflows: Definitions \(https://community.safe.com/s/article/Dynamic-Workflow-Definitions\)](https://community.safe.com/s/article/Dynamic-Workflow-Definitions) | **Next:** [Dynamic Workflows with the SchemaScanner \(https://community.safe.com/s/article/Dynamic-Workflows-with-the-SchemaScanner\)](https://community.safe.com/s/article/Dynamic-Workflows-with-the-SchemaScanner).

## Introduction

A dynamic writer can obtain its schema from a number of locations:

- Any incoming dataset
- Any external dataset
- A lookup table
- Defined within the workspace

So one of the key requirements is for the workspace author to define where that schema is to come from. The default for a dynamic writer is to obtain the schema from the incoming dataset, but that can be changed.

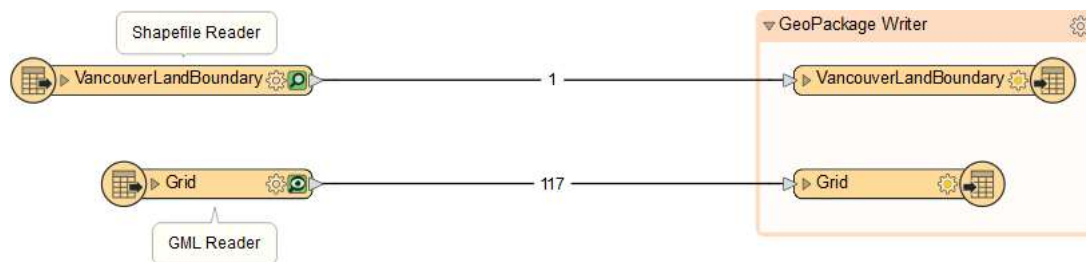
In some workspaces it's possible to define multiple schemas, so another key requirement is to be able to match the schema with the features to which it applies.

## Step-by-step Instructions

The aim here is to familiarize ourselves with the key dynamic components by starting with a static translation and turning it into a dynamic one

### 1. Open Starting Workspace

Open the "Begin" workspace provided 2-WorkflowComponentsExercise-Begin.fmw and run it.

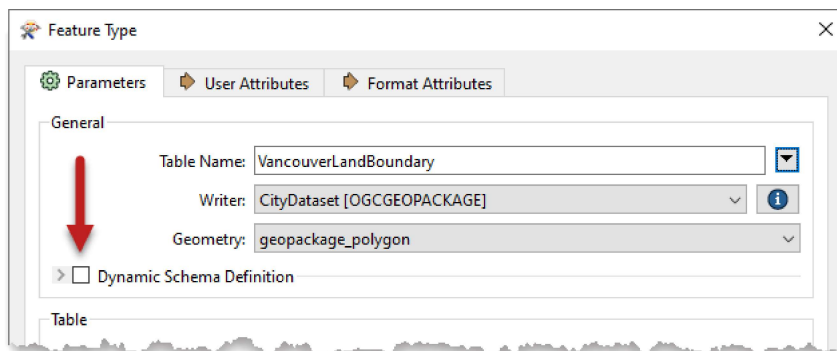


Notice that we're reading from two formats of data - GML and shapefile - and writing out to two tables in a GeoPackage database.

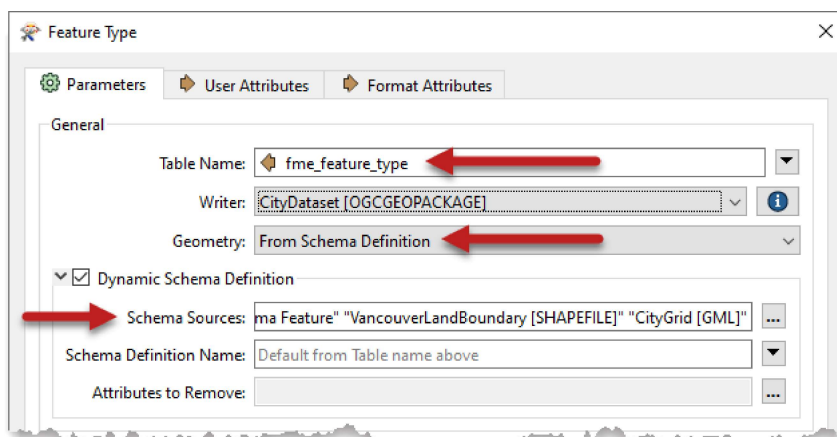
## 2. Change Writer to Be Dynamic

Let's say that we want to make the workspace dynamic so that we can read more source data with different schemas and still write it to the database.

We need to make the writer dynamic and also the readers. Let's start with the writer. Open the parameters dialog for the VancouverLandBoundary writer feature type. It will look like this:

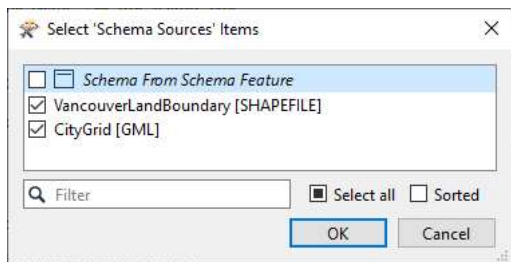


Notice how the table name and geometry are set to fixed values. But the key setting is labeled Dynamic Schema Definition. Click in that box to add a checkmark. The dialog will now change to this:



Notice how the Table Name has changed (it is now dynamic, set according to the original feature type name), that Geometry is now dynamic, and that there is a Schema Sources parameter.

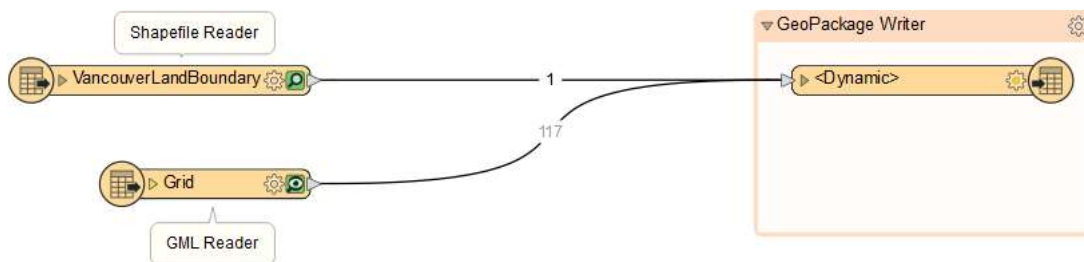
Click the ellipsis [...] button next to the Schema Sources parameter. This is where we choose from which source the schema should come from. In this case, it comes from the two readers. It does not come from Schema Features, so uncheck Schema From Schema Feature, and click OK.



Click OK again to return to the canvas.

### 3. Remove Static Writers

Now we have a <Dynamic> feature type, we don't need any remaining static ones. So, delete the Grid writer feature type and connect the Grid reader feature type to the <Dynamic> writer feature type.

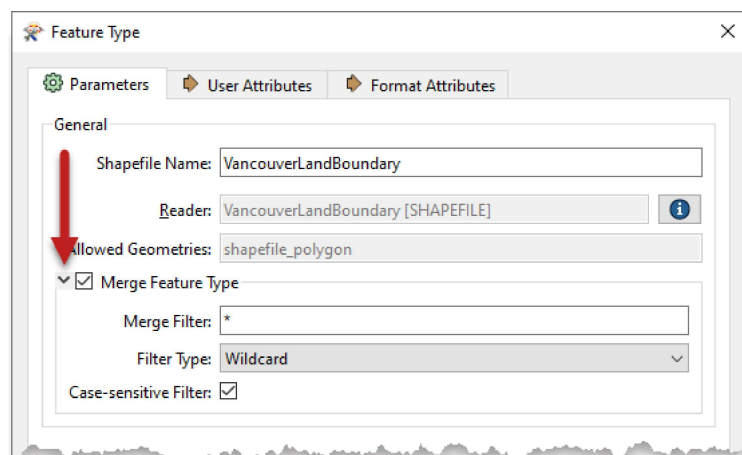


Now if you run the workspace, the output will be the same as before, except the schema is being fetched dynamically. If the source schema should change - a process called Schema Drift, which we will cover later - then the output will change to match.

It also means we can read more source datasets, as soon as we make the readers dynamic too.

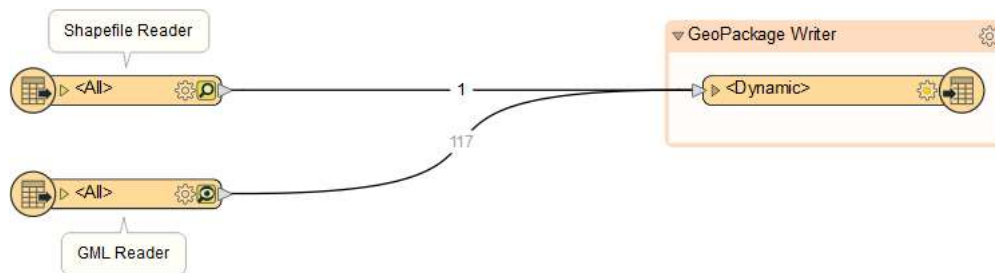
### 4. Change Readers to be Dynamic

Open the parameters dialog for the VancouverLandBoundary reader feature type. Check the box labeled Merge Feature Type:



Click OK to close this dialog. Repeat the process on the Grid reader feature type.

The name of the feature types changes to <ALL>, and so the canvas will now look like this:



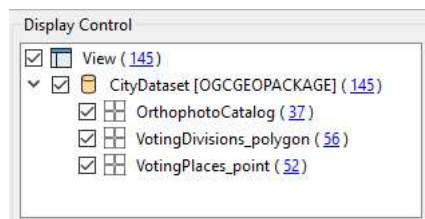
Question) Why do we need to repeat the process for both feature types? Why can't we delete one of them in the same way that we did for the writer?

## 5. Test Workspace

Now let's test how dynamic the workspace is. Make sure the Prompt for User Parameters is enabled, and re-run the workspace. This time you are prompted to pick the source dataset.

This time pick a different shapefile (say, Orthophotos\OrthophotoCatalog.shp) and a different GML dataset (say, Elections\ElectionVoting.gml). Click OK to run the workspace.

Inspect the output. Notice that the database now contains the OrthophotoCatalog data plus ElectionVoting data, in the correct structure:



In fact, because the VotingDivisions data had two different types of geometry, they've been dynamically separated into two tables (presumably, GeoPackage does not allow multiple geometry types per table).

Answer) The answer to the question is that we need two readers because the source data is in two different formats. You could replace those readers with a single Generic Format reader, if you wanted to reduce it all to a single feature type, but this method gives you an easier control of the reader parameters.

Also note that, if you want to keep adding tables to the database, instead of overwriting everything, the writer parameter Overwrite Existing Database can be set to No.

Continue to the next article: [Dynamic Workflows with the SchemaScanner](https://community.safe.com/s/article/Dynamic-Workflows-with-the-SchemaScanner)  
<https://community.safe.com/s/article/Dynamic-Workflows-with-the-SchemaScanner>

## Data Attribution

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

---

### First Published Date

7/29/2022, 5:13 PM

---

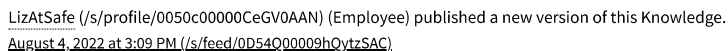
### Last Published Date

8/4/2022, 3:09 PM

---

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

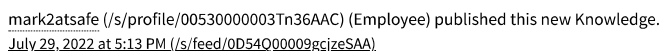
Latest Posts ▼



 Like



Log In to Comment



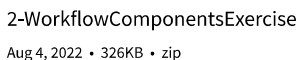
Like



Log In to Comment

Follow

Files (1) (/s/relatedlist/ka14Q000001DWymQAG/AttachedContentDocuments)

[View All](#)

(/s/relatedlist/ka140000001DWymOAG/AttachedContentDocuments)

Tutorial: Dynamic Workflows (</s/article/dynamic-workflow-tutorial-introduction>)

Dynamic Workflows: Advanced Example - Modifying the Schema Feature (/s/article/dynamic-workflows-advanced-example-modifying-the-s)

Dynamic Workflows: Destination Schema Derived from Multiple Sources (</s/article/dynamic-workflows-destination-schema-derived-from>)

Dynamic Workflows: Basics of Dynamics (/s/article/dynamic-workflow-tutorial-basics-of-dynamics)

Dynamic Workflows: Destination Schema as a Mirror Image of the Source Dataset (</s/article/dynamic-workflow-tutorial-destination-schema-as-a>)



**SAFE SOFTWARE®**

(<https://safe.com>).

[Forums \(/s/forums/\)](#)

[Groups](#)

[Knowledge Base \(/s/knowledge-base/\)](#), ([/s/group/CollaborationGroup/00Ba000000A0BxJEAV](#))

[Support \(/s/support/\)](#)

[Register / Log In \(/s/login/\)](#)



(<http://difficult.com/safe.com/legal/>)

© Safe Software Inc | [Legal \(https://www.safe.com/legal/\)](#)

**Land Acknowledgement —**

Safe Software respectfully acknowledges that we live, learn and work on the traditional and unceded territories of the Kwantlen, Katzie, and Semiahmoo First Nations.