FME Form Modules 1-5

Exercise Workbook

info@misoportal.com

0121 232 8000

Contents

1 Getting Started with FME

1.1 The Data Inspector and Generate Workspace Wizard

Demonstrates	Use of the Data Inspector to display and interrogate data. Create and run a simple workspace using the Generate Workspace Wizard within the Workbench.
Overall Goal	Familiarise yourself with the zoning data before processing. Create a workspace to translate the zoning data from MapInfo TAB format to GeoJSON (Geographic JavaScript Object Notation). Inspect the output of the translation.
Data	Zoning Data (MapInfo TAB) Neighborhoods (Google KML)
Start Workspace	none
End Workspace	C:\FMEModularData\Workspaces\Complete\1.01-GettingStarted-GenerateWorkspace-Complete.fmw

Before you start manipulating and converting data, you should familiarize yourself with it. Let's see how the FME Data Inspector interface works by inspecting some zoning data which will be the input for a future exercise.

1.1.1 Start FME Data Inspector

Start the FME Data Inspector by selecting it from the Windows start menu. You'll find it under Start > FME Form (Desktop) > FME Data Inspector

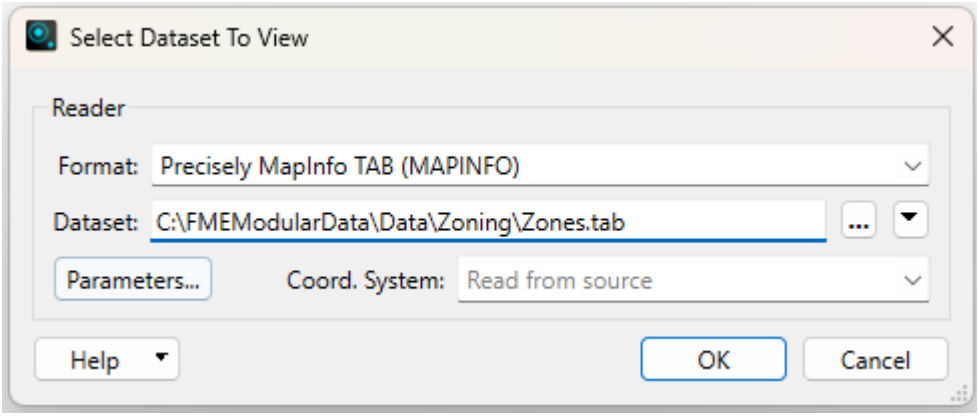
The FME Data Inspector will start up and begin with an empty view display.

1.1.2 Open Dataset

Use File > Open Dataset from the menu bar OR use the Open button

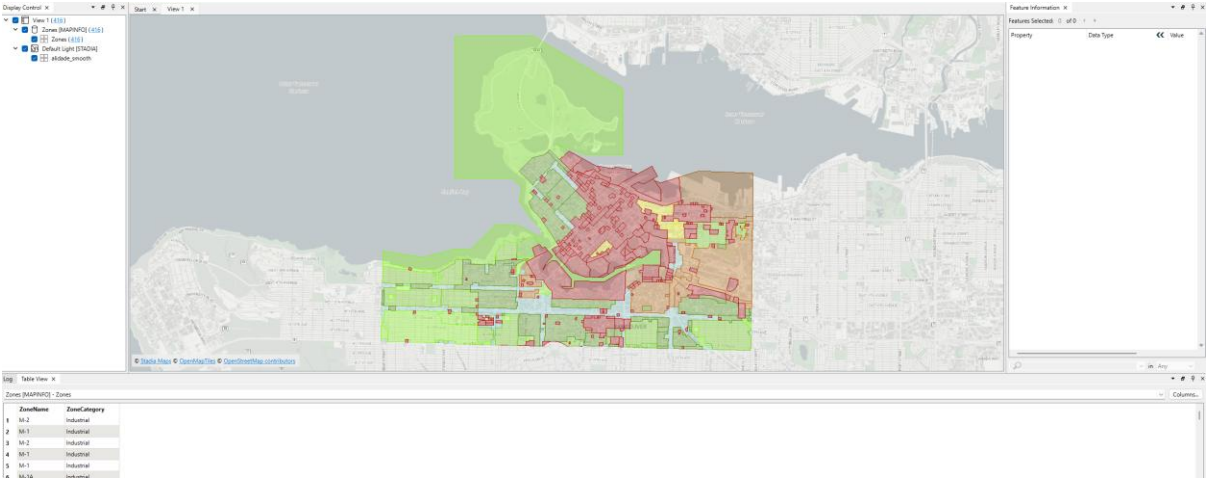
When prompted, fill in the fields inthe Select Dataset dialog as follows:

Reader Format	Precisely MapInfo TAB (MAPINFO)
Reader Dataset	C:\FMEModularData\Data\Zoning\Zones.tab



Click OK

The Zoning MapInfo dataset looks like this:



Note: We'll see in a later step how to add background mapping when required. However, if your session has automatically added background mapping, don't worry you can leave it switched on. Or if you prefer, switch it off.

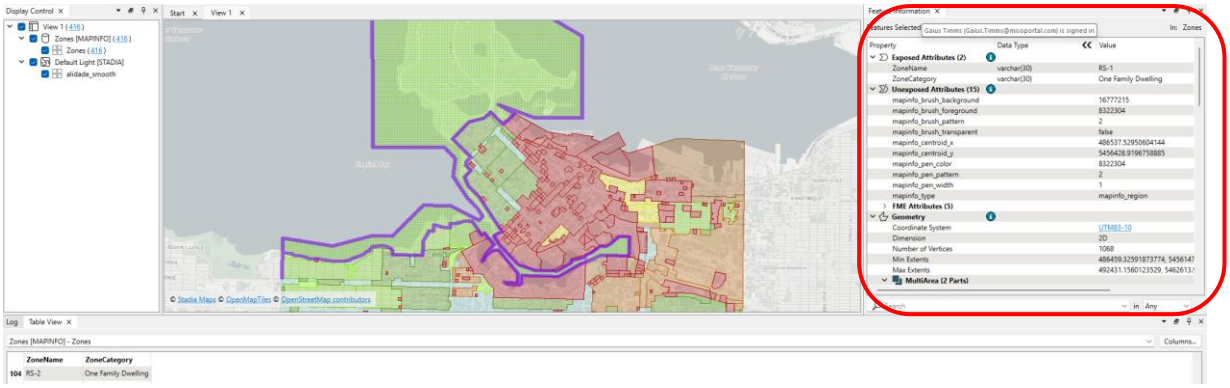
1.1.3 Browse Data

Use both the Table View and the Map View to browse through the dataset, inspecting it closely.

Use the Select Tool 

to *select a single feature* from either the Table of Map views.

Then inspect the information displayed in the Feature Information Window.



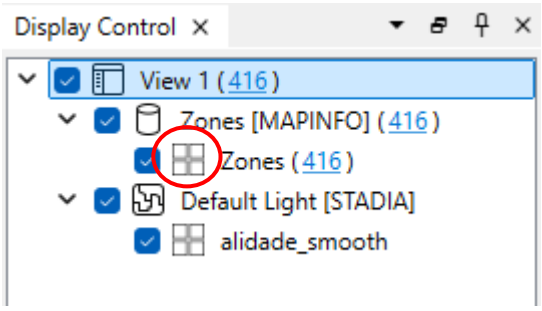
Notice that each feature has a ZoneCategory of either: *Commercial, Comprehensive Development, Historic Area, Industrial, Light Industrial, Multiple Family Dwelling, One Family Dwelling* or *Two Family Dwelling*.
This will be useful in a future exercise!

Tip

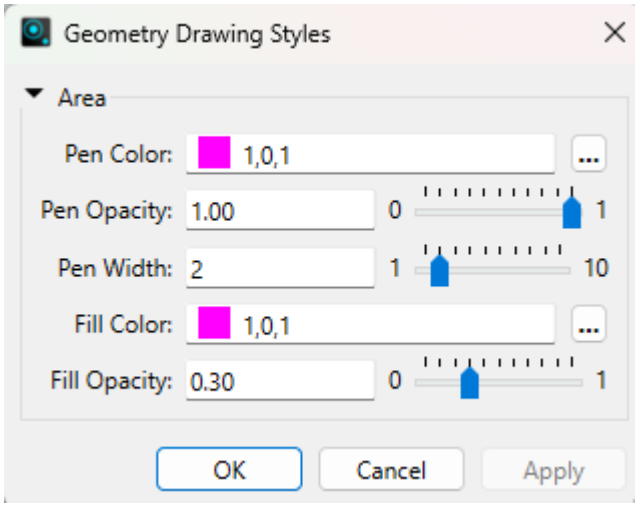
Use the Search function at the bottom of the Table View to find features with specified attribute values

1.1.4 Symbology

Within the Display Control the appearance of layers can be modified. This is particularly useful when the map view contains multiple layers.
Click the Symbology Icon for the Zones data in the Display Control window:



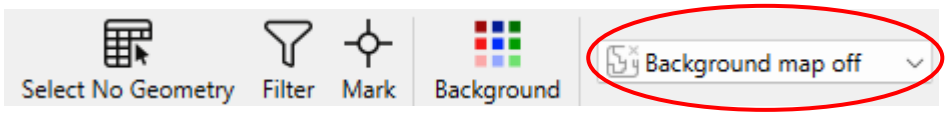
Set the Pen Colour and width for the polygon border appearance.
Set the Fill Colour and Fill Opacity for the polygon fill appearance.



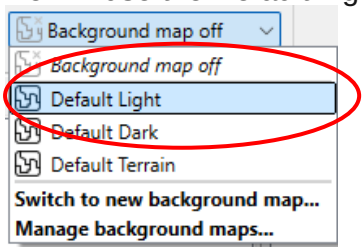
then click OK to apply

1.1.5 Add Background Map

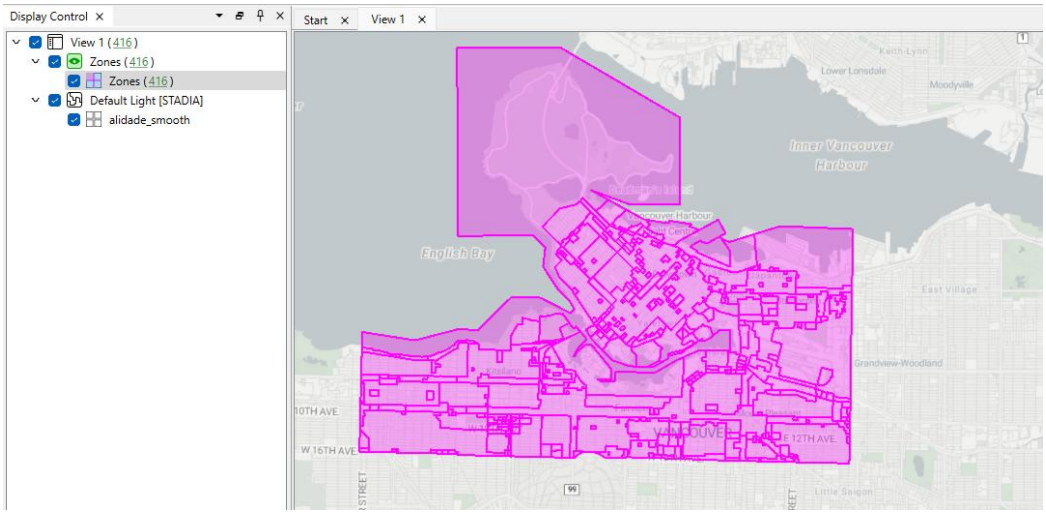
When inspecting data it will help to have a background map to provide a sense of location. You can choose from many mapping services. **If** the background map hasn't been selected by default, click the 'Background Map' drop down in the Visual Preview Graphics window toolbar:



In the drop down menu, select which background you would like to use, for this example we will use the Default Light option.

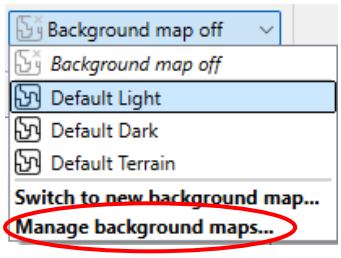


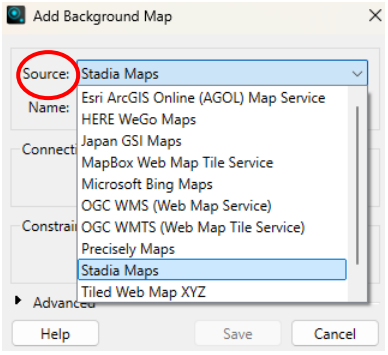
A background map is added to the display.



Notice that the data is reprojected to match the coordinate system of the chosen background.

We also have the option to use backgrounds outside of the default ones, however these do require licenses for the various providers (ESRI, Bing Maps etc.). We can do this by selecting 'Manage Background Maps' and then selecting an option from the 'Source'





1.1.6 Launch FME Workbench

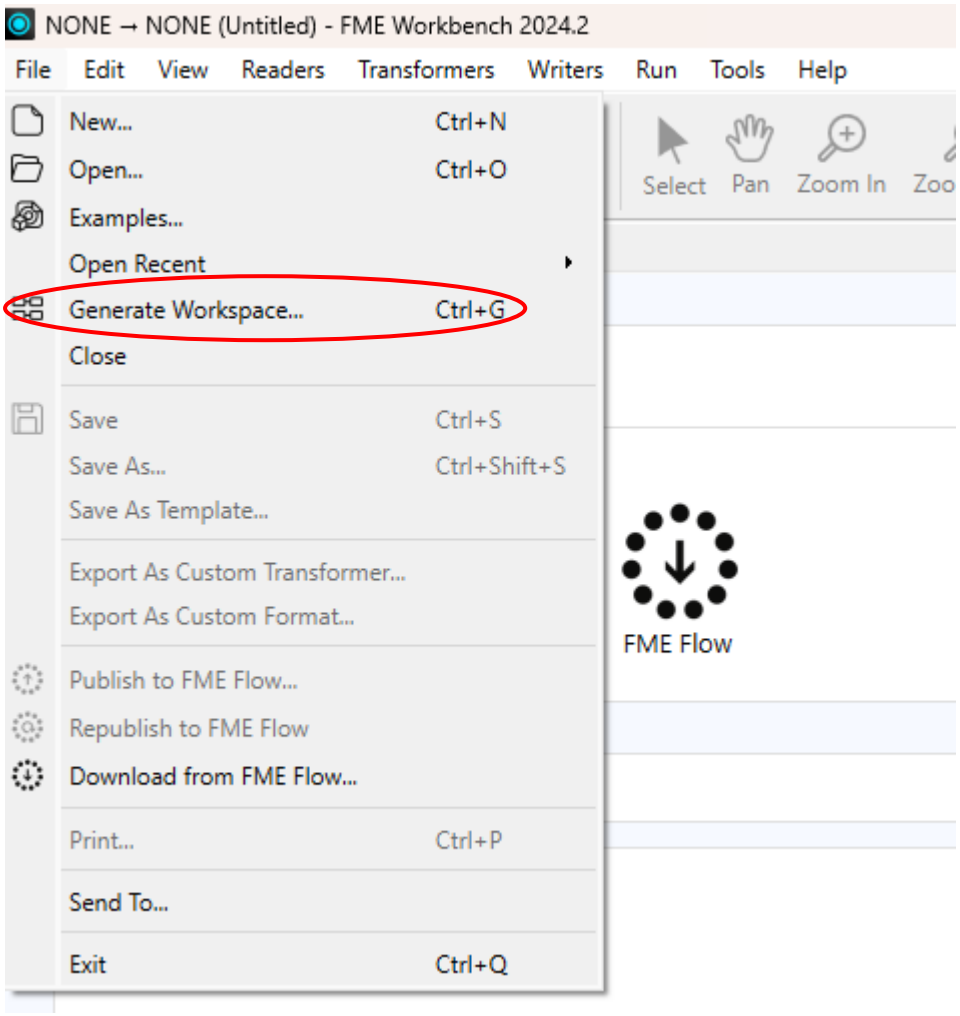
You are now going to create and run a workspace to translate the zoning data from MapInfo TAB format to GeoJSON (Geographic JavaScript Object Notation).

Start FME Workbench by selecting it from the Windows start menu. You'll find it under Start > FME Desktop > FME Workbench

1.1.7 Select Generate Workspace

You are going to use the *Generate* wizard to setup the workspace.

Within the File drop down menu of the Workbench, click on Generate Workspace or hit Ctrl + G as the shortcut.

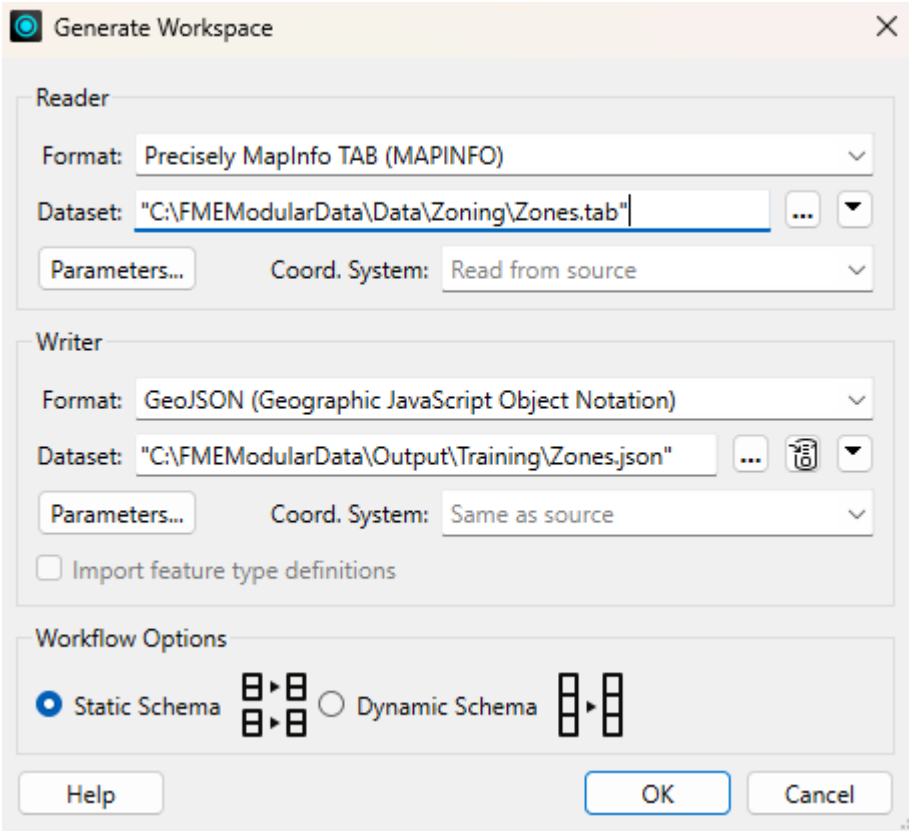


1.1.8 Define Translation

The Generate Workspace tool opens up a dialog in which to define the translation to be carried out. Fill in the fields in this dialog as follows:

Reader Format	Precisely MapInfo TAB (MAPINFO)
Reader Dataset	C:\FMEModularData\Data\Zoning\Zones.tab
Writer Format	GeoJSON (Geographic JavaScript Object Notation)
Writer Dataset	C:\FMEModularData\Output\Training\Zones.json

The dialog will look like this:



Remember, you can set a format by typing its name, by selecting it from the drop-down list, or by choosing“More Formats” and selecting the format from the full table of formats. For now, ignore the Workflow Options and leave the default of 'Static Schema.'

1.1.9 Generate and Examine Workspace

Click OK to close the Generate Workspace dialog. A new workspace will be generated into the FME Workbench canvas.

1.1.10 Save the Workspace

Click on the Save button 

or use File > Save As on the menu bar. Then navigate to where you’d like to save your workspace file: C:\FMEModularData\Output\Workspaces

Give your workspace a name.

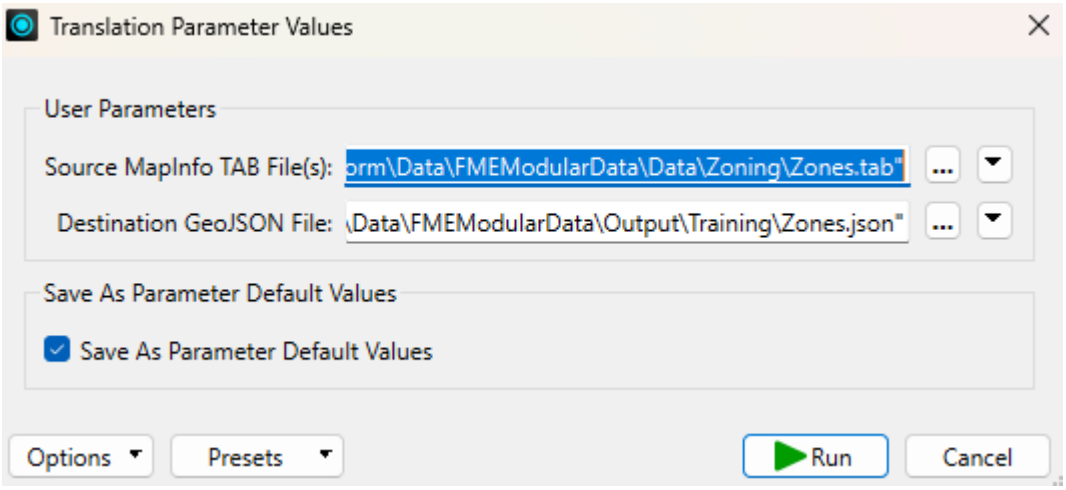
Notice the workspace file has the file extension .fmw
Click Save

1.1.11 Run Workspace

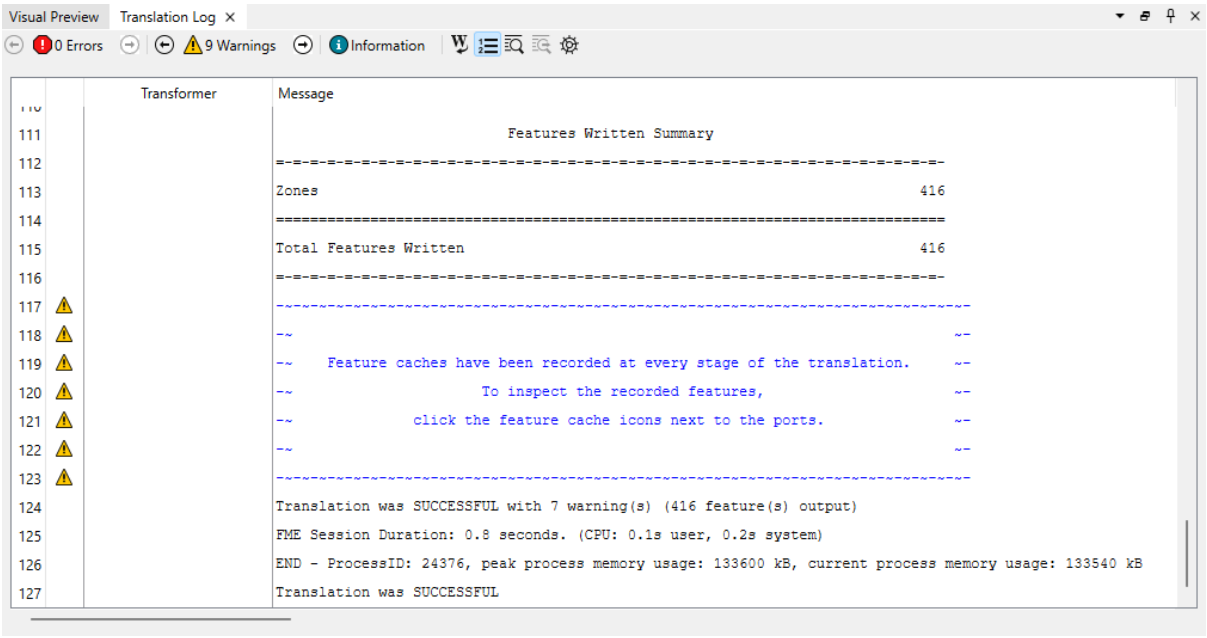
Run the workspace by clicking the run button on the 

toolbar, or by using Run > Run Workspace on the menu bar.

The Translation Parameters Values dialog may appear. This is useful to change the source or destination each time the workspace is rerun (if appropriate). Click Run to proceed.

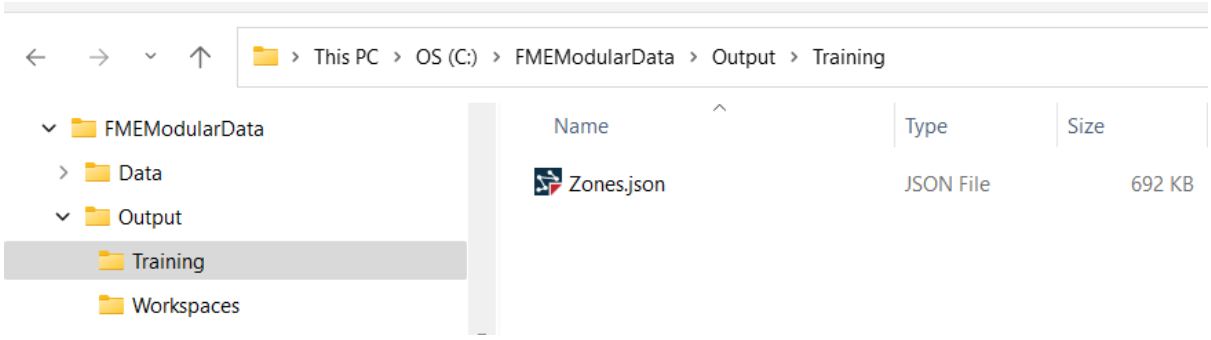


The workspace runs and the Translation Log file reports a successful translation:



1.1.12 Locate Output

Locate the output GeoJSON file using Windows Explorer



We'll now inspect the dataset visually to ensure that it is correct.

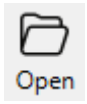
1.1.13 Start FME Data Inspector

Start the FME Data Inspector by selecting it from the Windows start menu. You'll find it under Start > FME Desktop > FME Data Inspector

The FME Data Inspector will start up and begin with an empty view display.

1.1.14 Open GeoJSON Dataset

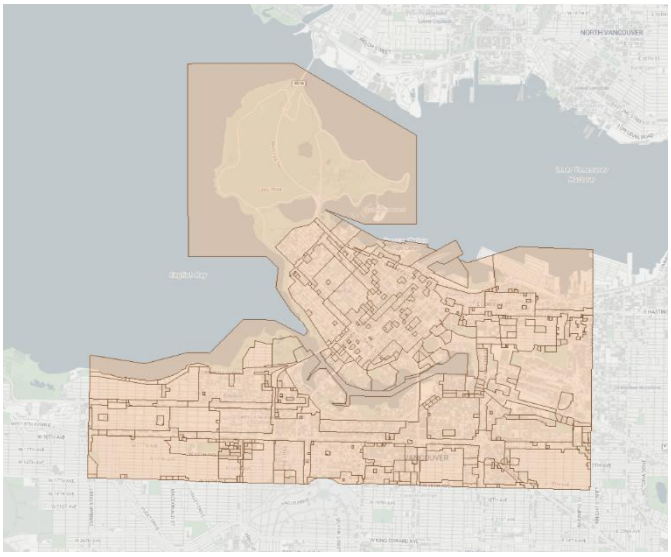
Use File > Open Dataset from the menu bar OR use the Open button



When prompted, fill in the fields inthe Select Dataset dialog as follows:

Reader Format	GeoJSON (Geographic JavaScript Object Notation)
Reader Dataset	C:\FMEModularData\Output\Training\Zones.json

The GeoJSON dataset looks like this:



1.1.15 Add Dataset

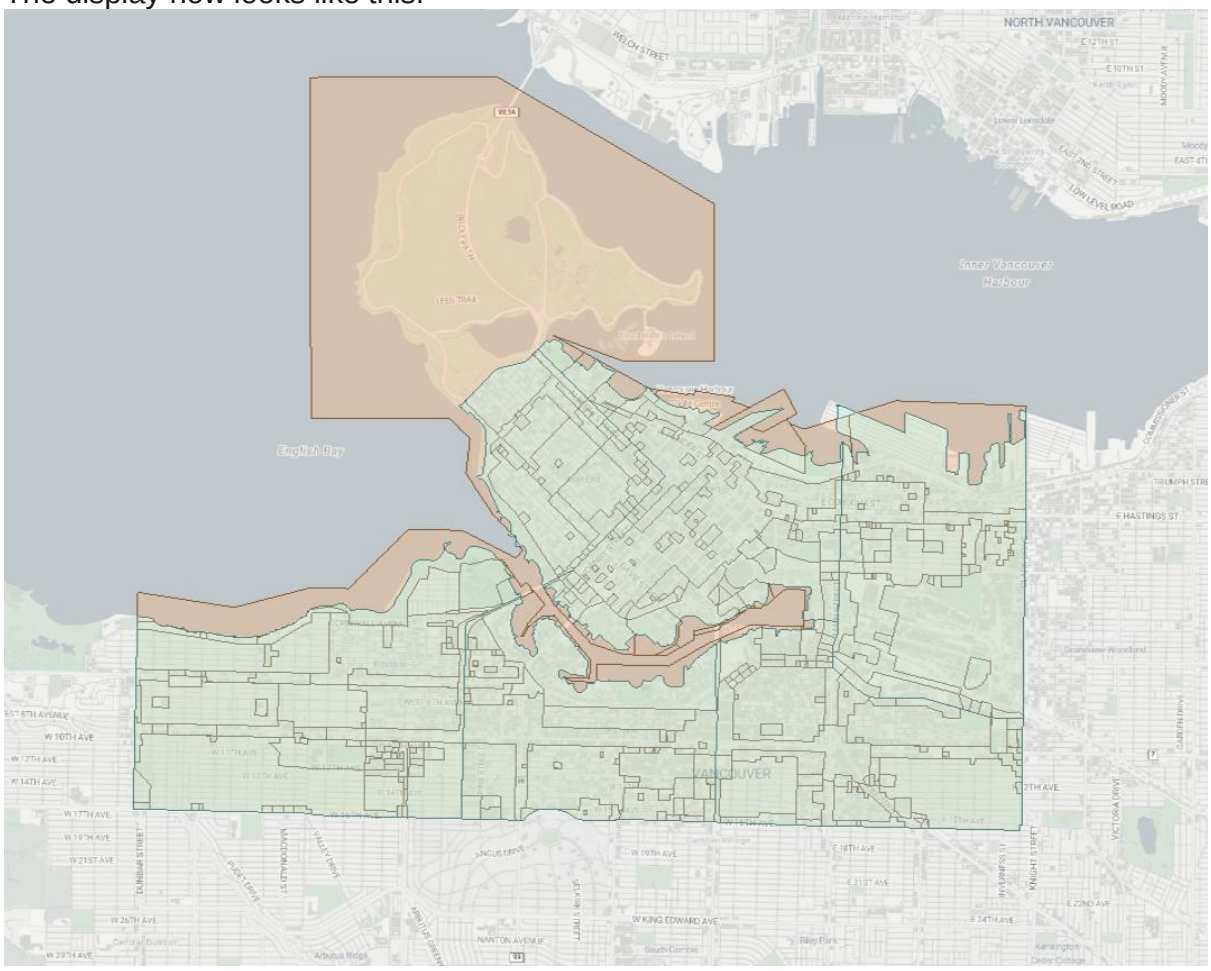
Let's add a second dataset to the display to compare to our zoning data. This dataset will be a KML file of neighborhood boundaries. Then we'll be able to see which neighborhood each zone overlaps.

To add a dataset, select File > Add Dataset from the menu bar. Or use the button

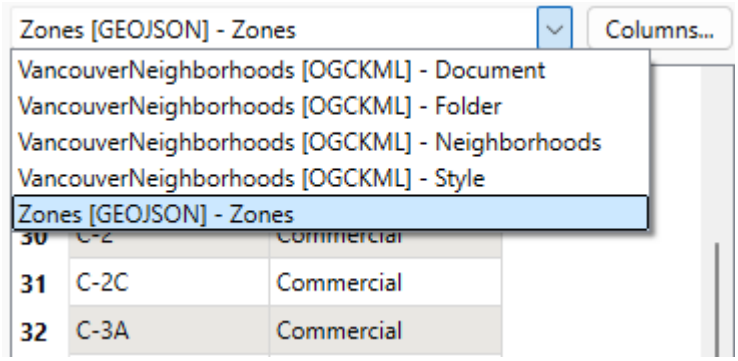
When prompted, fill in the fields in the Select Dataset dialog as follows:

Reader Format	OGC / Google KML
Reader Dataset	C:\FMEModularData\Data\Boundaries\ VancouverNeighborhoods.kml

The display now looks like this:



Use the Table View to practice inspecting the tabular data for each feature type. Click on the dropdown arrow at the top of Table View and switch back and forth between the Zones.json and Neighborhoods.kml tables:



Congratulations

By Completing this exercise you have learned how to:

- Use the Data Inspector to examine datasets
- Set symbology for inspected data
- Set a background map for inspecting data
- Create a workspace using the Generate Workspace Wizard
- Save a workspace (.fmw file)
- Run a workspace
- Examine multiple datasets within the same map view of the Data Inspector

1.2 Create a Workspace using Transformers and Visual Preview

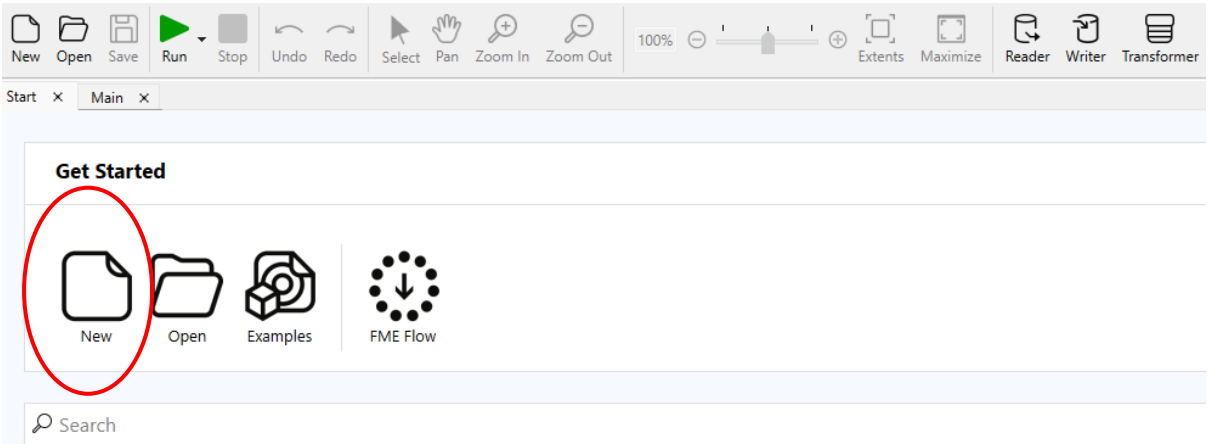
Demonstrates	Create a new workspace from a blank canvas, setting up required reader and writer. Use a transformer to perform data transformation. Use the Visual Preview tools within the Workbench to examine features at each stage of the workflow.
Overall Goal	Transform the MapInfo TAB zoning data into merged zone polygons and output to GeoPackage format
Data	Zoning Data (MapInfo TAB)
Start Workspace	none
End Workspace	C:\FMEModularData\Workspaces\Complete\1.02-GettingStarted-Transformers-Complete.fmw

We are going to create a new workspace (from a blank canvas) and use a transformer to merge the zones into new zone polygons based on their zone category.
We'll use the Visual Preview tools within the Workbench to examine the features as we design the workflow.

1.2.1 Launch FME Workbench and Create new Workspace from Blank

Launch the FME Workbench, if it isn't open already.

Within the Get Started section of the Workbench, click on New Workspace.



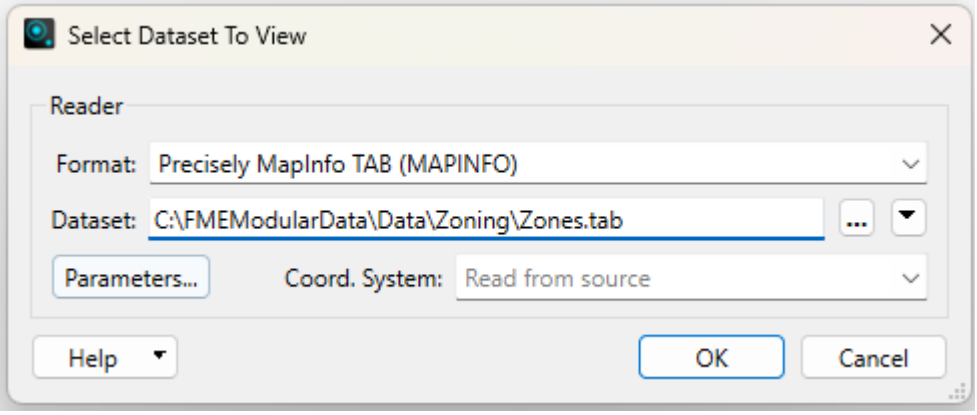
Your starting point for the workspace is a blank canvas, to which you will need to manually add the required Reader and Writer.

1.2.2 Add Reader

Use either the Reader button 

Or Readers > Add Reader… from the menu bar
The Add Reader dialog will open, in which define the Format and Dataset settings as follows:

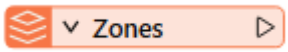
Reader Format	Precisely MapInfo TAB (MAPINFO)
Reader Dataset	C:\FMEModularData\Data\Zoning\Zones.tab



We'll look at Parameters and Workflow Options later. Leave them as default for now.

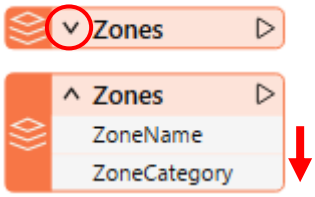
Click OK

The Zones data will be added to the canvas

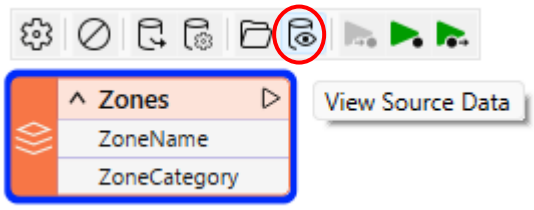


1.2.3 Examine the Zones data

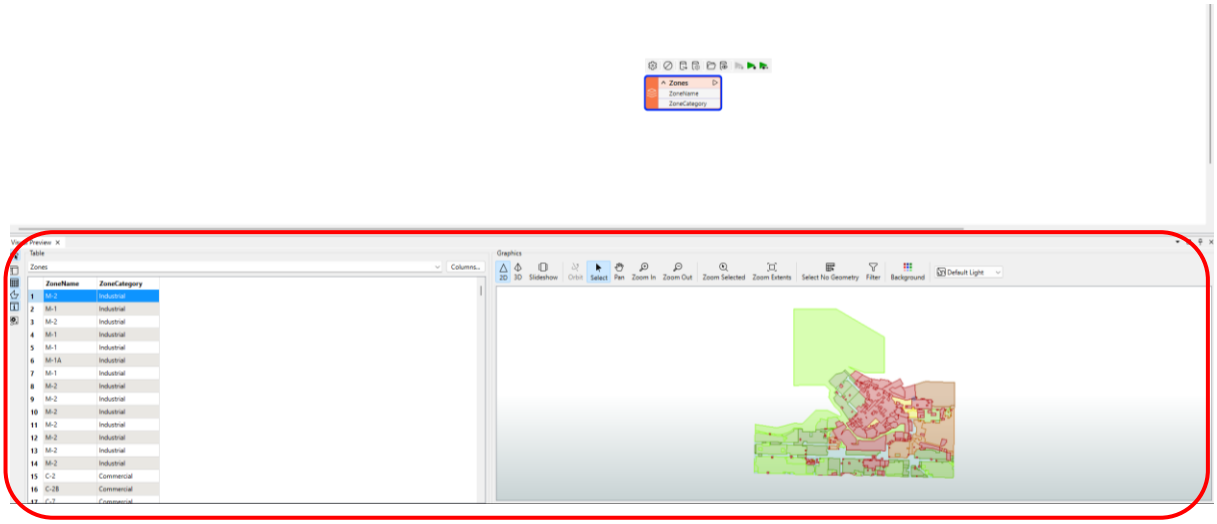
Expose the attribute list by clicking on the arrow.



Let's examine the features. Click on the Zones, so that it becomes highlighted in blue. Then click on the View Source Data button.



Within the Visual Preview section of the Workbench, the features can be viewed.



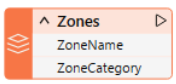
Use the Visual Preview toggle buttons to show/hide the Table view, Graphics view and Feature Information view.

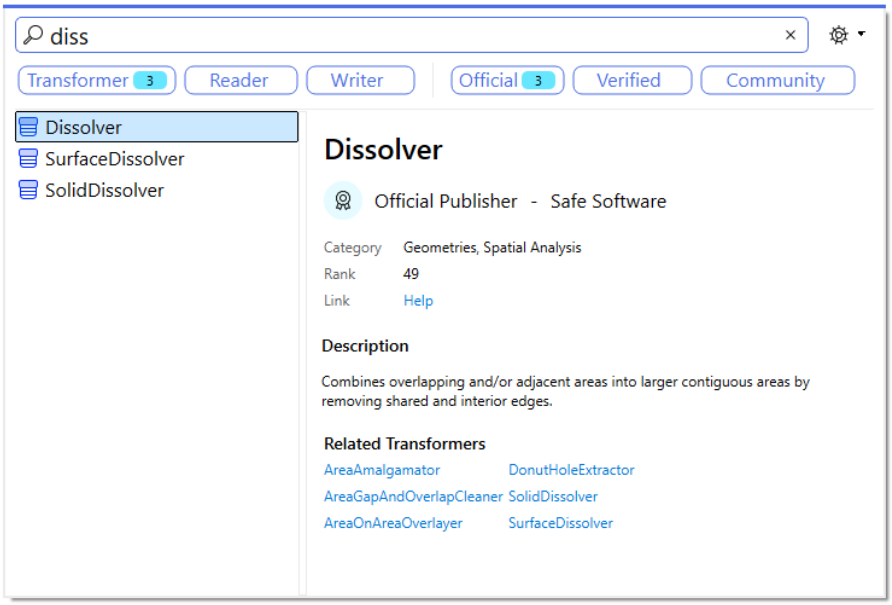
Notice that each feature has a ZoneCategory of either: *Commercial, Comprehensive Development, Historic Area, Industrial, Light Industrial, Multiple Family Dwelling, One Family Dwelling* or *Two Family Dwelling*.

1.2.4 Add a Transformer

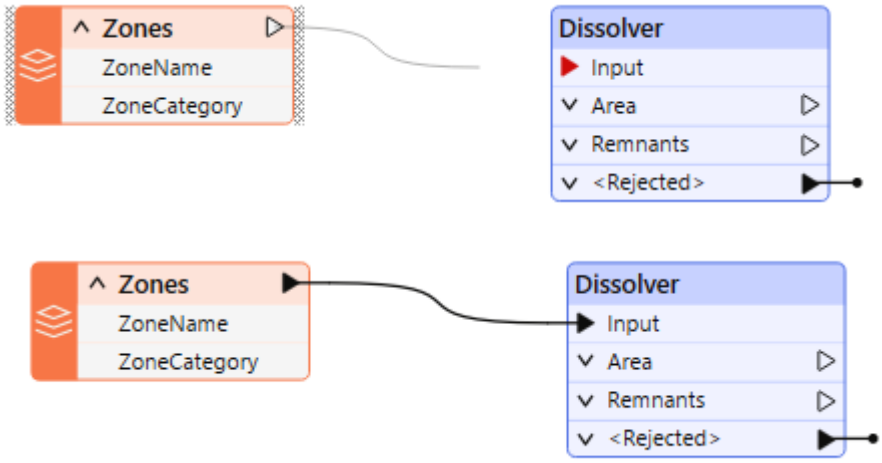
Now we are going to add a transformer, which we'll use to dissolve the zones to form new zones based on their zone category.

Click on the canvas. Nothing will appear, this just makes the canvas active. Then start typing the name of the required transformer; "Dissolver".





When you see the Dissolver transformer appear in the list, double-click on it to place it onto the canvas.
You now need to connect the Zones to the Dissolver transformer, by dragging a connector line between the two.



1.2.5 Save the Workspace

Click on the Save button, or use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces

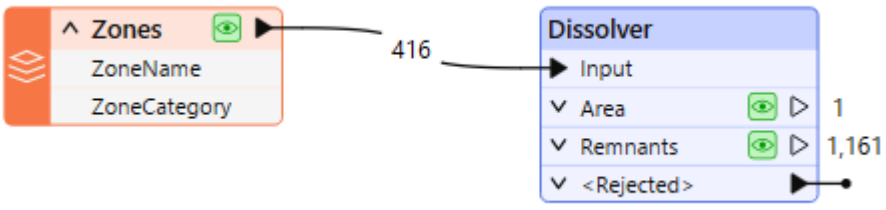
Give your workspace a name.
Click Save

1.2.6 Run Workspace

Run the workspace by clicking the run button on the toolbar,
or by using Run > Run Workspace on the menu bar.

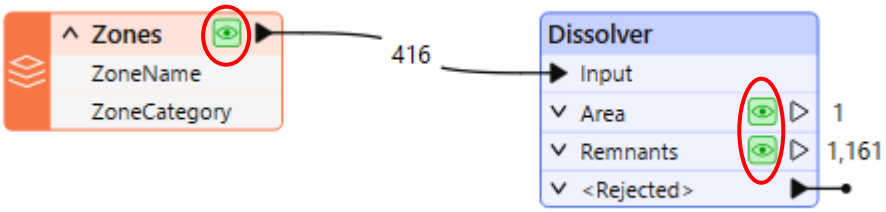
The Translation Parameters dialog may display. Click Run to proceed.

When the workspace runs successfully it will look similar to this, and feature counts will appear on the connector line and output ports of the transformer:



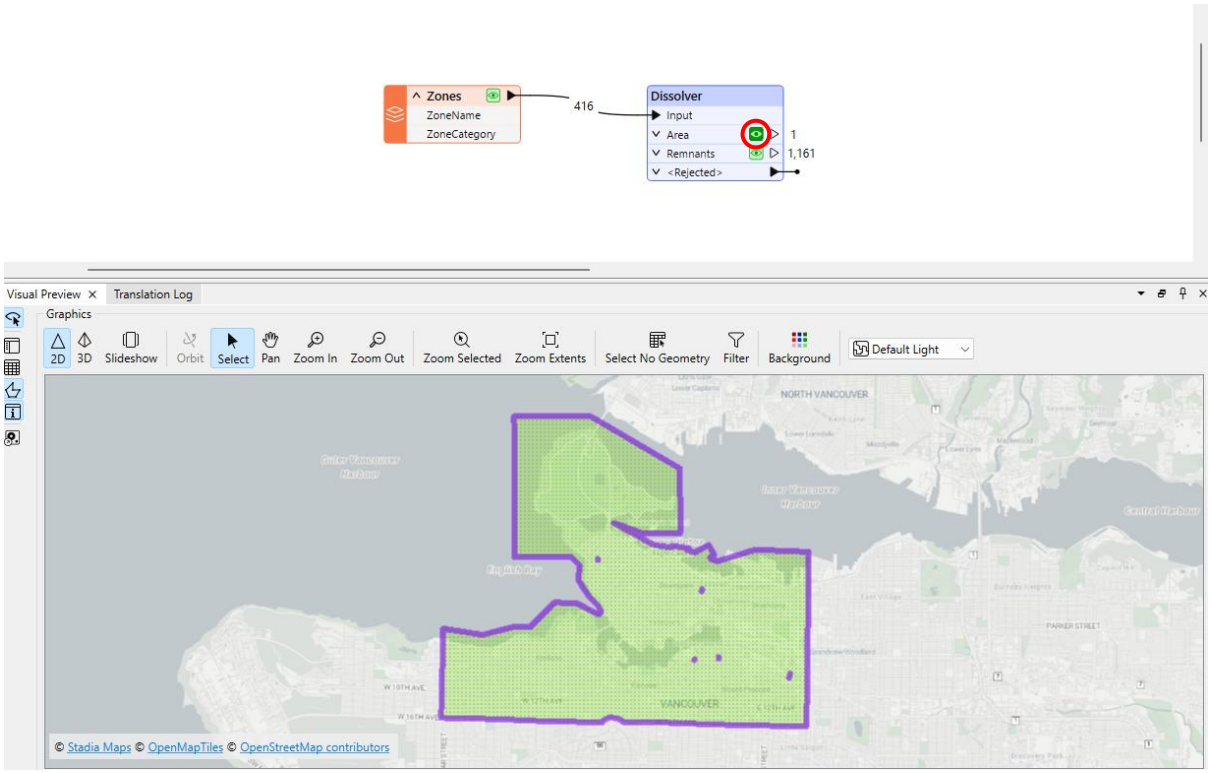
1.2.7 Inspect the Feature Caching using Visual Preview tools

The workbench will cache the zone features at each point of the workflow; the original zone features and two caches on the Dissolver: Area and Remnants.



Click on the Area feature cache to examine the new dissolved polygons (area).

All the zone polygons have been dissolved into a single new polygon – *this isn't quite what we were looking for. We'll need to modify the parameters within the Dissolver transformer in a moment.*

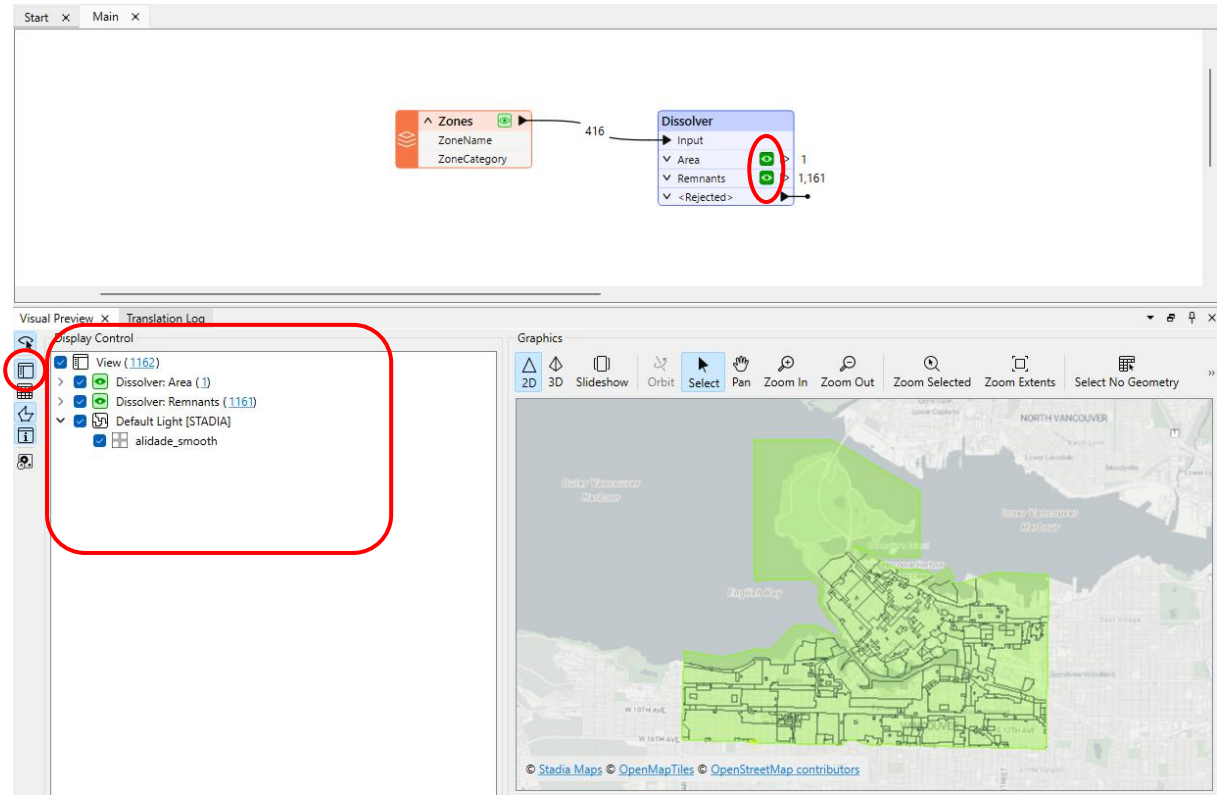


1.2.8 Visual Preview Display Control window

Let's examine the feature caches for both Area and Remnants at the same time – displaying them together in the same Graphics view.

Holding down the 'Ctrl' key on the keyboard, *click on both* the *Area* and *Remnants* feature cache buttons.

Both feature caches will then be displayed in the same graphics view.



The Display Control can be toggled on (if not already) and used to change the styling of each map layer. Here each map layer can also be turned on and off.

You could also compare the before and after results by selecting the original Zones feature cache as well!

1.2.9 Add Background Map

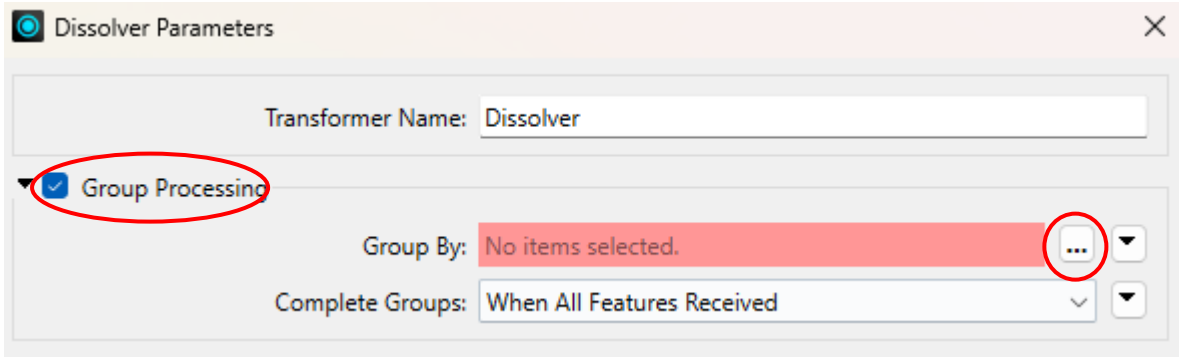
When inspecting data it will help to have a background map to provide a sense of location. You can choose from many mapping services. If it does not have a background map by default, follow the same process from Section 1.1.5

1.2.10 Configure Transformers Parameters

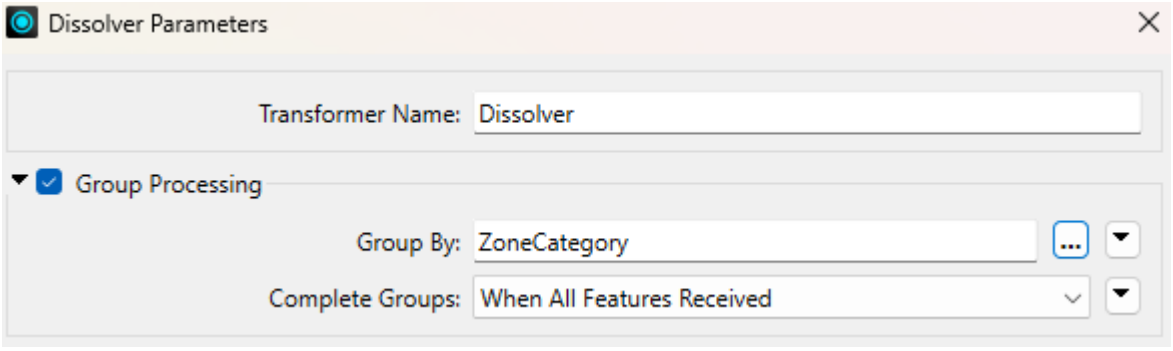
Currently the Dissolver transformer is dissolving (merging) all the zones into a single new zone polygon. Instead we only want touching zones of the same Zone Category to merge.

Click on the transformer's parameters cog (or double-click on the transformer titlebar).

Within the Dissolver Parameters *tick* for *Group Processing*.



Then for the Group By: parameter use the *ellipse button* to specify the attribute ZoneCategory



Click OK

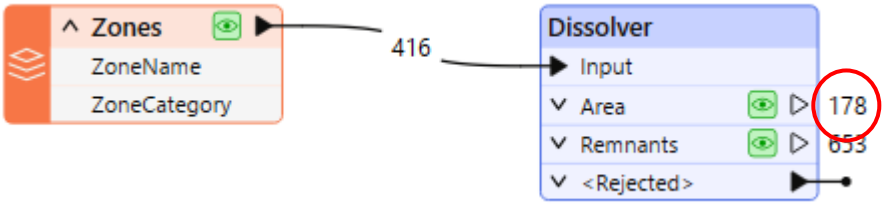
1.2.11 Save and Run Workspace

Resave the workspace, then Run the workspace by clicking the Run button on the toolbar, or by using Run > Run Workspace on the menu bar.

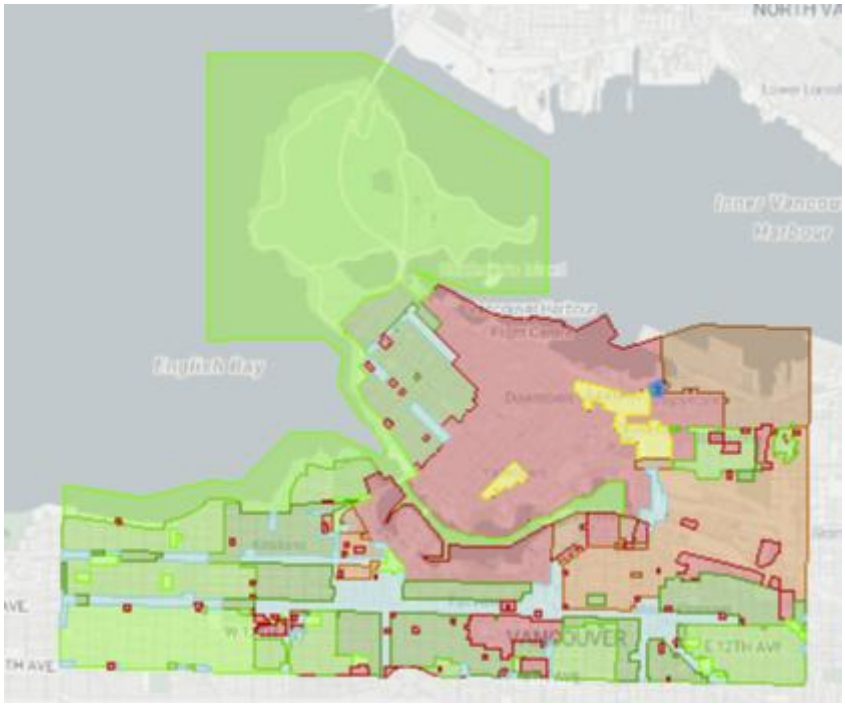
The Translation Parameters dialog may display. Click Run to proceed.

1.2.12 Inspect the Dissolver Feature Cache

Click on the Dissolver transformer Area feature cache to examine the new dissolved polygons (area).



This time the Dissolver transformer has dissolved/merged touching polygons which have the same ZoneCatogory value; creating 178 zones.



Now that the workflow design is as we intended, we are now ready to write our data out to a GeoPackage.

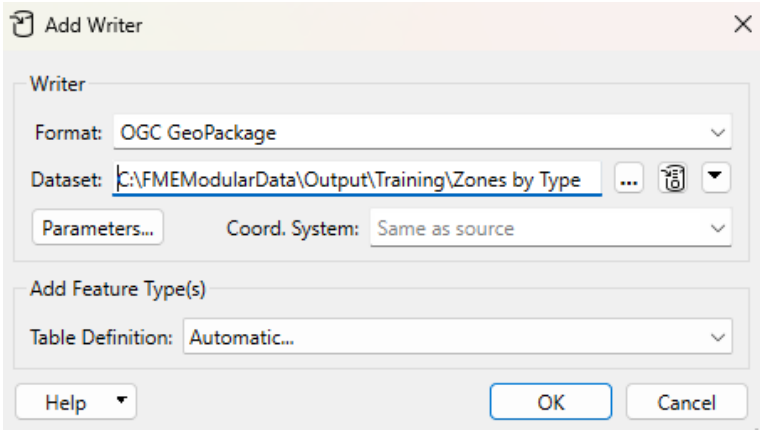
1.2.13 Add a Writer

Use either the Writer button

Or use Writers > Add Writer… from the menu bar

The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	OGC GeoPackage
Writer Dataset	"C:\FMEModularData\Output\Training\Zones by Type.gpkg"

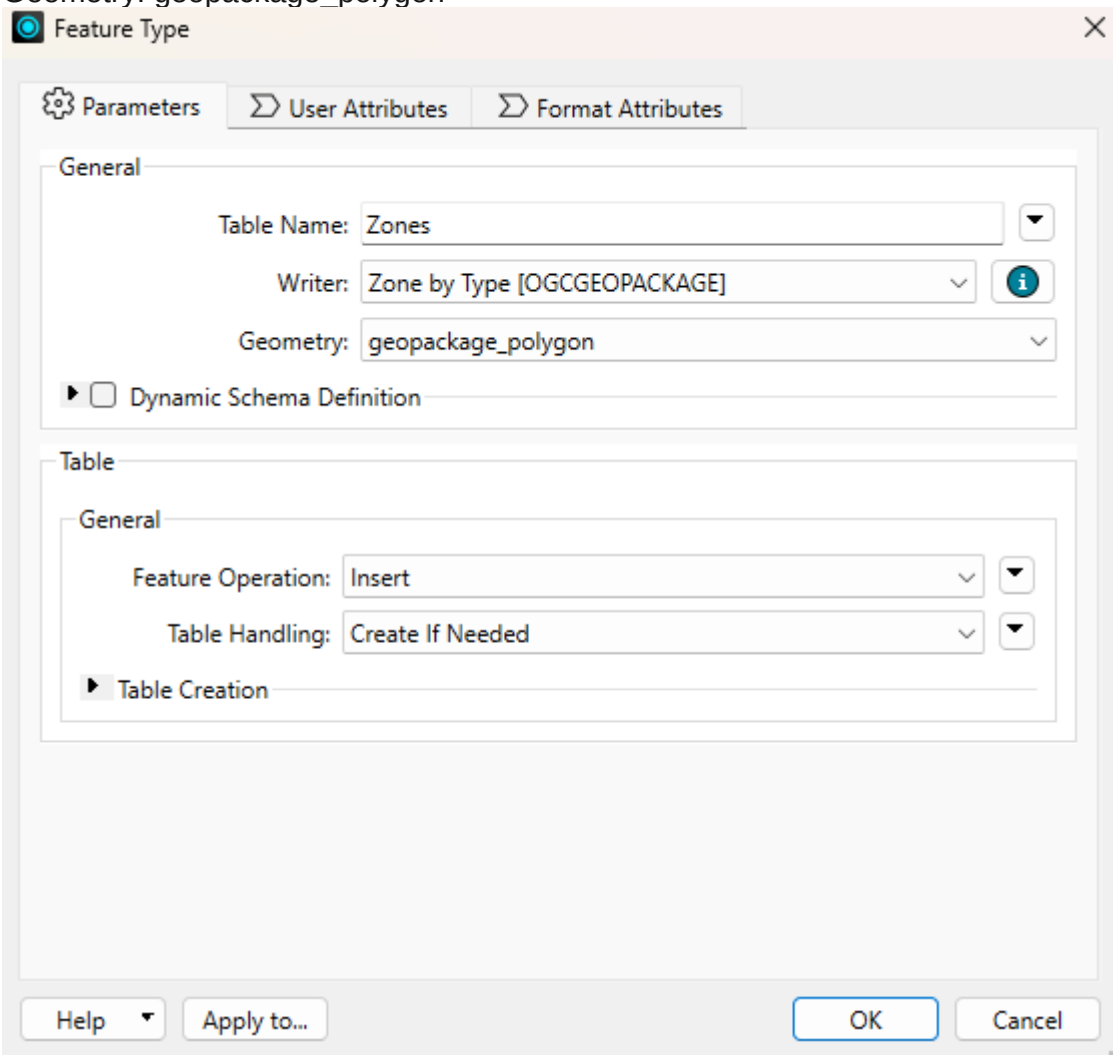


Also set the Table Definition: to Automatic…

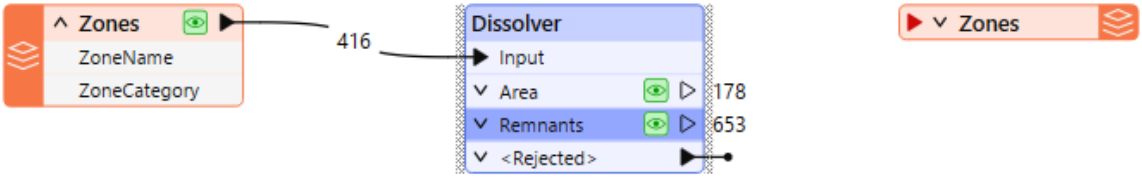
- *We'll explain this in a later module!*

Click OK

A dialog will open, in which set the following parameters:
Table Name: Zones
Geometry: geopackage_polygon

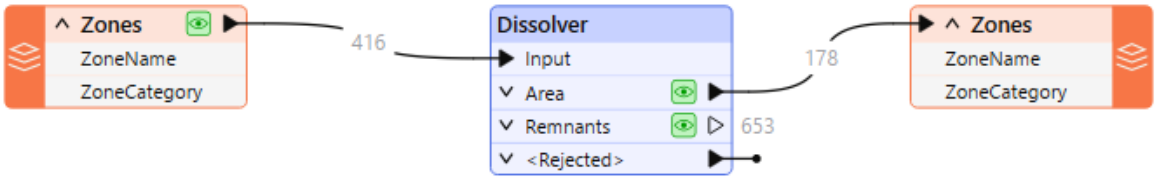


Click OK. Our Geopackage Writer will be added to the workspace.



1.2.14 Connect to the workflow

Connect the Dissolver transformer Area output port to our Zones output.



1.2.15 Save and Run Workspace

Resave the workspace, then Run the workspace by clicking the Run button on the toolbar, or by using Run > Run Workspace on the menu bar.

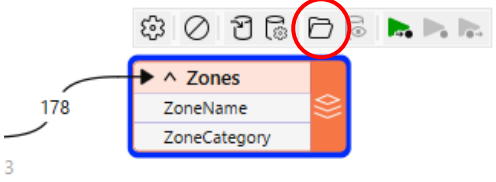
The Translation Parameters dialog may display. Click Run to proceed.

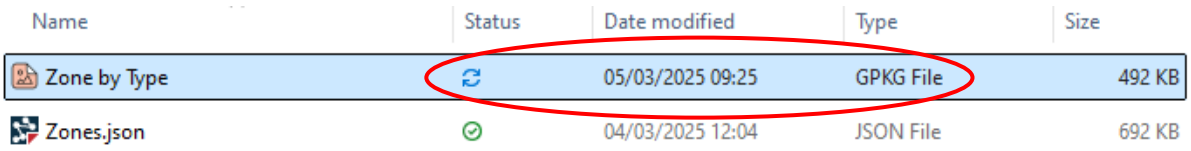
The Translation Log should confirm that the translation was successful.

1.2.16 Examine the output

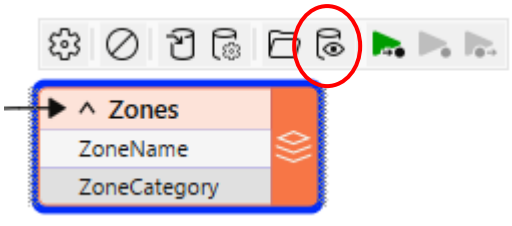
We can examine the output in a few of ways:

- Use the **Open Containing Folder** button to find the GeoPackage file within Windows Explorer. Then open the GeoPackage file using the **Data Inspector**

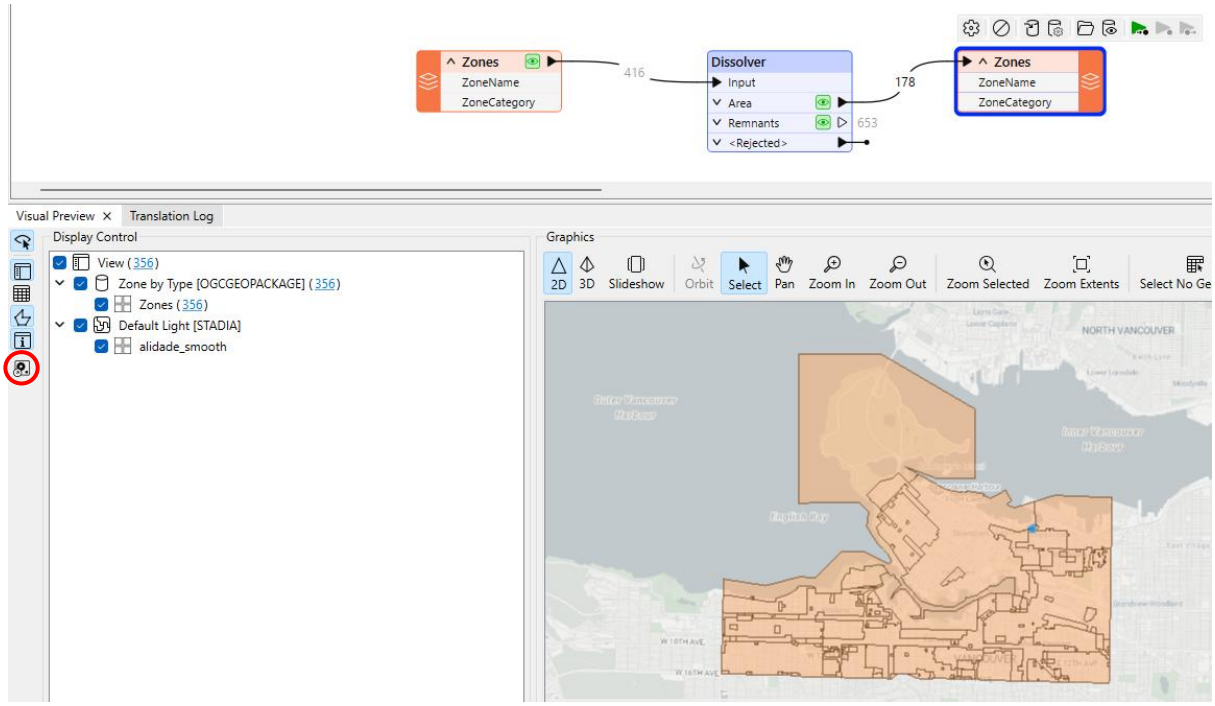




- Or use the **View Written Data** button, then examine the features using the **Visual Preview** tools within the Workbench



- Or with the Zones output highlighted in blue, within the **Visual Preview** click on the **button** called **Open in Data Inspector**



Congratulations

By Completing this exercise you have learned how to:

- Create a workspace from Blank, manually setting up the required Reader(s) and Writer(s)
- Use the Visual Preview tools to examine data within the Workbench
- Add a Transformers to your workspace
- Configure transformer parameters

1.3 FME Components and Hierarchy

Demonstrates	Differences in FME hierarchy regarding the Readers, Writers and Feature Types. Working with multiple Feature Types.
Overall Goal	Create a workspace to translate Bentley DGN data to ESRI Shapefile
Data	CommunityPlanMap (Bentley MicroStation Design V8)
Start Workspace	none
End Workspace	C:\FMEModularData\Workspaces\Complete\1.02-GettingStarted-Hierarchy-Complete.fmw

We have a Bentley DGN file containing multiple layers of data (similar to AutoCAD data), which we need to convert to several ESRI Shapefile datasets.

Before we design our workspace, we'll first examine the Bentley DGN data using the Data Inspector.

1.3.1 Open the Bentley file into the Data Inspector

Launch the Data Inspector, if it's not already open.

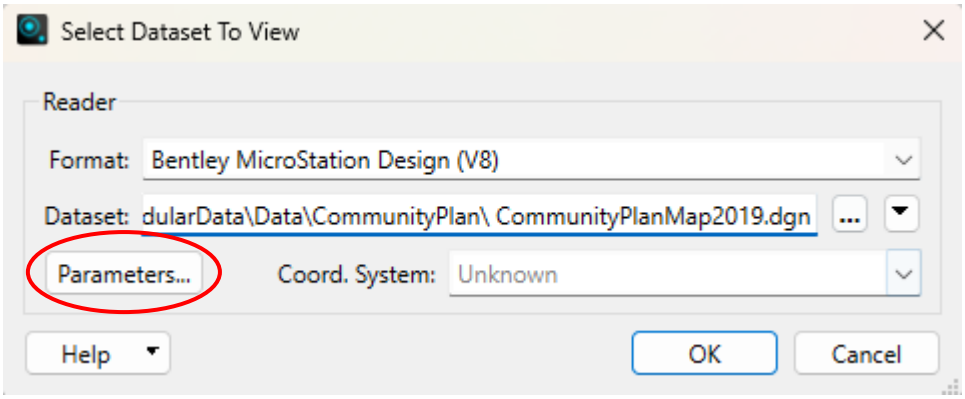
Use the Open Dataset button from the 'Get Started' section 

OR use the Open button 

on the top toolbar

When prompted, fill in the fields inthe Select Dataset dialog as follows:

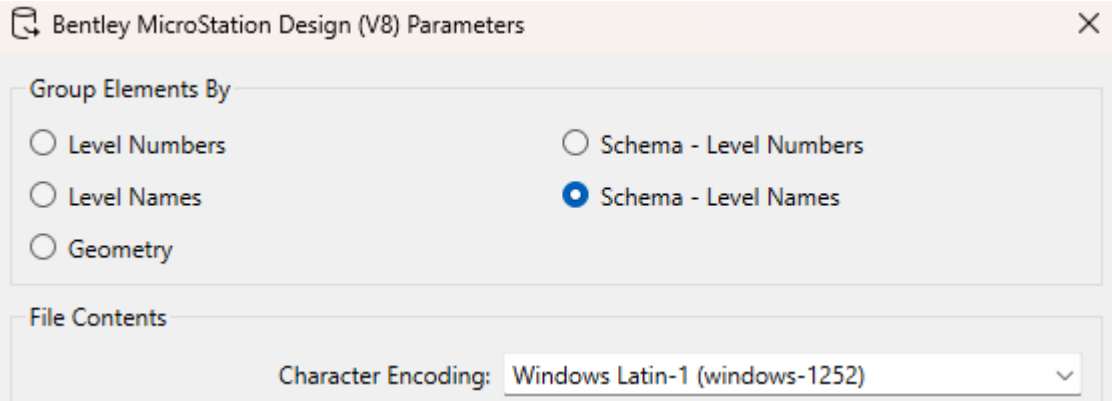
Reader Format	Bentley MicroStation Design (V8)
Reader Dataset	C:\FMEModularData\Data\CommunityPlan\ CommunityPlanMap2019.dgn



Also, click the **Parameters…** button

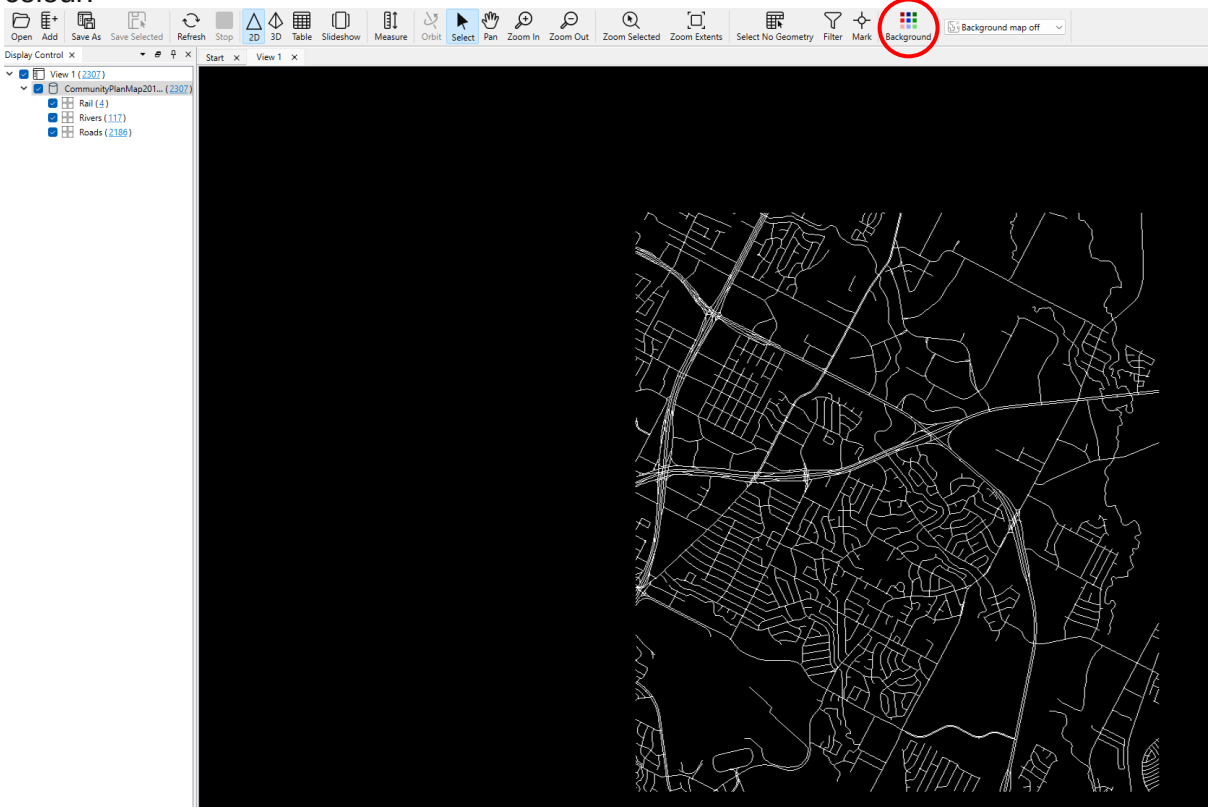
Bentley DGN files, like many formats, contain multiple layers of data. The Parameters section enables us to make choices around which data will be reading in and how certain elements/properties are exposed. When working with Bentley and AutoCAD data its important to expose the Schema Level Names (so that the layer names are available to us when working with the data).

In the Group Elements By section, choose Schema – Level Names.

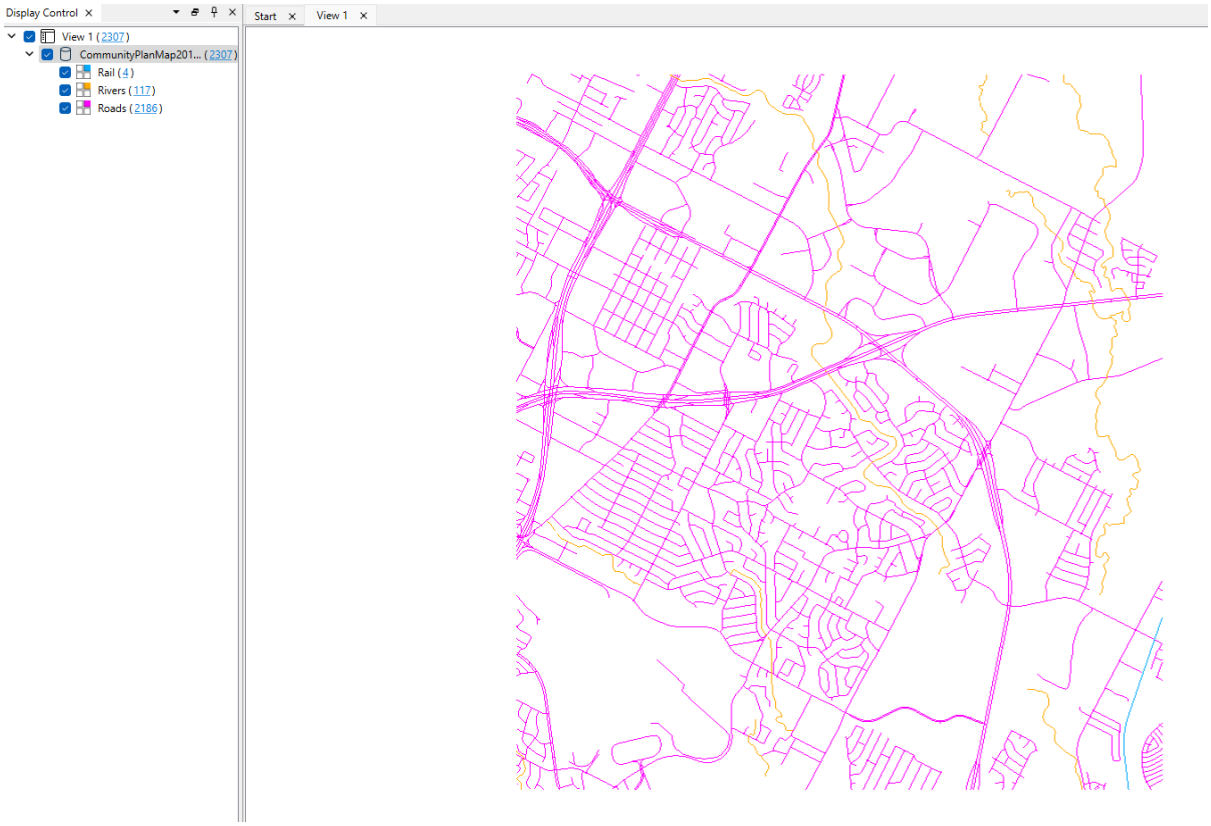


Click OK, then click OK again to read the data into the Data Inspector.

The Bentley DGN dataset contains line features that are styled white in colour. Therefore in order to see the features within the map view you'll need to either change the Background colour:



Or change the map layer styles within the Display Control:



There are 3 Feature Types in this dataset; Rail (with 4 Features), Rivers (117 Features), Roads (2186 Features).

1.3.2 Launch FME Workbench and Create new Workspace from Blank

Launch the FME Workbench, if it isn't open already.

Within the Get Started section of the Workbench, click on Blank Workspace.

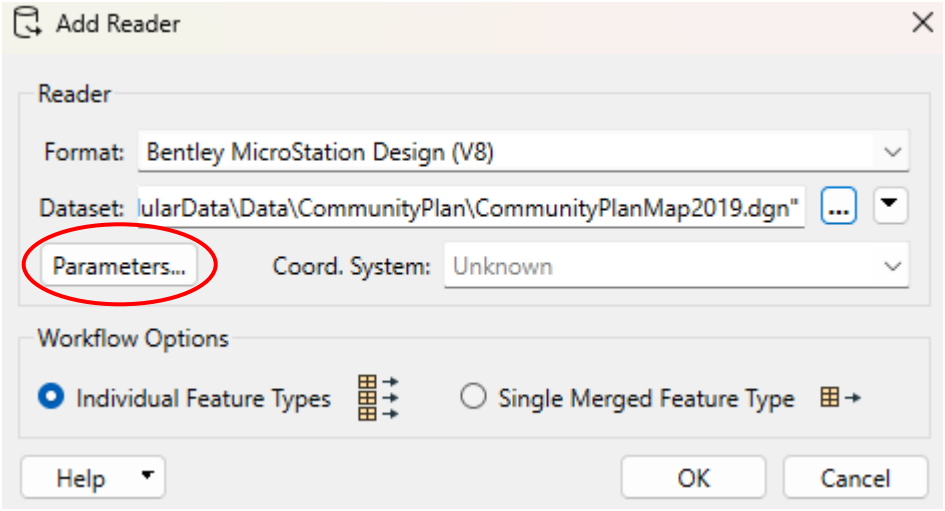
1.3.3 Add Reader

Use either the Reader button, or use Readers > Add Reader… from the menu bar.

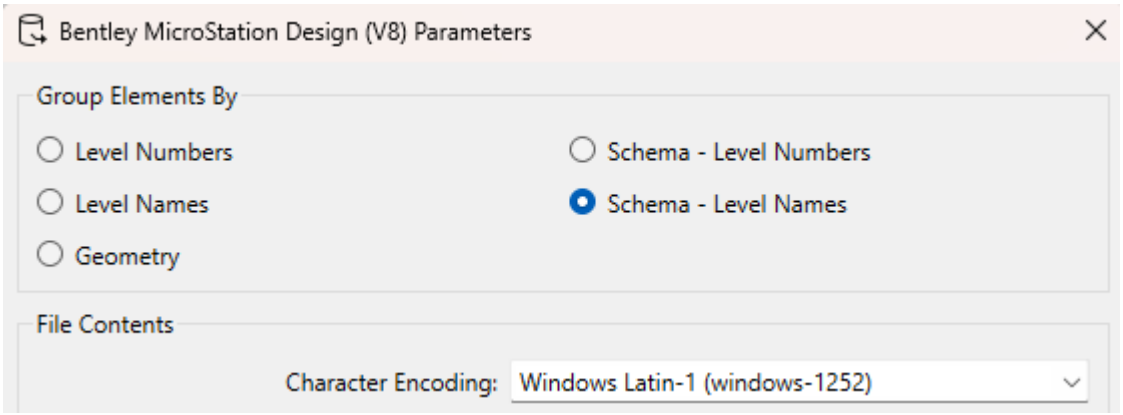
The Add Reader dialog will open, in which define the Format and Dataset settings as follows:

Reader Format	Bentley MicroStation Design (V8)
Reader Dataset	C:\FMEModularData\Data\CommunityPlan\ CommunityPlanMap2019.dgn

Also, click the **Parameters…** button

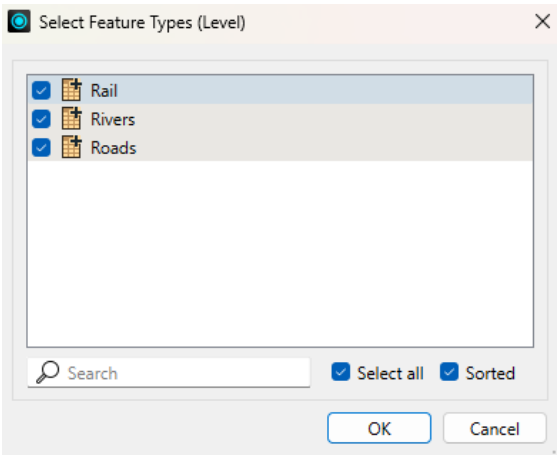


In the Group Elements By section, choose Schema – Level Names.

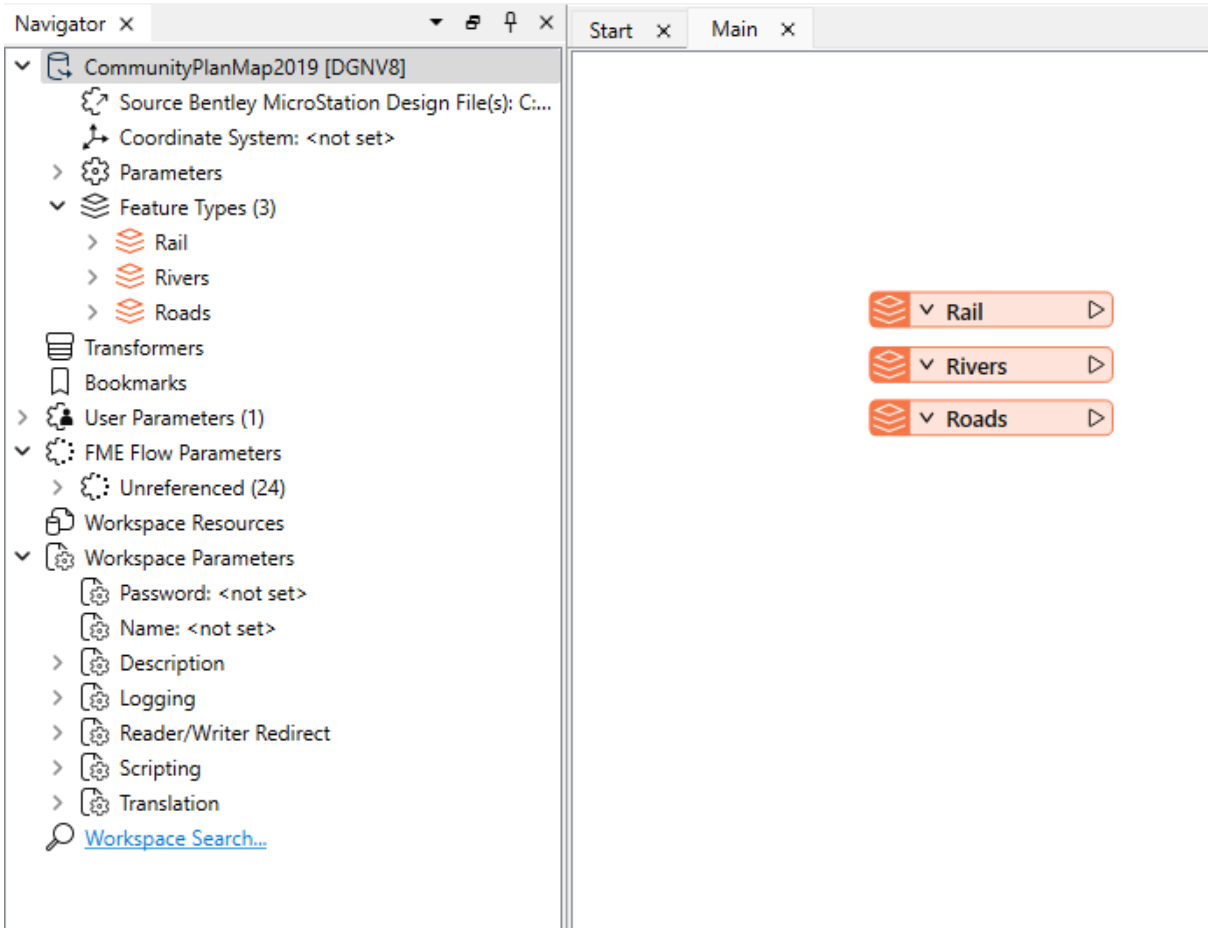


Click OK, then click OK again.

The Select Feature Types dialog will display, enabling us to choose which Feature Types (layers within the DGN) we'd like to include/read in.
Select all of them.



The Bentley DGN Reader will be added to the Navigator panel, and three Feature Types belonging to the Reader will appear on the canvas.

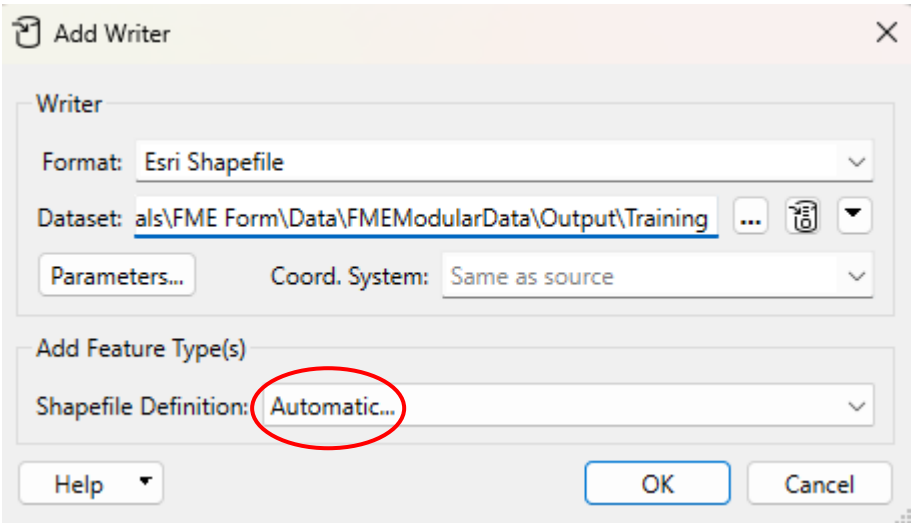


1.3.4 Add a Writer

Use either the Writer button, or use Writers > Add Writer… from the menu bar

The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	Esri Shapefile
Writer Dataset	C:\FMEModularData\Output\Training

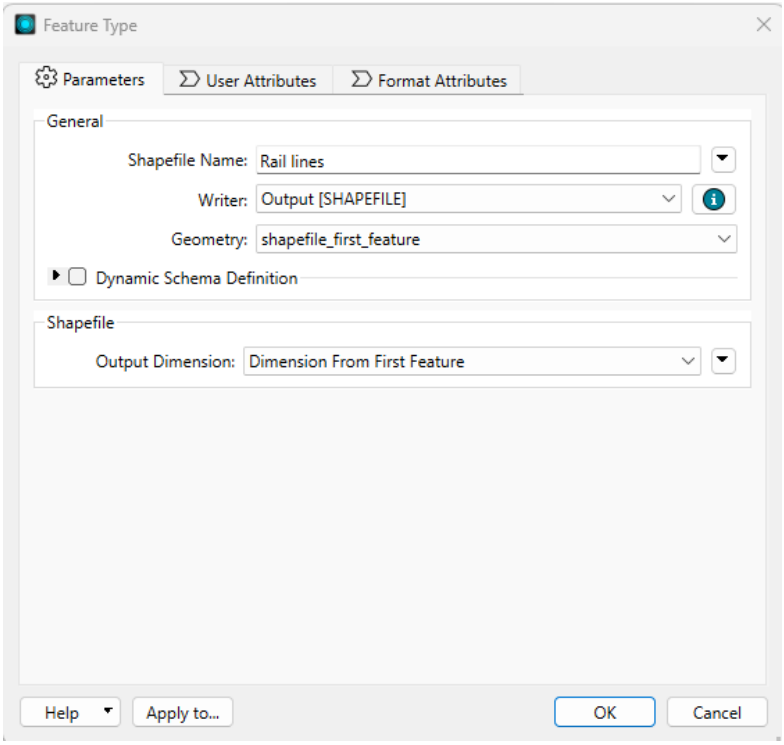


Also set the Table Definition: to Automatic…
- again, we'll explain this in a later module.

Notice that when we specify the Shapefile Dataset: destination we are selecting a folder, not setting the name of a .shp file!

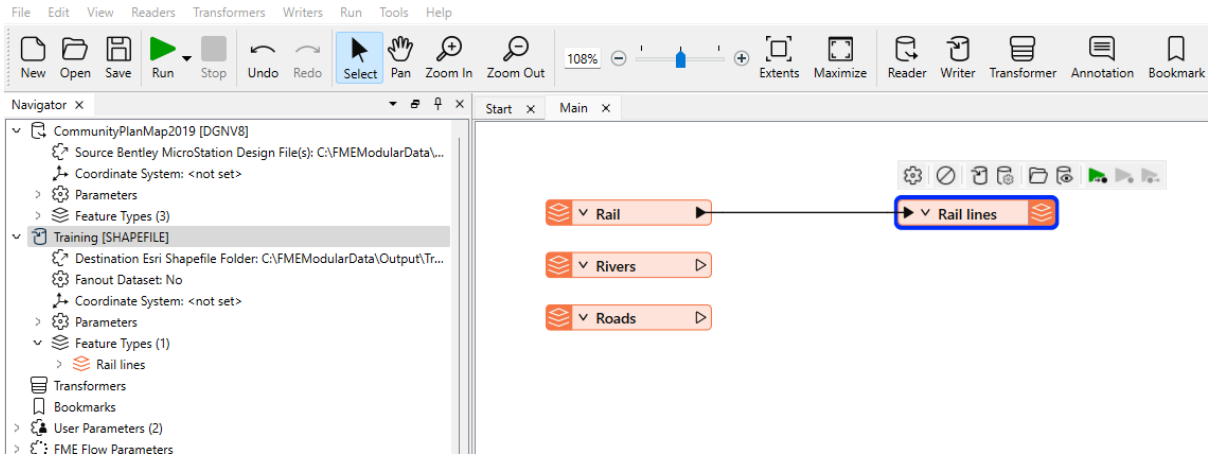
Click OK. A Feature Type dialog will open.

We need to set the name of our first Shapefile dataset. Name it *Rail lines*



Then click OK.

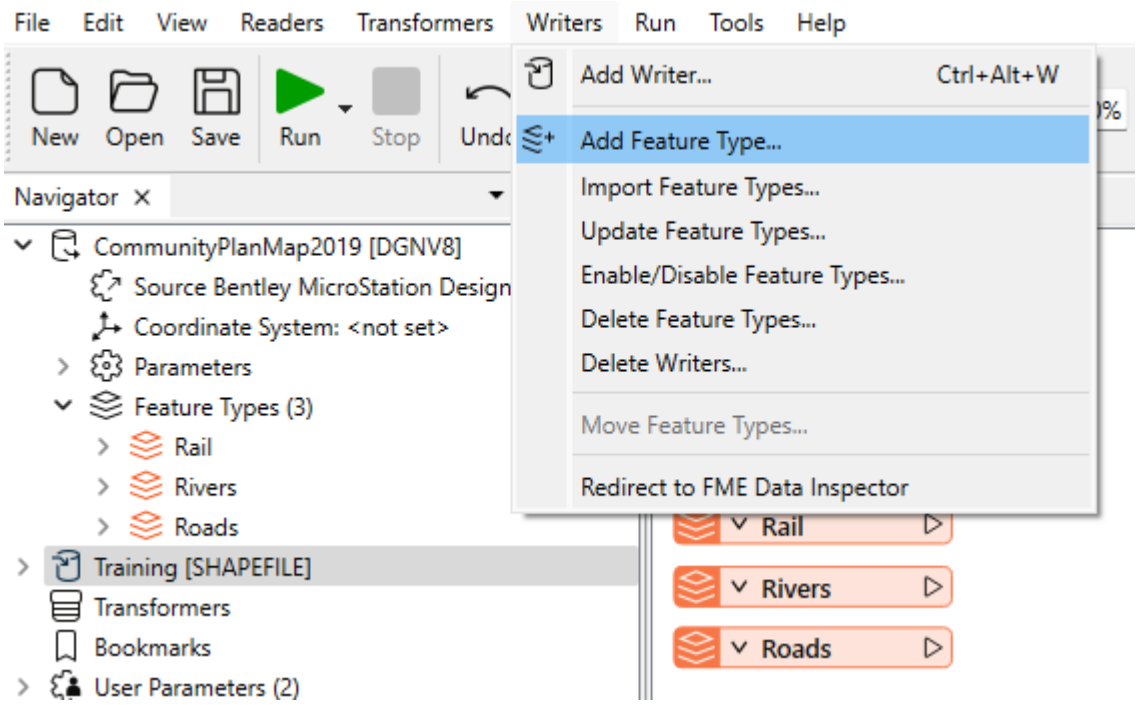
The ESRI Shapefile Writer will be added to the Navigator panel, and its single Feature Type (which we named Rail lines) is added to the canvas:



We want to create separate ESRI Shapefile datasets for Rail, Rivers and Roads. So, we will need two more Feature Types adding to our ESRI Shapefile Writer.

1.3.5 Add Feature Types to a Writer

Use Writers > Add Feature Type… from the menu bar

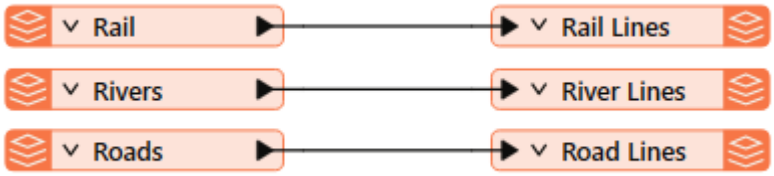


A Feature Type dialog will open.

We need to set the name of our second Shapefile dataset. Name it *River lines.* Then click OK.
Repeat this again to add a third Feature Type to the Writer, naming it *Road lines*.

1.3.6 Connect Reader Feature Types to the Writer Feature Types

Connect the Reader Feature Types to the appropriate Writer Feature Types with connection lines.



1.3.7 Save and Run Workspace

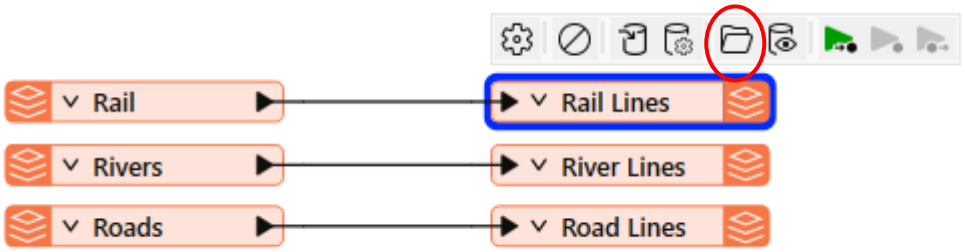
Save the workspace, then Run the workspace by clicking the Run button on the toolbar, or by using Run > Run Workspace on the menu bar.

The Translation Parameters dialog may display. Click Run to proceed.

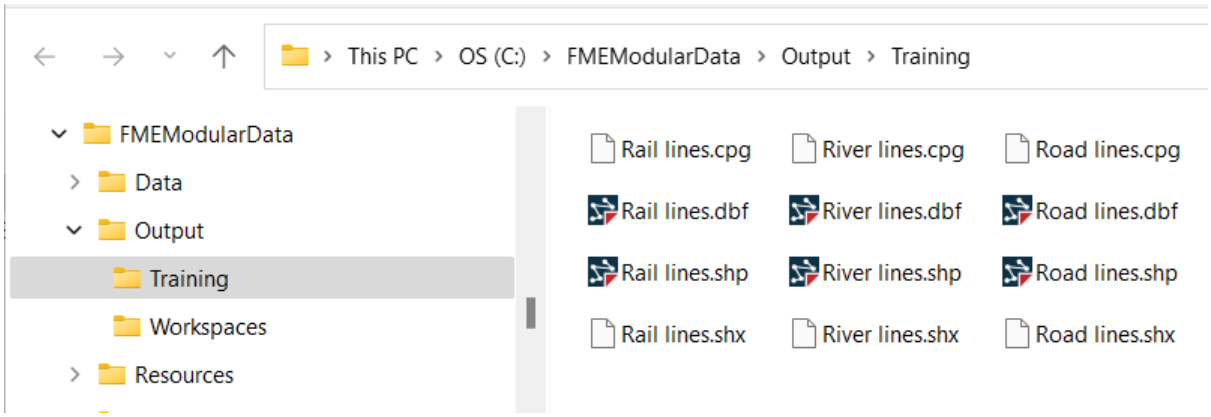
The Translation Log should confirm that the translation was successful.

1.3.8 Inspect output ESRI Shapefiles

Use a shortcut ‘Open Containing Folder’ to check the results in File Explorer window.



From a single DGN file, FME created 3 separate Shapefiles; Rail lines.shp, River lines.shp and Road lines.shp



There is the difference between FME hierarchy levels in the two different formats that we worked with in this example:

- the Bentley DGN format is **file-based** – the Reader referenced the .dgn file, whilst the Feature Types were the layers of data within the dataset.

- the Shapefile format is **folder-based** – the Writer referenced a destination **folder** (not .shp files), whilst its Feature Types referenced the actual .shp files created.

Congratulations

By Completing this exercise you have learned how to:

- Create a workspace to convert CAD data to a GIS format
- Create a Workspace with multiple Feature Types

2 Managing Attributes & Filtering Features

2.1 Managing Attributes

Demonstrates	Schema editing on the Writer Feature Type. Manage attribute mapping using Transformers.
Overall Goal	Update the attribute schema for grounds maintenance data. Calculate the average area of parks in the city as a new attribute. Output results to multiple formats.
Data	Parks (MapInfo TAB)
Start Workspace	none
End Workspace	C:\FMEModularData\Workspaces\Complete\2.01-Attributes-ManagingAttributes-Complete.fmw

You are working as a technical analyst in the GIS department of your local city.
The team responsible for maintaining parks and other grassed areas needs to know the area and facilities of each park in order to plan their budget for the upcoming year.
You will need to remove unnecessary attributes, rename existing attributes and create new ones – including area calculations.
The grounds maintenance team want the data in both spatial and CSV format.

2.1.1 Launch FME Workbench and Create new Workspace from Blank

Launch the FME Workbench, if it isn't open already.

Within the Get Started section of the Workbench, click on Blank Workspace.

2.1.2 Add Reader

Use either the Reader button, or use Readers > Add Reader… from the menu bar.

The Add Reader dialog will open, in which define the Format and Dataset settings as follows:

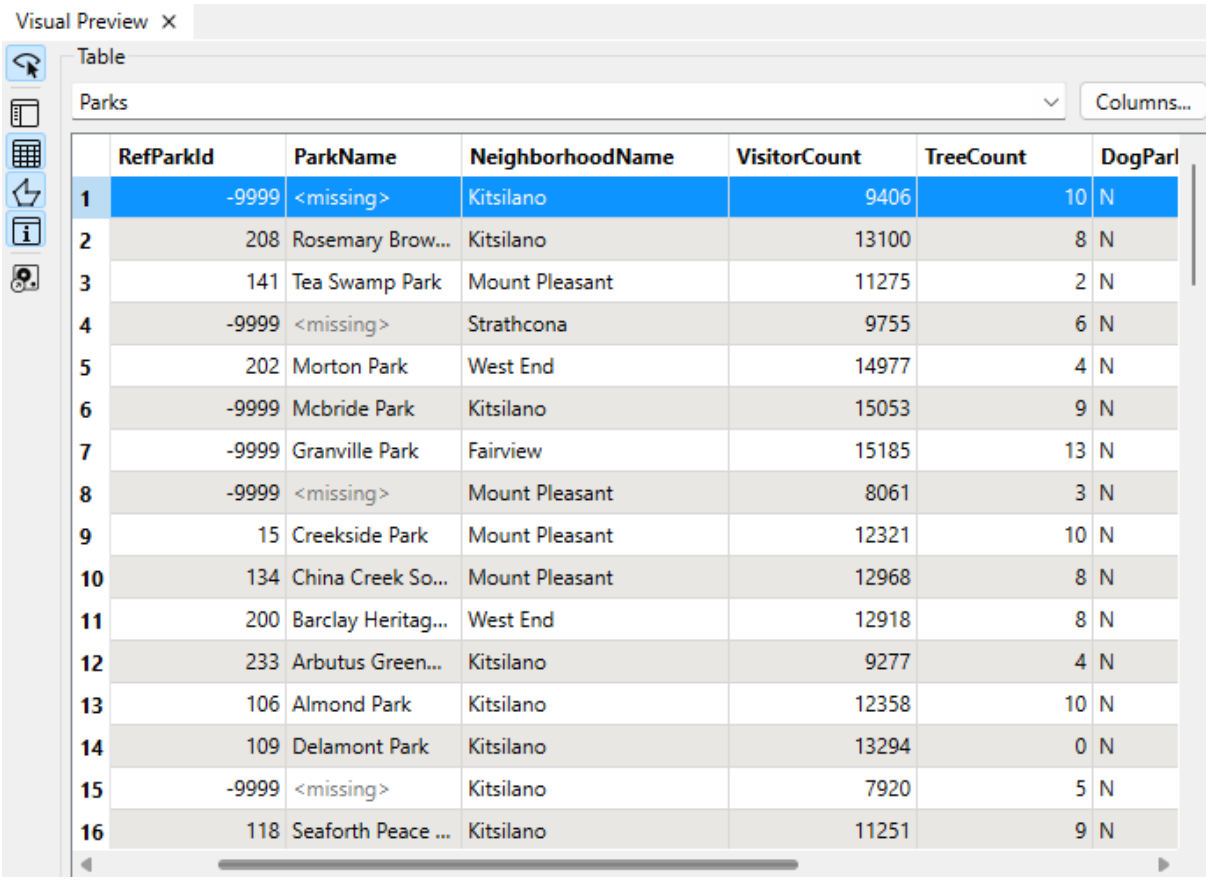
Reader Format	Precisely MapInfo TAB (MAPINFO)
Reader Dataset	C:\FMEModularData\Data\Parks\Parks.tab

Into the Navigator panel a MapInfo Reader will be added and on the canvas there will be a single reader feature type for Parks.

2.1.3 Examine the Parks Data

Before we start designing our workflow, first examine the parks data. Select the Parks reader feature type on the canvas (so it becomes highlighted in blue) then use the View Source Data button.

Within the Visual Preview panel view the data within a Table view. This will display the Parks attributes and attribute values.



	RefParkId	ParkName	NeighborhoodName	VisitorCount	TreeCount	DogParl
1	-9999	<missing>	Kitsilano	9406	10	N
2	208	Rosemary Brow...	Kitsilano	13100	8	N
3	141	Tea Swamp Park	Mount Pleasant	11275	2	N
4	-9999	<missing>	Strathcona	9755	6	N
5	202	Morton Park	West End	14977	4	N
6	-9999	Mcbride Park	Kitsilano	15053	9	N
7	-9999	Granville Park	Fairview	15185	13	N
8	-9999	<missing>	Mount Pleasant	8061	3	N
9	15	Creekside Park	Mount Pleasant	12321	10	N
10	134	China Creek So...	Mount Pleasant	12968	8	N
11	200	Barclay Heritag...	West End	12918	8	N
12	233	Arbutus Green...	Kitsilano	9277	4	N
13	106	Almond Park	Kitsilano	12358	10	N
14	109	Delamont Park	Kitsilano	13294	0	N
15	-9999	<missing>	Kitsilano	7920	5	N
16	118	Seaforth Peace ...	Kitsilano	11251	9	N

In this exercise we are interested in setting up the required attributes on our writers; the attribute names as well as assigning the data the correct data type.

We need to output the results to both CSV and a spatial format. We'll start with our CSV Writer.

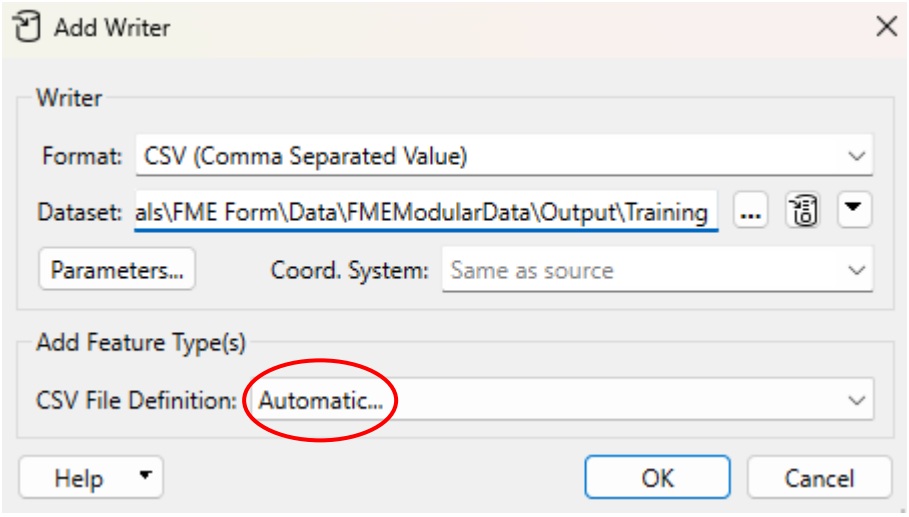
2.1.4 Add a Writer – CSV format

Use either the Writer button, or use Writers > Add Writer… from the menu bar

The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	CSV (Comma Separated Value)
Writer Dataset	C:\FMEModularData\Output\Training

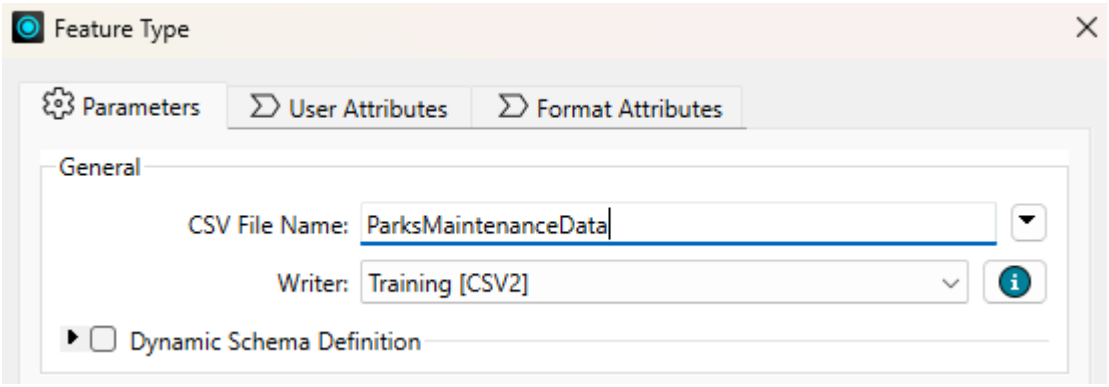
Also set the CSV File Definition: to **Automatic…** - *we'll explain this in a moment.*



Click OK. A Feature Type dialog will open.

2.1.5 Rename Feature Type

Click in the field labelled CSV File Name: and change the name to *ParksMaintenanceData*:



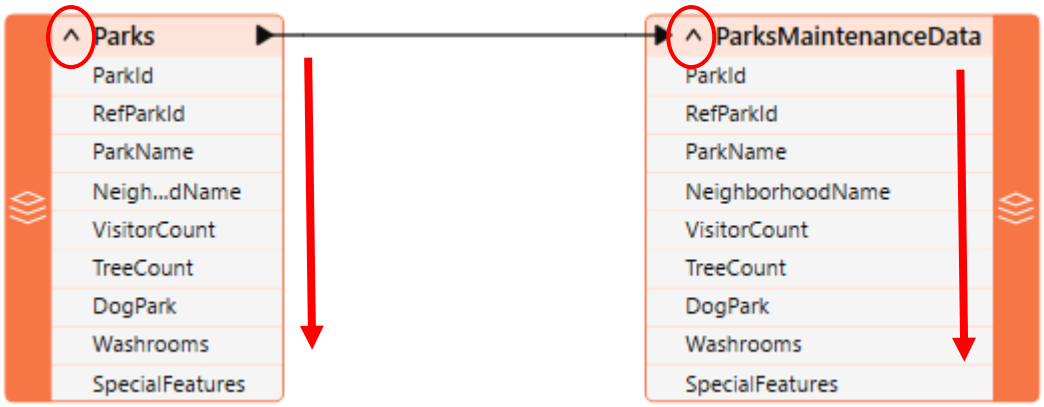
Click OK

2.1.6 Connect the Feature Types

Drag a connector line between the Parks (reader) feature type and the ParksMaintenceData (writer) feature type:

2.1.7 Examine the Attribute lists

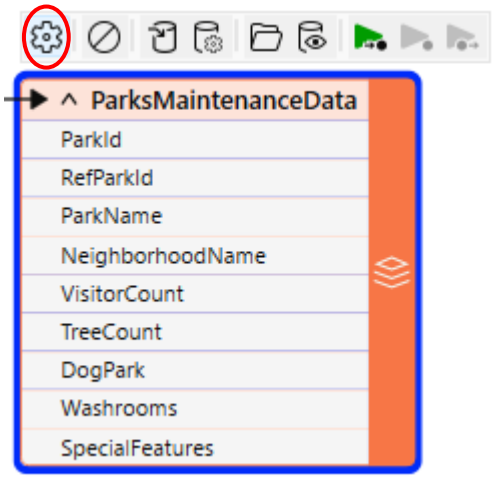
Expose the attribute lists of both feature types by clicking on the arrows:

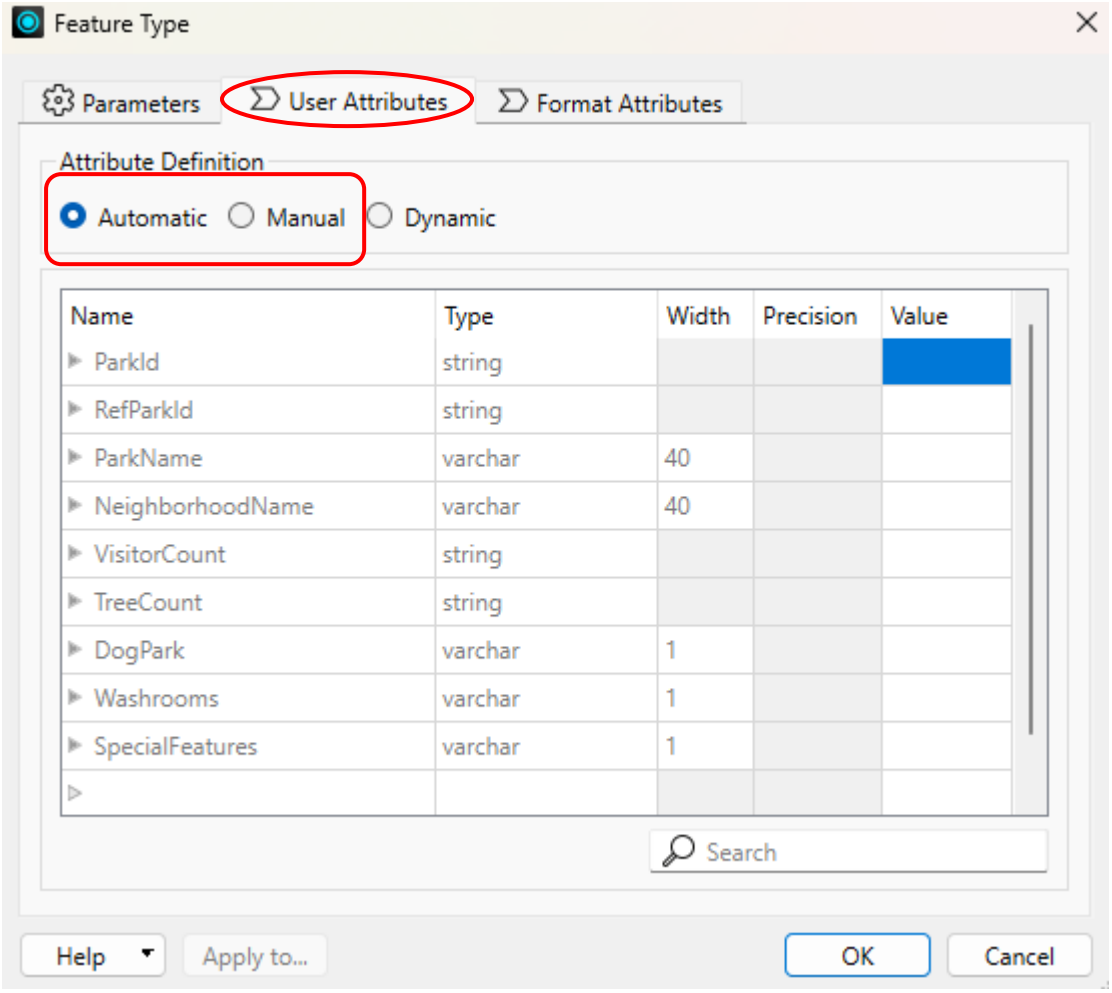


The ParksMaintenceData (writer) feature type is currently configured to automatically include any attributes passed into it from the workflow. This is because at the time of adding the CSV Writer, *Automatic* mode was chosen for attribute definition (instead of *Manual…*, *Copy from Reader…*, *etc*).
This means that as we configure our workspace and alter the attributes being passed into the ParksMaintenceData (writer) feature type, the attributes on the ParksMaintenceData (writer) feature type will automatically update.

2.1.8 Attribute Definition on Writer Feature Type

We can check (and modify) the mode used for the Attribute Definition on a writer feature type; click on the writer feature type parameters cog. The Attribute Definition mode can be found on the User Attributes tab:





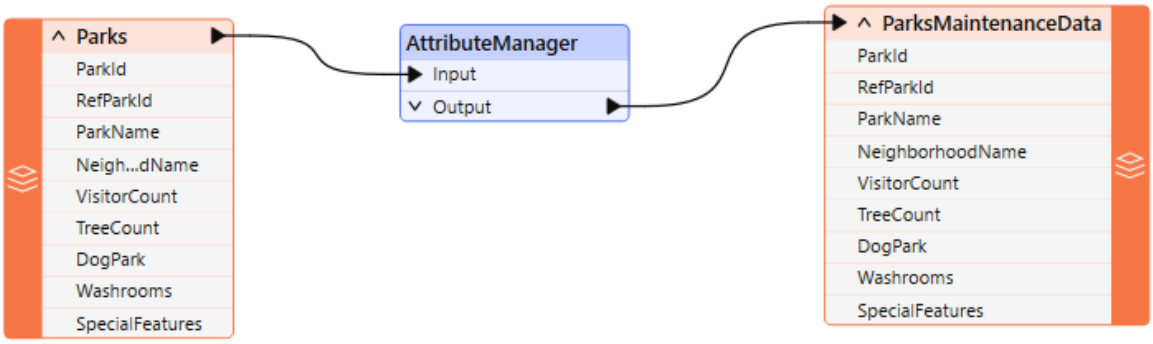
For now, we will leave the Attribute Definition mode set to Automatic – to save us time and effort whilst we manage our attributes within the workflow using transformers. But we'll return later and switch to Manual attribute definition mode when we are ready to finalise our Data Types.

2.1.9 Manage Attributes within workflow using Transformer

The end user of the data has requested the schema be changed; this includes changing attribute names, removing unwanted attributes and creating new attributes. There are a number of attribute management transformers that we could use, but in this example we'll use the AttributeManager.

Click on the canvas. Nothing will appear, this just makes the canvas active. Then start typing the name of the required transformer; "AttributeManager".
When you see the AttributeManager transformer appear in the list, double-click on it to place it onto the canvas.

You now need to connect AttributeManager transformer between the two feature types.

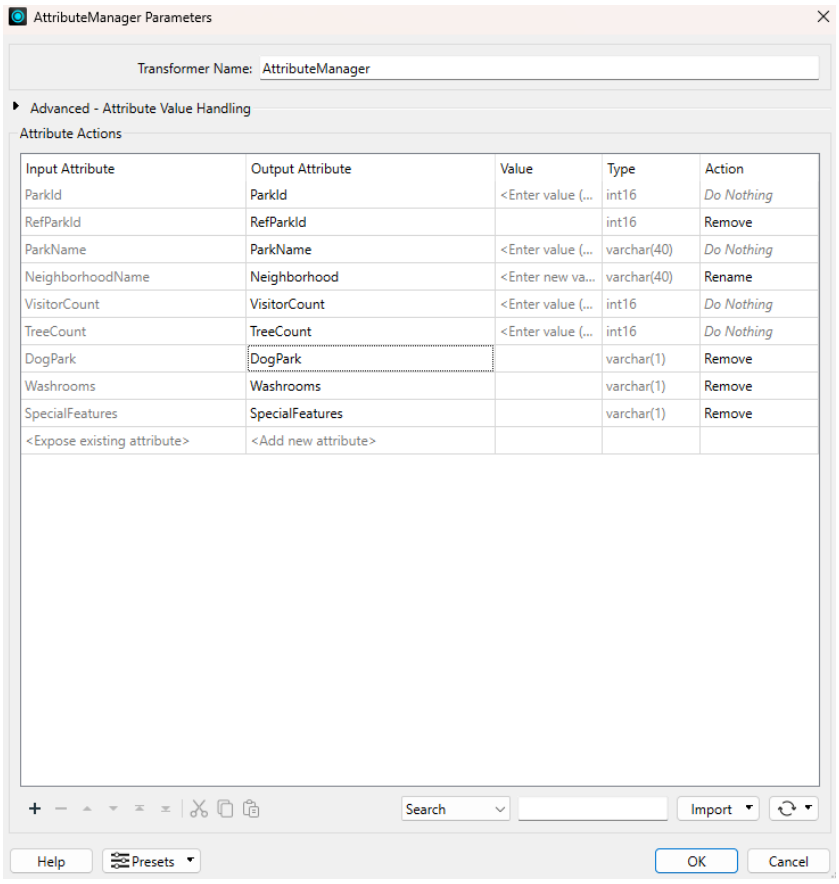


2.1.10 Configure AttributeManager parameters

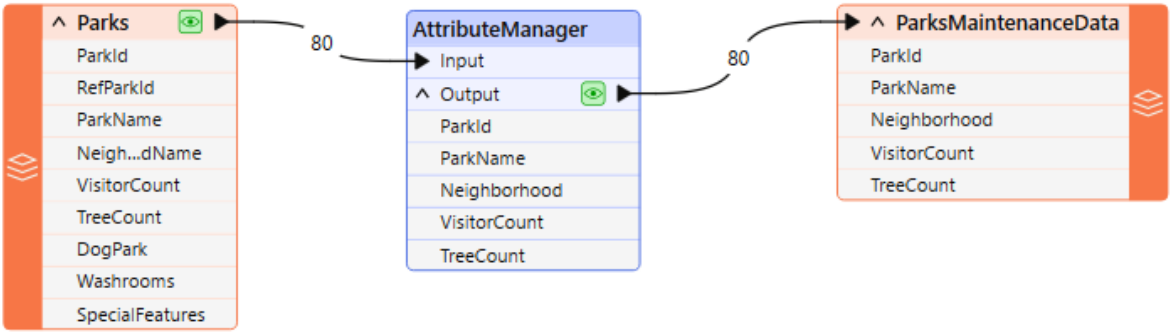
Click on the transformer's parameters cog (or double-click on the transformer titlebar). Within the AttributeManager make the following changes:

Remove Attribute	RefParkID
Remove Attribute	DogPark
Remove Attribute	Washrooms
Remove Attribute	SpecialFeatures
Rename Attribute	change Output Attribute value from: NeighborhoodName to: Neighborhood

The AttributeManager parameters should now look similar to this:



When you return to the canvas, notice that the attributes on the ParksMaintenceData (writer) feature type have automatically updated, reflecting the changes made by the AttributeManager transformer.



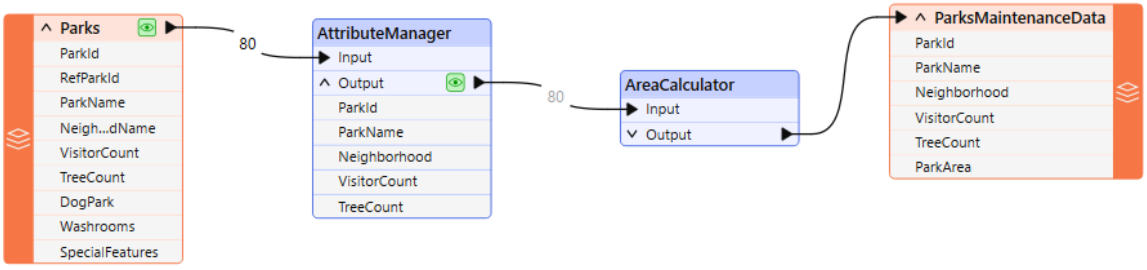
Next we need to identify the size of each park, then calculate the average park size for the city.

2.1.11 Add an AreaCalculator Transformer

This transformer will analyse the park polygons to measure the size (area) of each park.

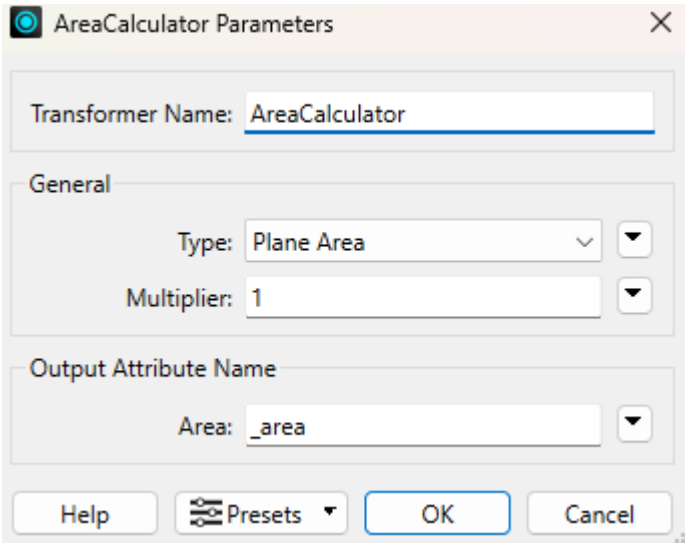
Click on the canvas and start typing “AreaCalculator”. When you see the AreaCalculator transformer appear in the list, double-click on it to place it onto the canvas.

Next, connect the AreaCalculator transformer between the AttributeManager and the ParksMaintenceData feature type.

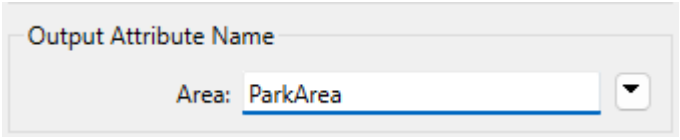


2.1.12 Check the AreaCalculator Parameters

Inspect the parameters of the AreaCalculator transformer:
The default settings cause the calculated value to be placed into an attribute called *_area*.



However, the *ParksMaintenanceData* schema requires an attribute called *ParkArea*, so change this parameter to createthe correct attribute:



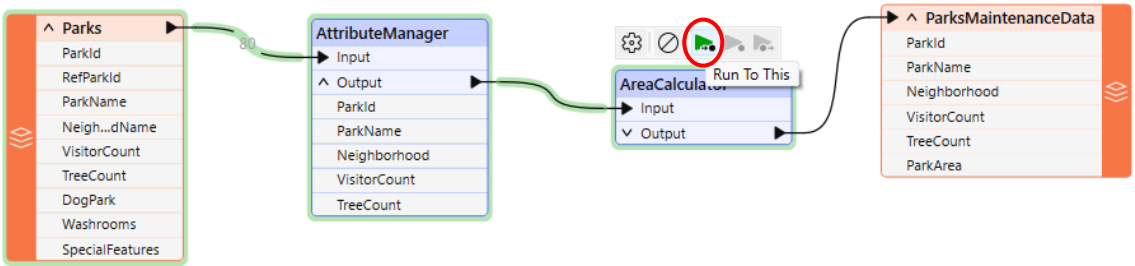
Let's save and run the workspace to this point and examine the features coining out of the AreaCalculator.

2.1.13 Save the Workspace

Click on the Save button, or use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save

2.1.14 Run To This

Select the AreaCalculator transformer, then click on the button *Run To This*



The workspace will run as far as the AreaCalculator transformer. So we can now examine the features to this point of the workflow - instead of running the entire workspace and writing output data (which we'd need to later overwrite).

2.1.15 Inspect the AreaCalculator Feature Cache

Click on the AreaCalculator transformer feature cache to examine the features at this point in the workflow.

Within the Visual Preview window use the Toggle Table View button to display the table view, where we can then examine the attributes.

Table

AreaCalculator: Output Columns...

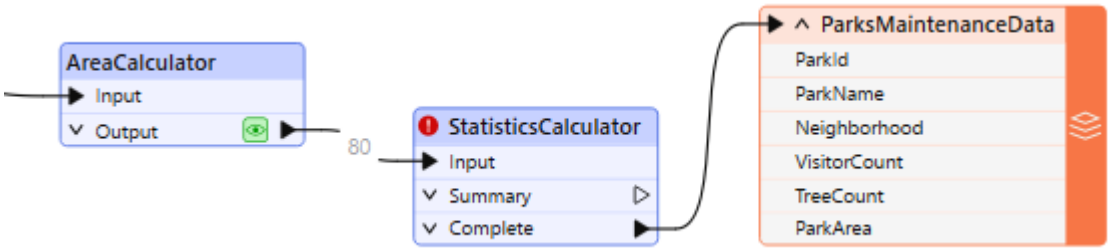
	ParkId	ParkName	Neighborhood	VisitorCount	TreeCount	ParkArea
1	1	<missing>	Kitsilano	9406	10	448.12468066
2	2	Rosemary Brow...	Kitsilano	13100	8	1035.0770808
3	3	Tea Swamp Park	Mount Pleasant	11275	2	2631.2639861
4	4	<missing>	Strathcona	9755	6	1984.8363571
5	5	Morton Park	West End	14977	4	2197.3181996
6	6	Mcbride Park	Kitsilano	15053	9	17125.717083
7	7	Granville Park	Fairview	15185	13	19655.705362

The AreaCalculator transformer has added a new attribute called ParkArea, which is populated with the polygon size for each park.

2.1.16 Add a StatisticsCalculator Transformer

Using the same method as before for finding and adding a transformer, place a StatisticsCalculator transformer between the AreaCalculator output port and the *ParksMaintenanceData* feature type.

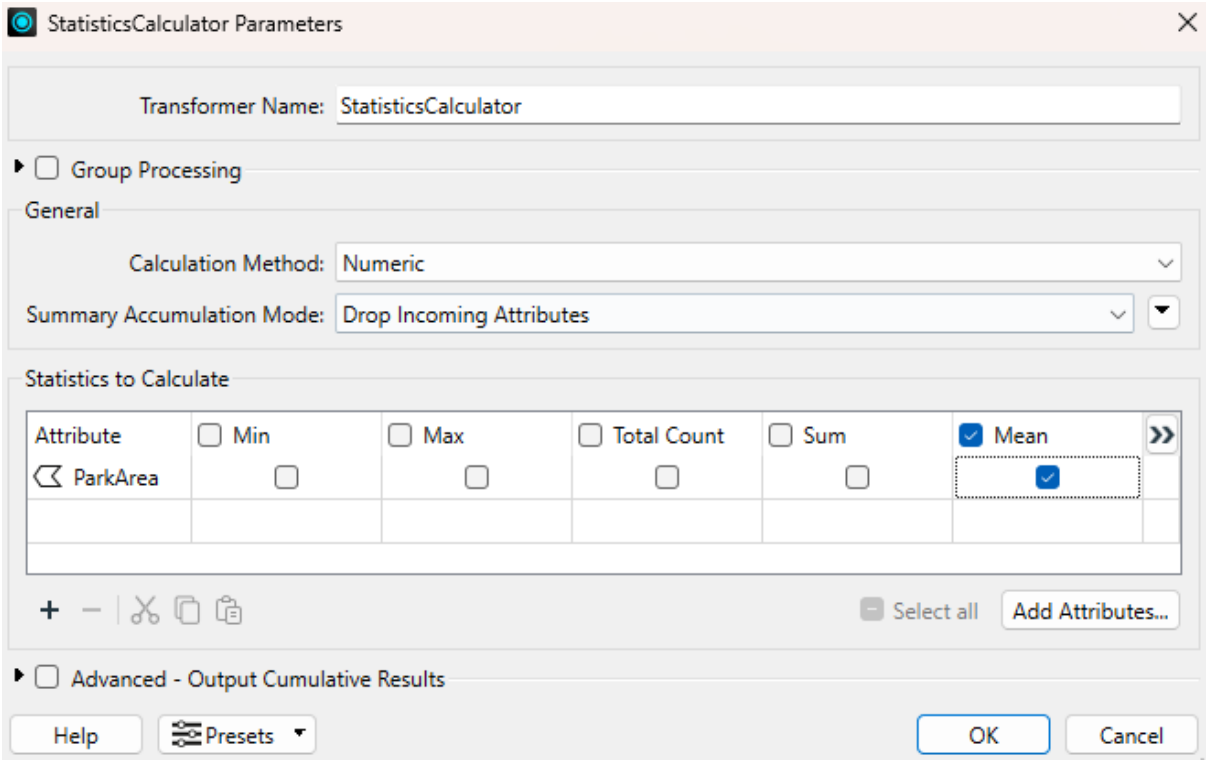
The StatisticsCalculator transformer has two output ports, we want to take the results from the Complete port. But by default the output connection will be from the Summary port, so **move the connection to the *Complete*** port:



2.1.17 Configure the StatisticsCalculator Parameters

A red ! icon indicates the StatisticsCalculator has parameters that need to be defined. Inspect the parameters of the StatisticsCalculator transformer:

The *Statistics to Calculate section* enables a user to analyse attributes of choice and return statistics about them. We need to analyse the ParkArea attribute and return the Mean (giving us the average).
So select ParkArea as the Attribute, then tick to calculate the Mean:



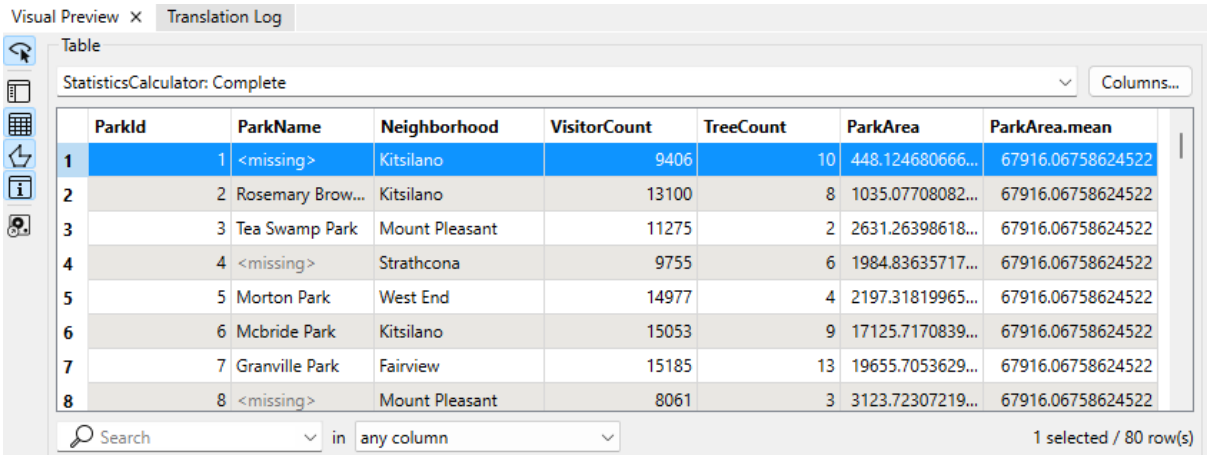
2.1.18 Run To This

Select the StatisticsCalculator transformer, then click on the button *Run To This.*
The workspace will run as far as the StatisticsCalculator transformer. So we can now examine the features to this point of the workflow

2.1.19 Inspect the AreaCalculator Feature Cache

Click on the Complete port feature cache of the StatisticsCalculator transformer to examine the features at this point.

Within the Visual Preview window use the Toggle Table View button to display the table view, where we can then examine the attributes.



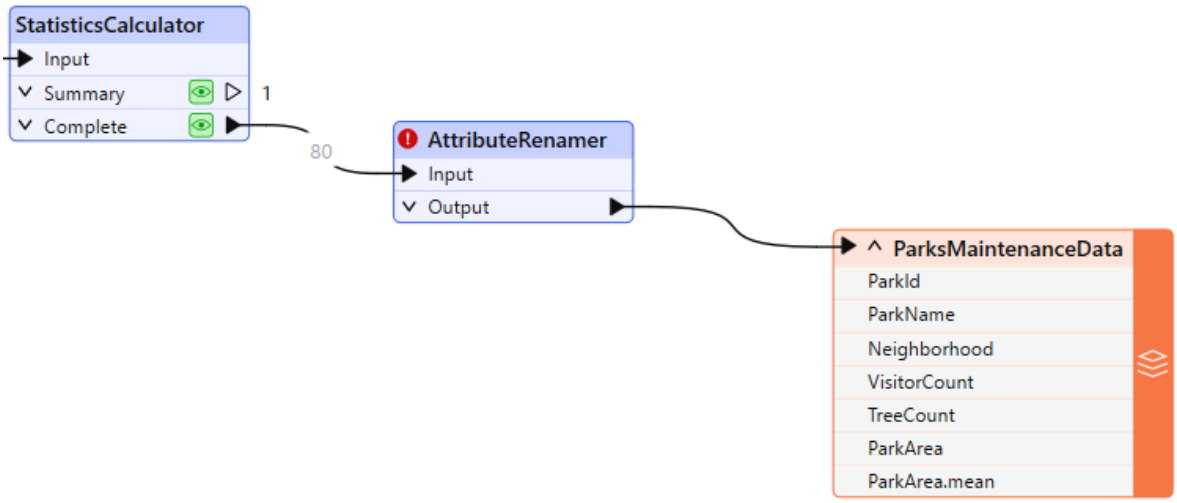
	ParkId	ParkName	Neighborhood	VisitorCount	TreeCount	ParkArea	ParkArea.mean
1	1	<missing>	Kitsilano	9406	10	448.124680666...	67916.06758624522
2	2	Rosemary Brow...	Kitsilano	13100	8	1035.07708082...	67916.06758624522
3	3	Tea Swamp Park	Mount Pleasant	11275	2	2631.26398618...	67916.06758624522
4	4	<missing>	Strathcona	9755	6	1984.83635717...	67916.06758624522
5	5	Morton Park	West End	14977	4	2197.31819965...	67916.06758624522
6	6	Mcbride Park	Kitsilano	15053	9	17125.7170839...	67916.06758624522
7	7	Granville Park	Fairview	15185	13	19655.7053629...	67916.06758624522
8	8	<missing>	Mount Pleasant	8061	3	3123.72307219...	67916.06758624522

The average (mean) of the ParkArea values has been calculated and added to the data

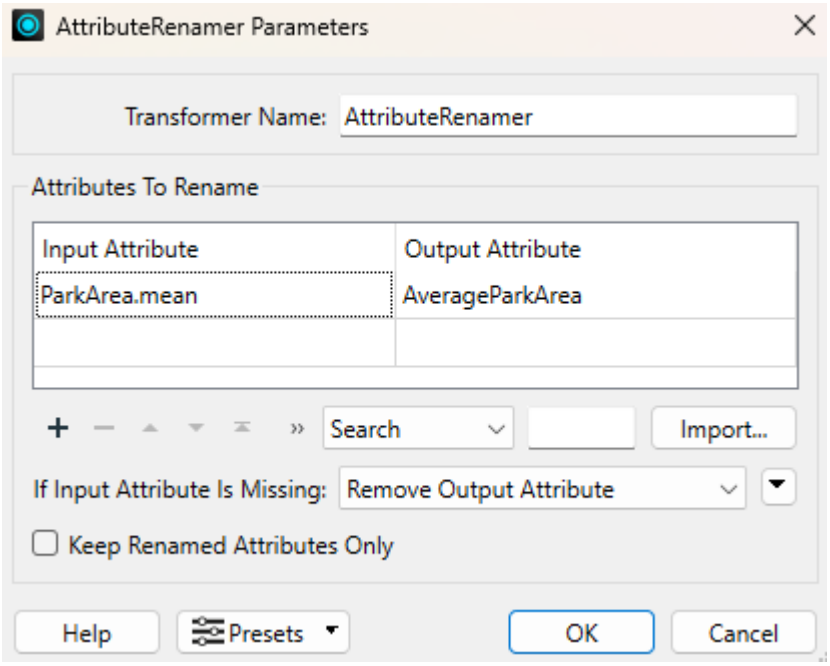
as a new attribute called ParkArea.mean. We need to rename this attribute to match the requested schema.

2.1.20 Rename Attribute

We could use an AttributeManager, but an alternative transformer is the AttributeRenamer. Find and add a AttributeRenamer transformer to the canvas.
Then connect the transformer into the workflow between the StatisticsCalculator transformer (from the Complete port) and the *ParksMaintenanceData* feature type.



Then set the transformer parameters to rename ParkArea.mean to AverageParkArea

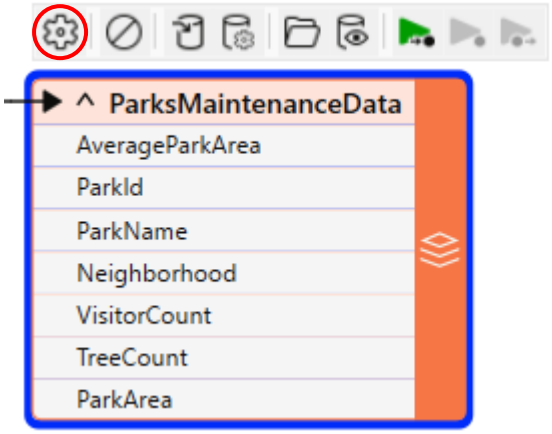


We now have the final schema editing tasks to do on the *ParksMaintenanceData writer* feature type.

2.1.21 Edit Attributes on Writer Schema

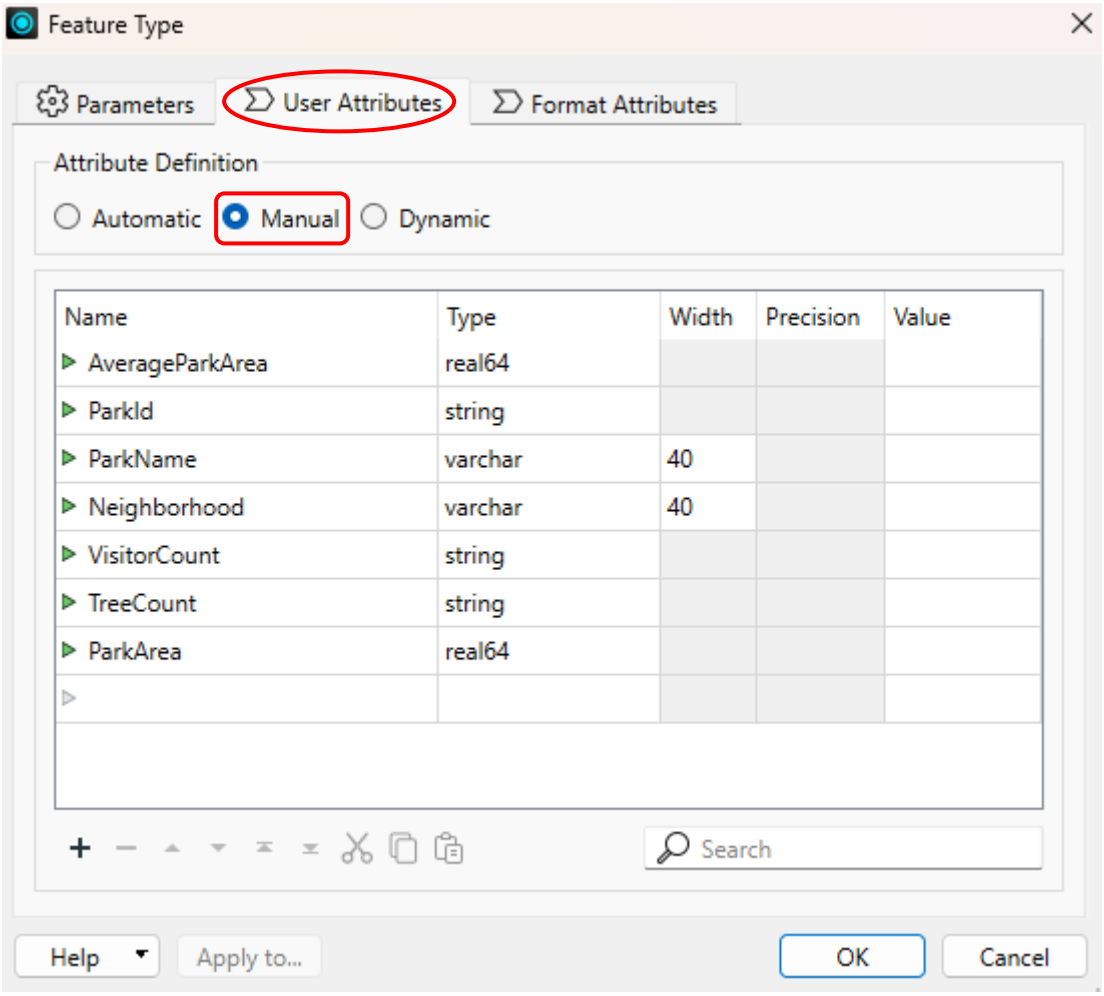
We've used transformers within the workflow to manage and create attributes, but now we need to make some final edits on the Writer Schema. Click on the

ParksMaintenanceData writer feature type parameters cog.



Currently the Attribute Definition mode is set to Automatic. We are ready to finalise our Data Types for each attribute, however we cannot make any manual edits whilst in Automatic definition mode.

Switch to Manual attribute definition mode:



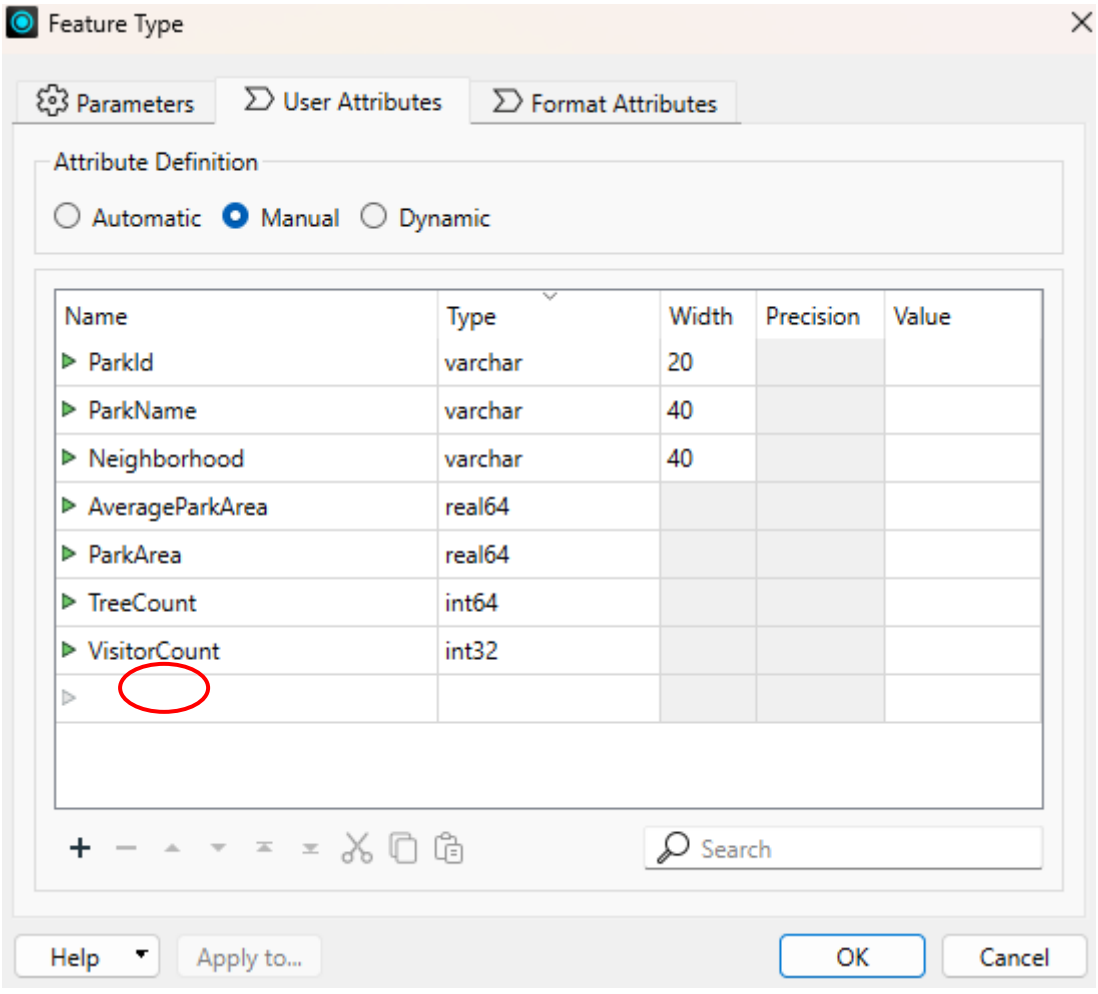
With the mode set to Manual, we can now access and edit the attributes on the feature type.

For CSV format, these are the required data types for the following attributes:

ParkId	varchar width 20	*the data values are text strings, which do not exceed 20 characters in size*
VisitorCount	int32	*the data values are whole number, no decimals (32-bits)*
TreeCount	int16	*the data values are whole number, no decimals (16-bits)*

Whilst we're here we'll make an adjustment to the order of the attributes. (Attribute reordering can also be performed using an AttributeManager transformer).

Select the attribute called *AverageParkArea* then use the Move Down arrow to demote it to the bottom of the attribute list:



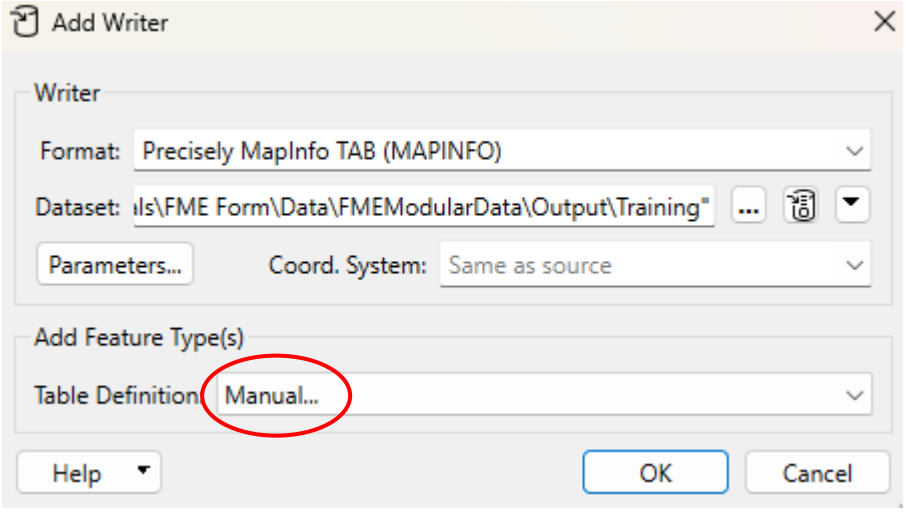
We've now completed our schema editing and all the required attributes have been configured and created. In addition to supplying the results data in CSV format, then end user has also requested the data in GIS format. Let's provide it in MapInfo TAB format.

2.1.22 Add another Writer

Use either the Writer button, or use Writers > Add Writer… from the menu bar

The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	Precisely MapInfo TAB (MAPINFO)
Writer Dataset	C:\FMEModularData\Output\Training



This time set the Table Definition: to *Manual…*

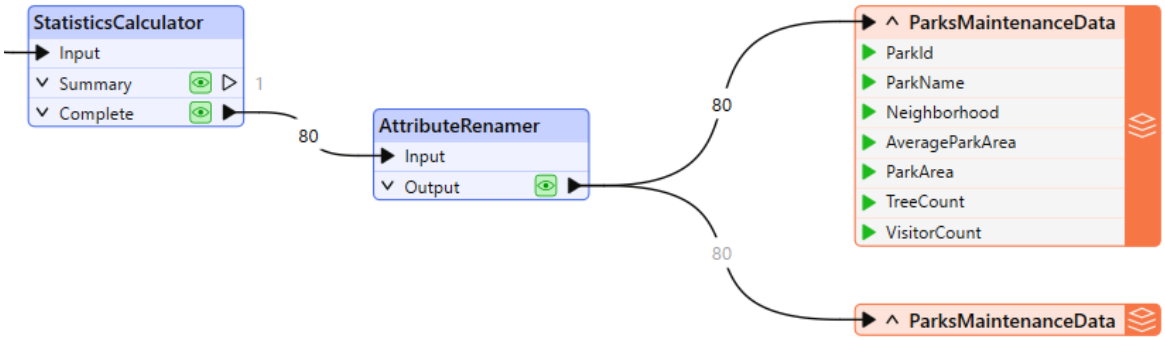
Within our workspace we already have the required attributes defined (including their data Types). So, it will be quicker and less effort for us to set the attribute definition mode of our new writer to manual, then simply copy the attribute definition from our existing CSV feature type.

Click OK. A Feature Type dialog will open.

Set the Table Name: to *ParksMaintenanceData.* Then click OK.

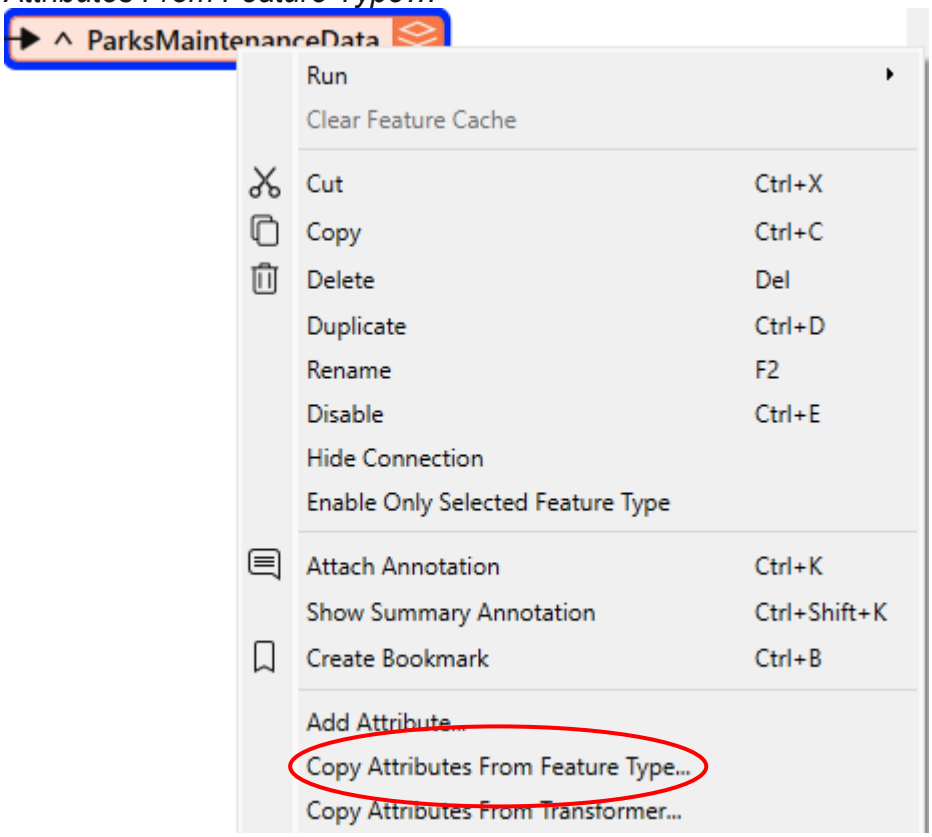
The new MapInfo writer will be added to the Navigator panel, and its feature type will be added to the canvas.

Drag a connector line between the AttributeRenamer and the new MapInfo writer ParksMaintenanceData feature type.

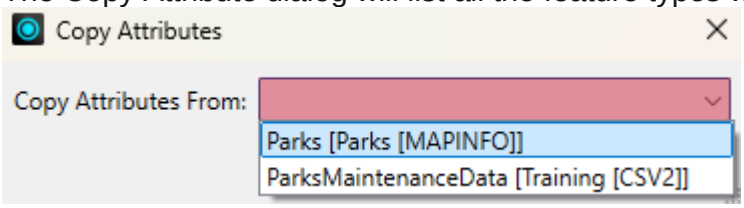


2.1.23 Copy Attributes From Feature Type

Right-click on the MapInfo writer ParksMaintenanceData feature type, then choose *Copy Attributes From Feature Type…*

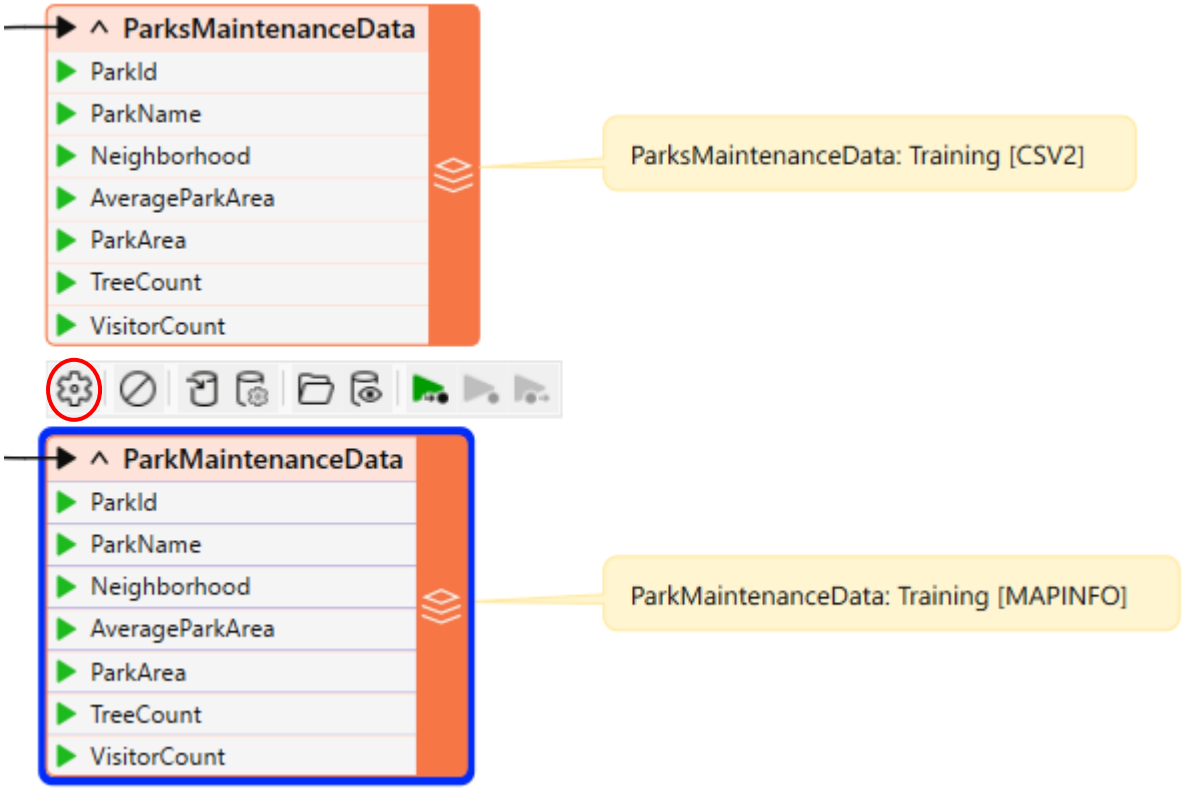


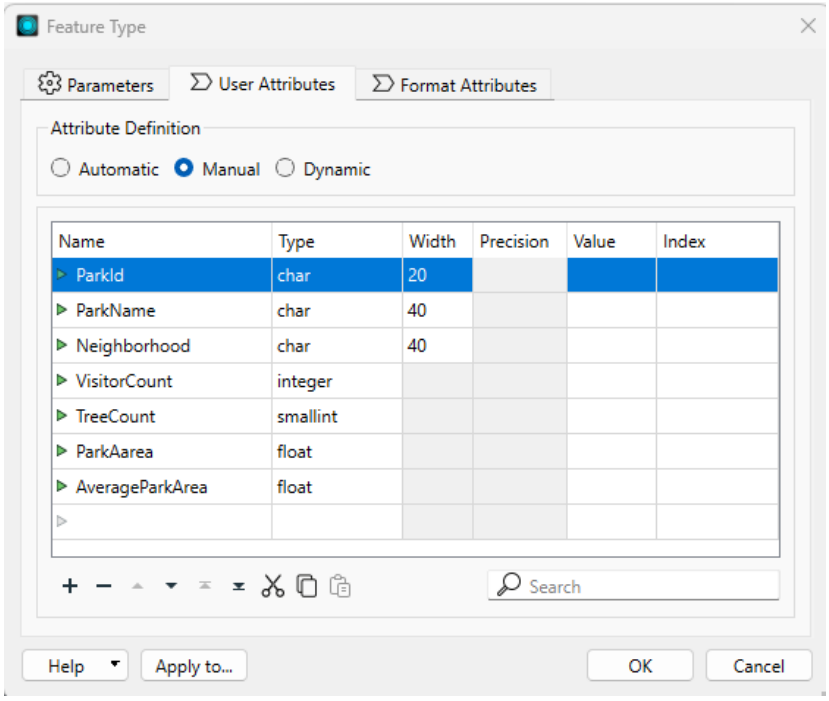
The Copy Attribute dialog will list all the feature types within the workspace.



Choose *ParksMaintenanceData [Training [CSV]]*

The attribute definition will be copied onto the MapInfo ParksMaintenanceData feature type:





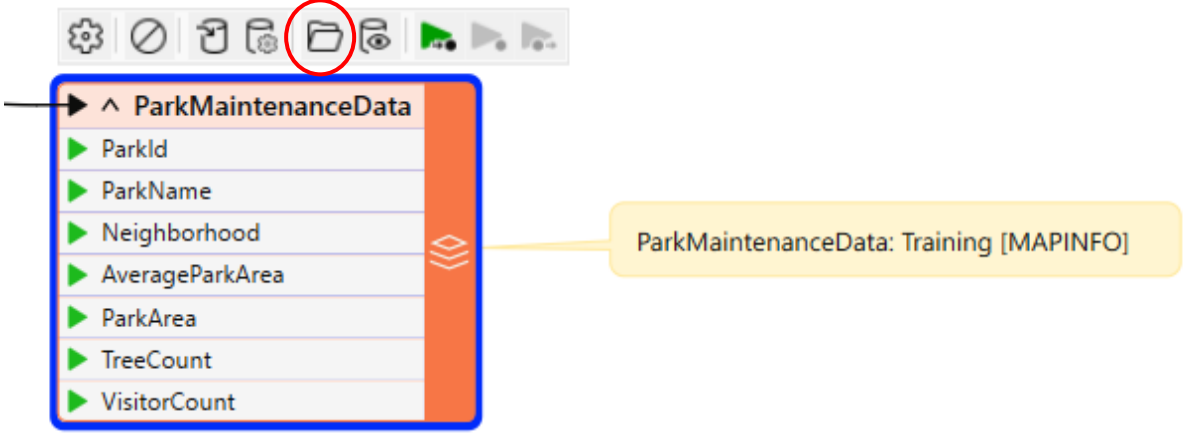
2.1.24 Resave the Workspace then ReRun Entire Workspace

Click on the Save button on the menu bar.

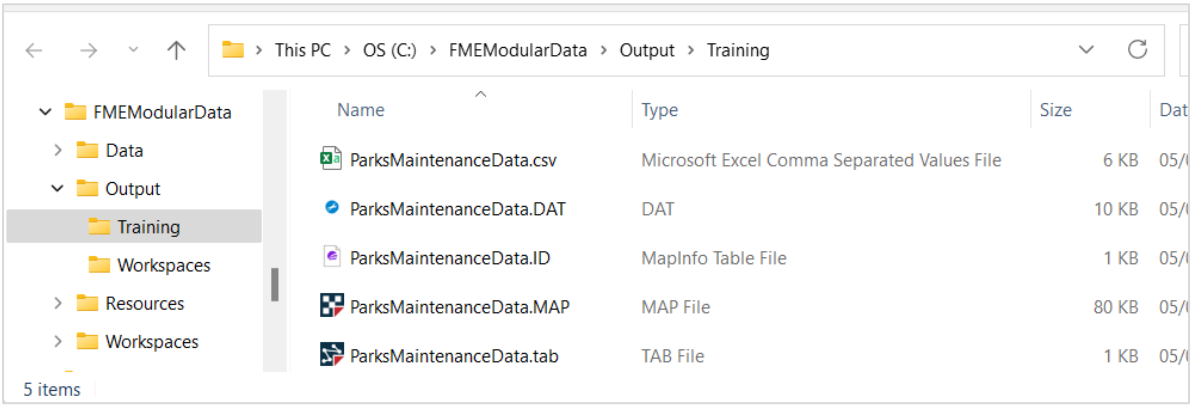
Then rerun the entire workspace using Run > Rerun Entire Workspace from the menu bar. The Translation Parameters dialog may display. Click Run to proceed.

2.1.25 Open Containing Folder

Select one of the writer feature types, then use the button Open Containing Folder



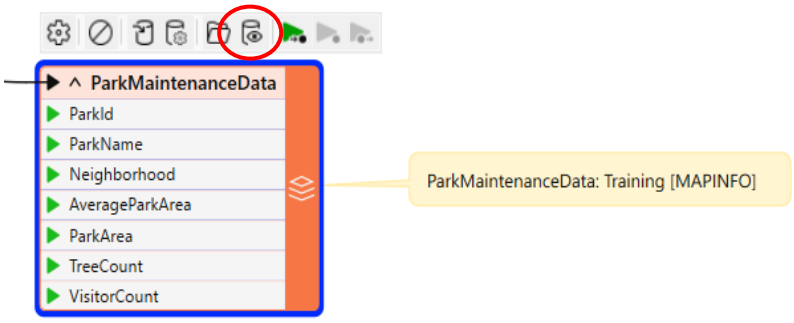
ParksMaintenanceData output datasets should have been written out to both CSV and MapInfo TAB format:



2.1.26 Examine the outputs

We can examine the output in a few of ways including;

- From the Windows Explorer using the **Data Inspector**
- Or, back in the Workbench using the **View Written Data** button and the **Visual Preview** tools.



Table

ParkMaintenanceData — Columns...

	ParkId	ParkName	Neighborhood	AverageParkArea	ParkArea	TreeCount	VisitorCount
1	1	<missing>	Kitsilano	67916.067586245	448.124680666...	10	9406
2	2	Rosemary Brow...	Kitsilano	67916.067586245	1035.077080825	8	13100
3	3	Tea Swamp Park	Mount Pleasant	67916.067586245	2631.26398618...	2	11275
4	4	<missing>	Strathcona	67916.067586245	1984.836357179	6	9755
5	5	Morton Park	West End	67916.067586245	2197.318199651	4	14977
6	6	Mcbride Park	Kitsilano	67916.067586245	17125.7170839...	9	15053
7	7	Granville Park	Fairview	67916.067586245	19655.7053629...	13	15185
8	8	<missing>	Mount Pleasant	67916.067586245	3123.72307219...	3	8061

Search in any column — 1 selected / 80 row(s)

Advanced Exercise

Notice that the numbers in the Table View show the results have been calculated to 12 decimal places. This is in excess of the precision that you require. As an advanced task - if you have time - use the AttributeRounder transformer to reduce the values to just 2 decimal places.

Congratulations

By Completing this exercise you have learned how to:

- Edit the output feature type name on a writer schema
- Edit the attributes on a writer schema
- Use Attribute Definition modes Automatic and Manual
- Copy attribute definition from another feature type or transformer within the workspace
- Carry out schema mapping using transformers; AttributeManager, AttributeRenamer
- Carry out content transformation with transformers; AreaCalculator, StatisticsCalculator
- Output to multiple writers in a single workspace

2.2 Generate Summary Statistics

Demonstrates	Using the StatisticsCalculator transformer to generate summary statistics
Overall Goal	Calculate summary statistics for the city parks at neighborhood level
Data	Parks (MapInfo TAB)
Start Workspace	C:\FMEModularData\Workspaces\2.02-Attributes-SummaryStatistics-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\2.02-Attributes-SummaryStatistics-Complete.fmw

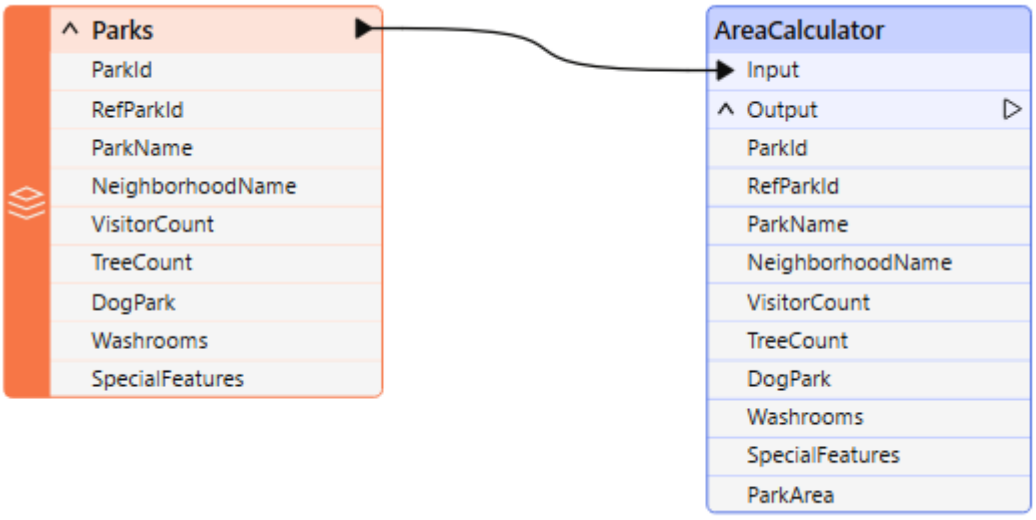
The parks team has decided that they want to know the average size of parks in each neighborhood. They also want to know the smallest and largest park size for each neighborhood. So let's generate these summary stats for them.

2.2.1 Launch FME Workbench and Open Workspace

A workspace already exists with the first part of our required workflow already setup.

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open Workspace. Navigate to and open C:\FMEModularData\Workspaces\2.02-Attributes-SummaryStatistics-Begin.fmw

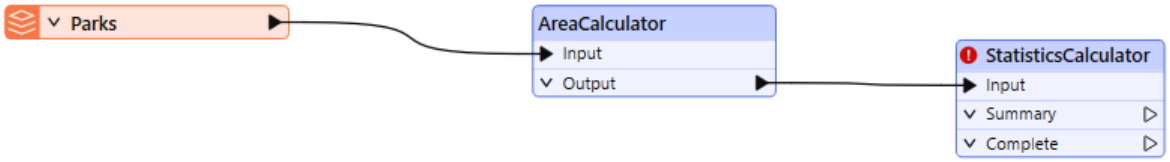
The workspace contains a reader for the Parks data and an AreaCalculator; calculating the size of each park (in an attribute called ParkArea):



2.2.2 Add a StatisticsCalculator Transformer

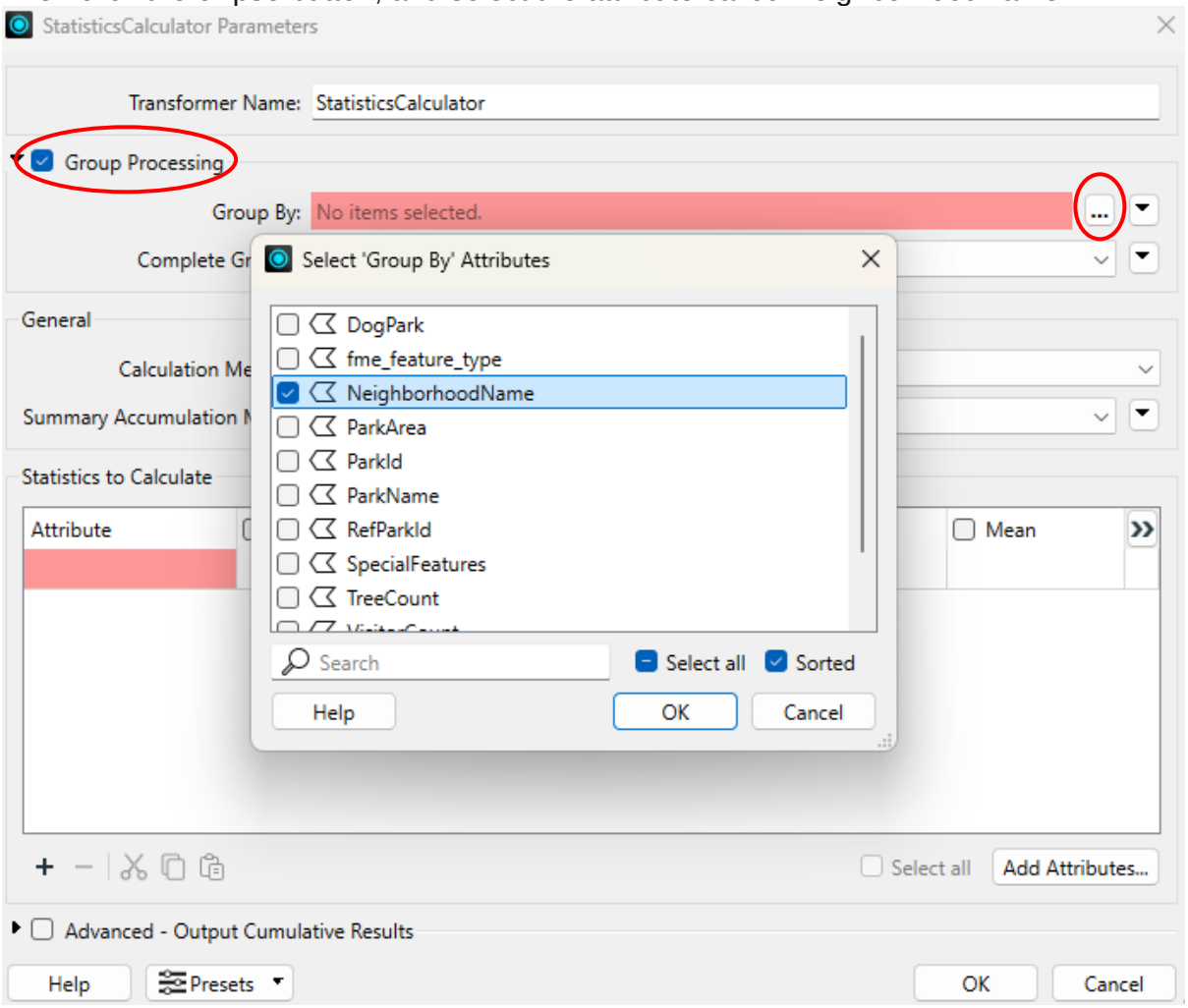
Find and place a StatisticsCalculator transformer onto the canvas. Then connect it to the

output port of the AreaCalculator.



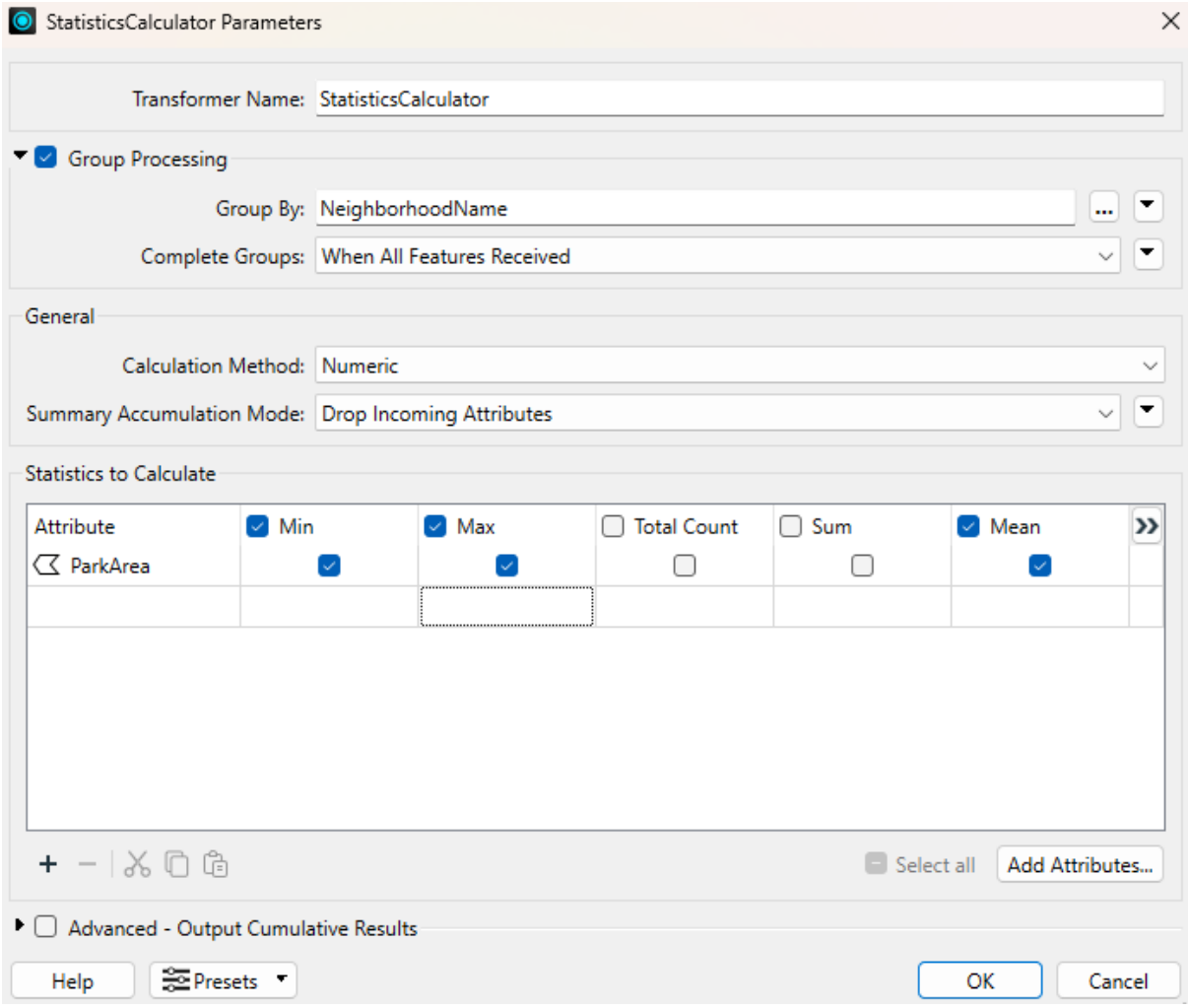
2.2.3 Configure the Group-By Parameter in the StatisticsCalculator

Within the transformer tick to use *Group Processing*
Then click the ellipse button, and select the attribute called NeighborhoodName



2.2.4 Configure the Statistics to Calculate

Within the Statistics to Calculate section we now need to specify the attribute to analyse and choose the statistics that we want to calculate.
Select the attribute ParkArea. Then tick to calculate Min, Max and Mean:



2.2.5 Save and Run Workspace

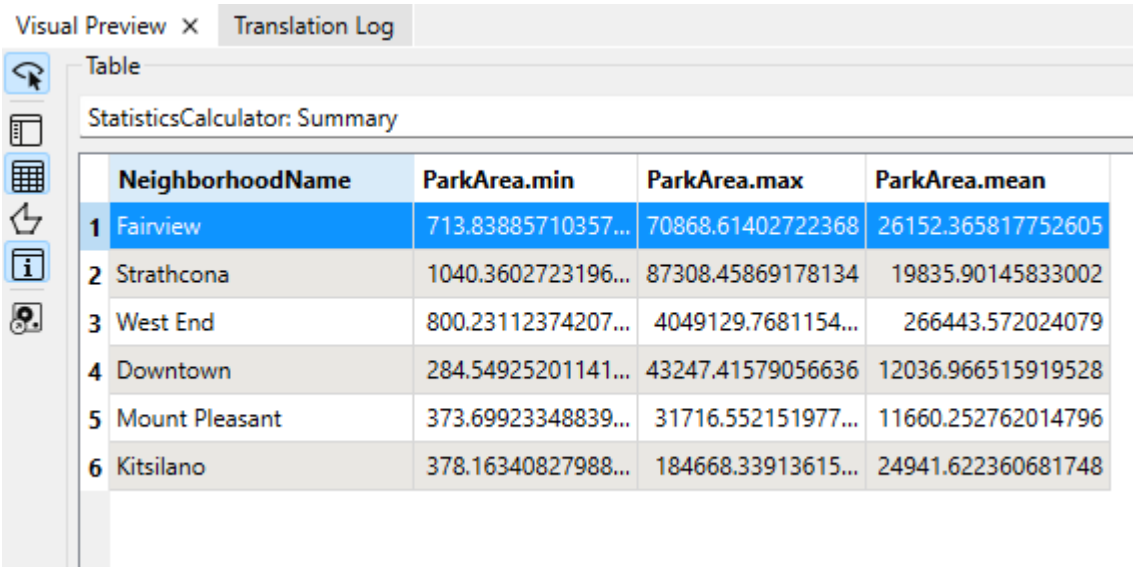
Save the workspace, then Run the workspace by clicking the Run button on the toolbar, or by using Run > Run Workspace on the menu bar.

The Translation Parameters dialog may display. Click Run to proceed.

2.2.6 Examine the StatisticsCalculator Feature Cache

Click on the Summary port feature cache of the StatisticsCalculator to examine the features at this point.

Within the Visual Preview window use the Toggle Table View button to display the table view, where we can then examine the summary statistics that have been calculated for each neighborhood.



	NeighborhoodName	ParkArea.min	ParkArea.max	ParkArea.mean
1	Fairview	713.83885710357...	70868.61402722368	26152.365817752605
2	Strathcona	1040.3602723196...	87308.45869178134	19835.90145833002
3	West End	800.23112374207...	4049129.7681154...	266443.572024079
4	Downtown	284.54925201141...	43247.41579056636	12036.966515919528
5	Mount Pleasant	373.69923348839...	31716.552151977...	11660.252762014796
6	Kitsilano	378.16340827988...	184668.33913615...	24941.622360681748

Let's tidy up the attribute names.

2.2.7 Add an AttributeManager transformer

Find and place an AttributeManager transformer onto the canvas. The StatisticsCalculator transformer has two output ports, we want to take the results from the Summary port.

Use the AttributeManager to rename the attributes as follows:

Input Attribute	Output Attribute
ParkArea.min	SmallestParkSize
ParkArea.max	LargestParkSize
ParkArea.mean	AverageParkSize

The final step is to output the statistics to CSV format.

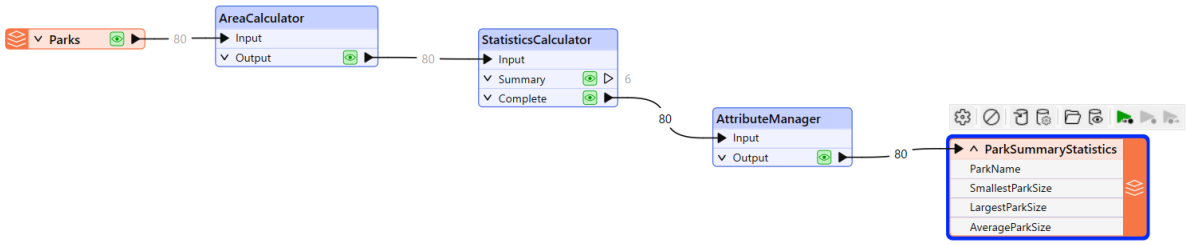
2.2.8 Add a Writer

Use either the Writer button, or use Writers > Add Writer… from the menu bar
The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	CSV (Comma Separated Value)
Writer Dataset	C:\FMEModularData\Output\Training

Also set the CSV File Definition: to *Automatic…*

Within the Feature Type dialog set the CSV File Name: to *ParksSummaryStatistics*
Connect the ParksSummaryStatistics feature type to the AttributeManager.



2.2.9 Resave and Run Workspace

Save the workspace, then Run the workspace by clicking the Run button on the toolbar, or by using Run > Run Workspace on the menu bar.
The Translation Parameters dialog may display. Click Run to proceed.

Congratulations

By Completing this exercise you have learned how to:

- Use the StatisticsCalculator transformer to generate summary statistics
- Use the Group-by parameter in FME transformers

2.3 Create, View and Use List Attributes

Demonstrates	Use a transformer to generate List attributes. Inspect list attributes using the Feature Information window. Extract data from list attributes to create a new attribute.
Overall Goal	Use List attribute generated by a transformer to create a new attribute
Data	Addresses Full (GeoPackage)
Start Workspace	none
End Workspace	C:\FMEModularData\Workspaces\Complete\2.03-Attributes-ListAttributes-Complete.fmw

We have some address data for the city. We need to generate a new ID for each record. This new ID will be based on a component of an existing attribute called GlobalID.

We will first split the GlobalID into its component parts, using a transformer. These component parts will be output as a list attribute. Which we will examine, then extract the required data to create our new ID attribute.

2.3.1 Launch FME Workbench and Create new Workspace from Blank

Launch the FME Workbench, if it isn't open already.
Within the Get Started section of the Workbench, click on Blank Workspace.

2.3.2 Add Reader

Use either the Reader button, or use Readers > Add Reader… from the menu bar.

The Add Reader dialog will open, in which define the Format and Dataset settings as follows:

Reader Format	OGC GeoPackage
Reader Dataset	C:\FMEModularData\Data\Addresses\Addresses Full.gpkg

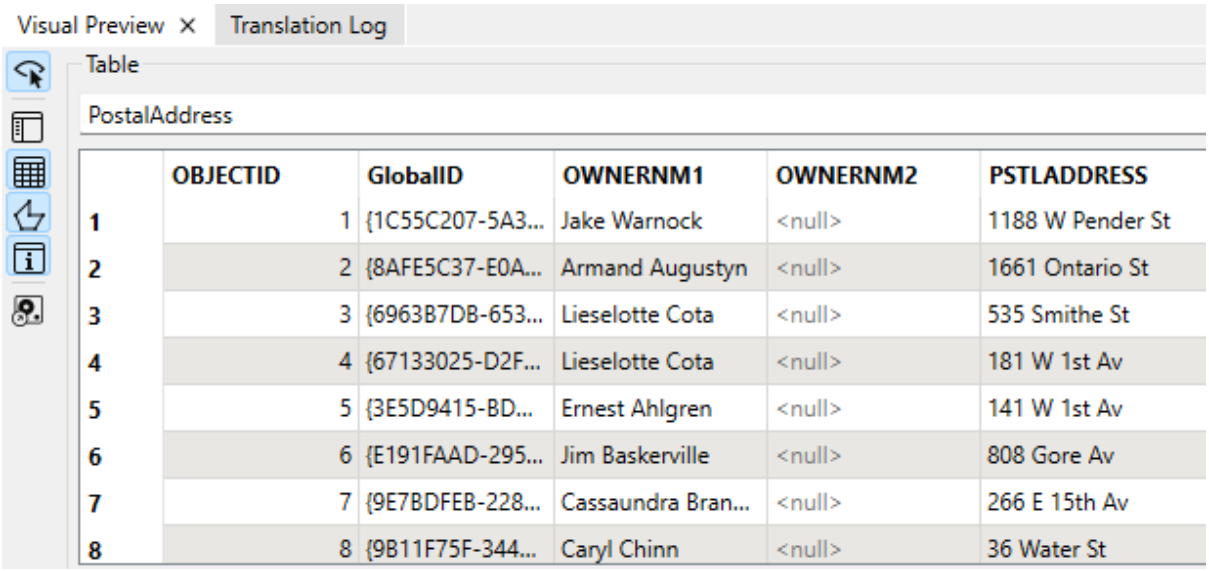
Into the Navigator panel a GeoPackage Reader will be added and on the canvas there will be a single reader feature type for PostalAddress.

2.3.3 Examine the PostalAddress Data

Before we start designing our workflow, first examine the data. Select the PostalAddress feature type on the canvas (so it becomes highlighted in blue) then use the View Source Data button.

Within the Visual Preview panel view the data within a Table view. This will display the PostalAddress attributes and attribute values.

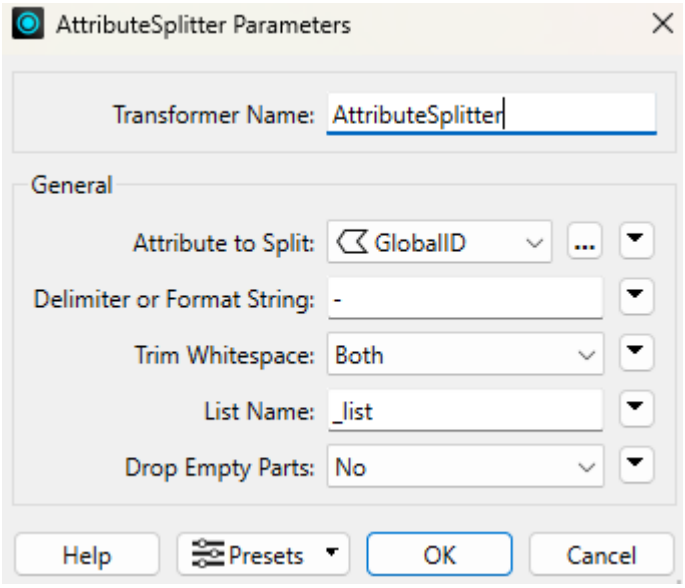
In this exercise we are interested in splitting the GlobalID into its component parts.



	OBJECTID	GlobalID	OWNERNM1	OWNERNM2	PSTLADDRESS
1	1	{1C55C207-5A3...	Jake Warnock	<null>	1188 W Pender St
2	2	{8AFE5C37-E0A...	Armand Augustyn	<null>	1661 Ontario St
3	3	{6963B7DB-653...	Lieselotte Cota	<null>	535 Smithe St
4	4	{67133025-D2F...	Lieselotte Cota	<null>	181 W 1st Av
5	5	{3E5D9415-BD...	Ernest Ahlgren	<null>	141 W 1st Av
6	6	{E191FAAD-295...	Jim Baskerville	<null>	808 Gore Av
7	7	{9E7BDFEB-228...	Cassaundra Bran...	<null>	266 E 15th Av
8	8	{9B11F75F-344...	Caryl Chinn	<null>	36 Water St

2.3.4 Add and configure an AttributeSplitter Transformer

Find and add an AttributeSplitter transformer, then connect it to the PostalAddress feature type.



This transformer splits attribute values into parts, based on a delimiter or fixed-width pattern, and creates a list attribute containing one list element for each part.
Configure the transformer parameters as follows:

Attribute to Split: *GlobalID*
Delimiter or Format String: -

2.3.5 Save and Run the Workspace

Click on the Save button, or use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.
Then run the workspace.

2.3.6 Inspect the AttributeSplitter Feature Cache

Click on the AttributeSplitter transformer feature cache to examine the features at this point in the workflow.

Within the Visual Preview window use the Toggle Table View button to display the table view, where we can then examine the attributes.

The attributes will be unchanged. Where are the List attribute results?

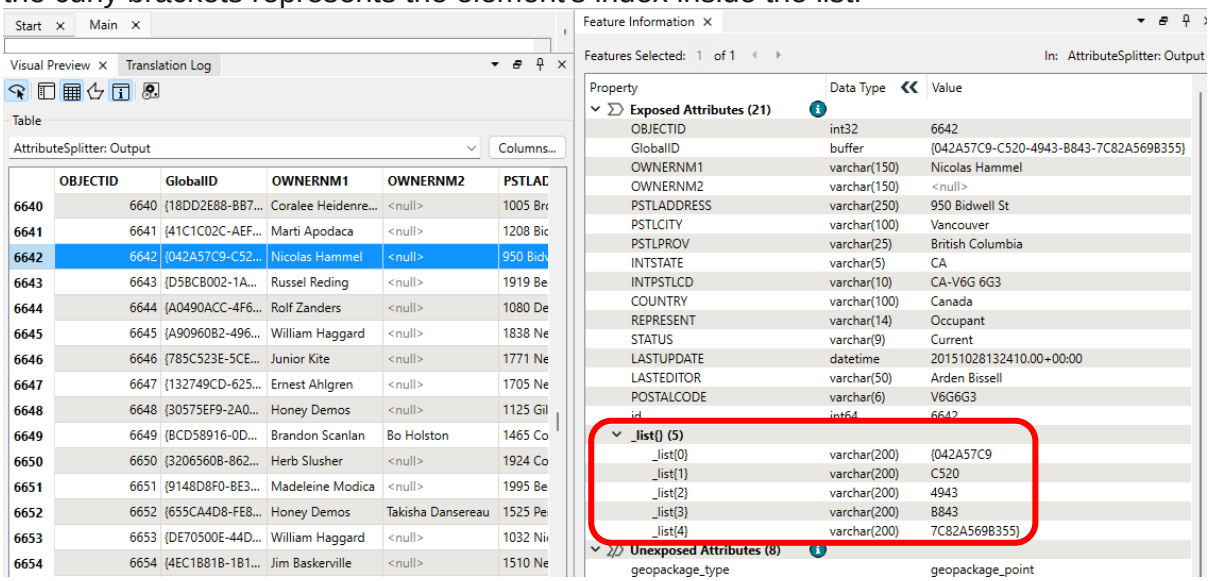
Because list attributes contain more values than can fit in a single cell of the Table View, list attributes cannot be exposed and will not appear in Table View or, in most cases, written data. You can inspect list attributes using the Feature Information Window.

2.3.7 Inspect List Attributes

Within the Visual Preview panel use the Show/Hide Feature Information Window toggle button.

Within the Table View select a feature.
Then look in the Feature Information Window under Attributes. The list attribute will be visible.

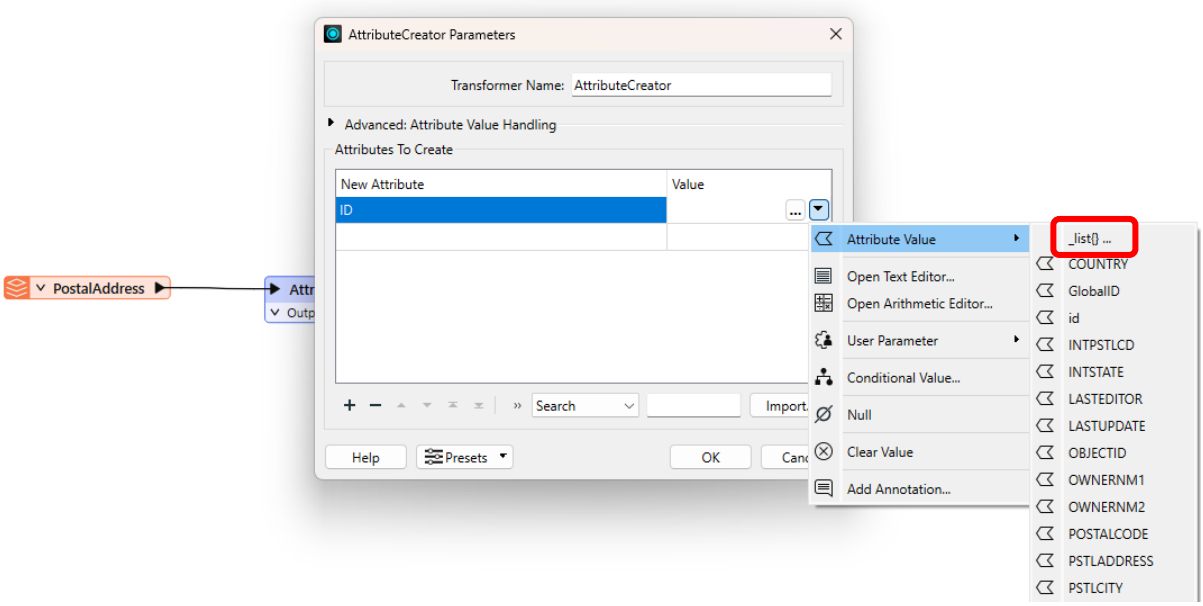
List attributes are denoted using curly brackets after the list name and a number inside the curly brackets represents the element's index inside the list.



2.3.8 Use List Elements to Create a New Attribute

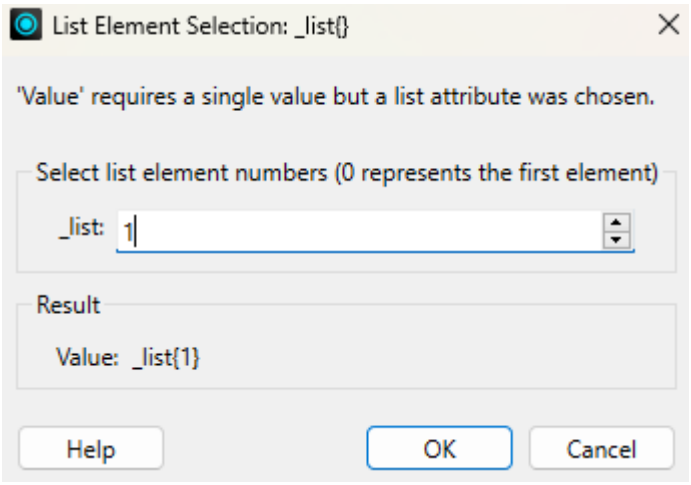
There are a couple of transformers that could be used to create the new attribute. Let's use the AttributeCreator. Find and add an AttributeCreator transformer, then connect it to the AttributeSplitter transformer.

Within the transformer parameters, choose to create a new attribute called *ID*.
Notice that the list attribute is available in the attribute dropdown menu:

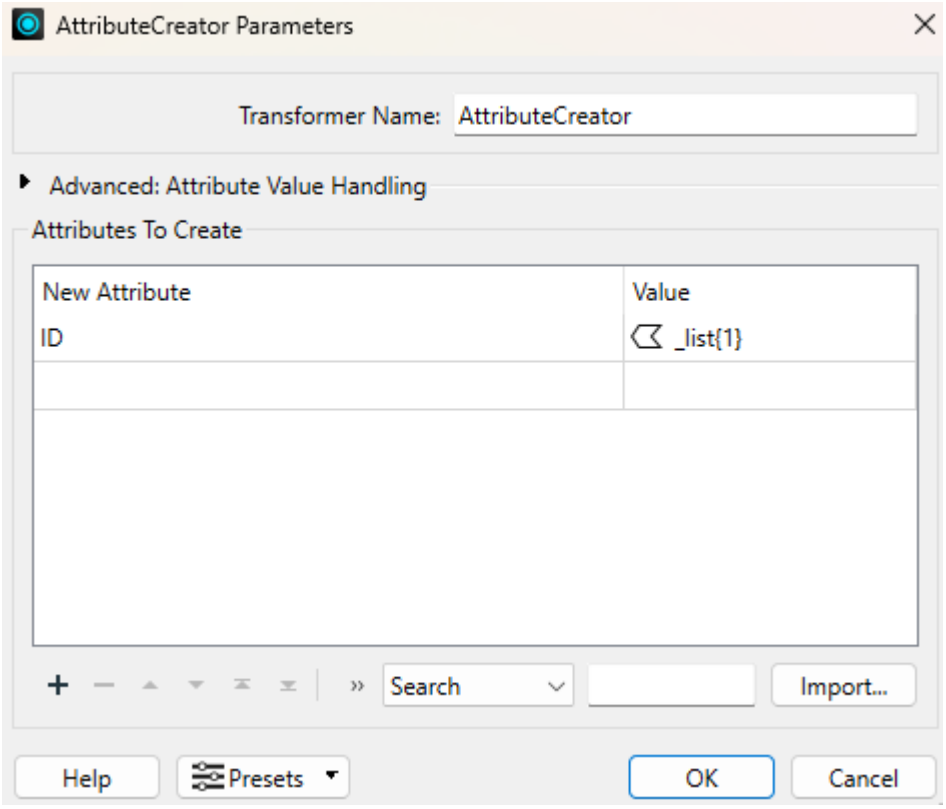


We want to use the second list element of the list attribute to build our new ID attribute. So we need to use the Attribute list to select the list attribute.

Within the List Element Selection window, change the list element to *1*



Click ok to return to the AttributeCreator.



2.3.9 Resave and Run the Workspace

Click on the Save button, or use File > Save As on the menu bar.
Then run the workspace.

2.3.10 Inspect the AttributeCreator Feature Cache

Click on the AttributeCreator transformer feature cache to examine the features at this point in the workflow.

Within the Visual Preview window use the Toggle Table View button to display the table view, where we can then examine the attributes.

Visual Preview | Translation Log

Table

AttributeCreator: Output

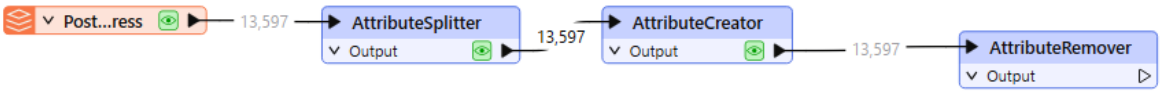
	ID	OBJECTID	GlobalID	OWNERNM1	OWNERNM2	PSTLADDRESS
1	5A3E	1	{1C55C207-5A3...	Jake Warnock	<null>	1188 W Pender St
2	E0A7	2	{8AFE5C37-E0A...	Armand Augustyn	<null>	1661 Ontario St
3	653A	3	{6963B7DB-653...	Lieselotte Cota	<null>	535 Smithe St
4	D2F6	4	{67133025-D2F...	Lieselotte Cota	<null>	181 W 1st Av
5	BDD8	5	{3E5D9415-BD...	Ernest Ahlgren	<null>	141 W 1st Av
6	295B	6	{E191FAAD-295...	Jim Baskerville	<null>	808 Gore Av
7	2288	7	{9E7BDFEB-228...	Cassaundra Bran...	<null>	266 E 15th Av
8	3443	8	{9B11F75F-344...	Caryl Chinn	<null>	36 Water St
9	2B6A	9	{42946EE8-2B6...	Gerald Woodburn	<null>	2762 W 3rd Av
10	95E0	10	{591FE3D2-95E...	Many Purdy	<null>	989 Main St

The new ID attribute has been created and populated with the specified list attribute element.

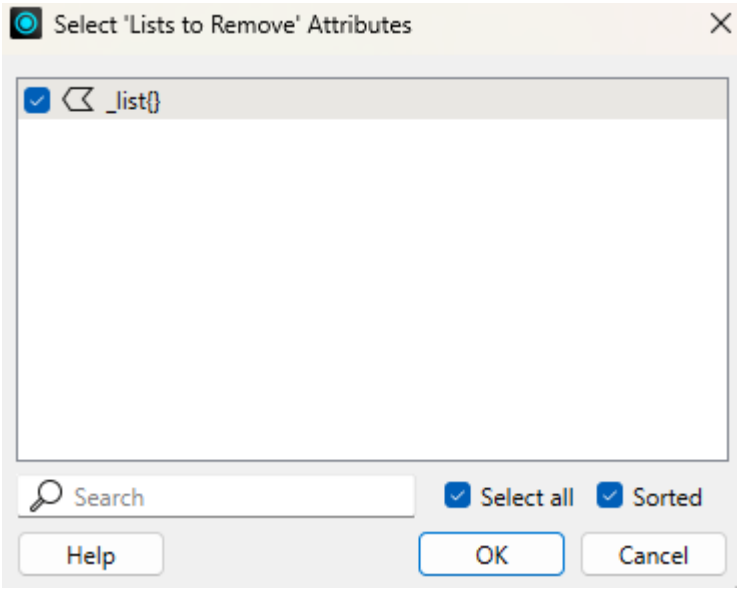
2.3.11 Remove List Attributes

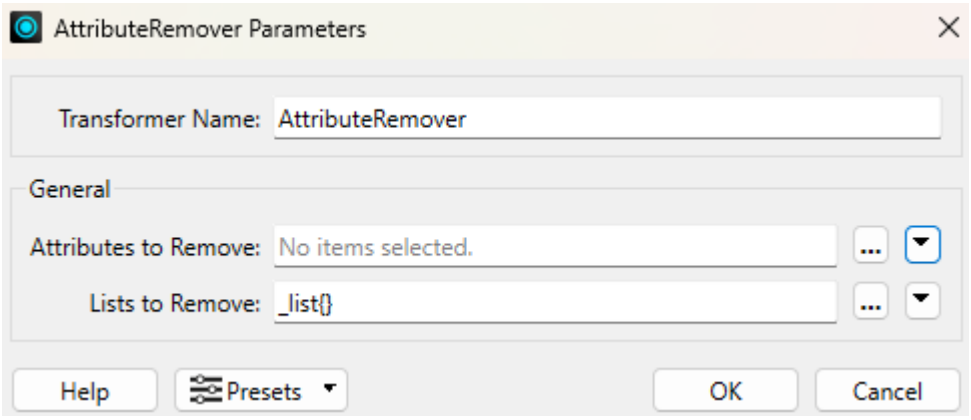
Finally, we need to clean-up any unwanted list attributes, by removing them. There are a couple of transformers that we could use; AttributeManager or AttributeRemover.

To the canvas find and add an AttributeRemover. Then connect it to the AttributeCreator.



Within the AttributeRemover parameters click on the ellipse button for *Lists to Remove:* Select the *_list{}* attribute and click OK.





2.3.12 Resave and Run the Workspace

Click on the Save button, or use File > Save As on the menu bar.
Then run the workspace.

FME Lizard

Did you notice that there was an existing attribute called 'id' on the dataset. Our new attribute is called 'ID'. FME is case sensitive when it comes to attribute names, so recognises that they are different attributes.

Maybe it would have been more appropriate to call our new attribute something else (maybe 'Ref') to avoid later confusion. However this highlights how important it is to be mindful of ***case*** *and attribute names.*

This is particularly important when merging multiple datasets together – ensure that their attribute names are the same including their case! The BulkAttributeRenamer transformer is very handy for setting the case of all or selected attribute names.

Congratulations

By Completing this exercise you have learned how to:

- Use the AttributeSplitter to split attributes into a list attribute
- Inspect List elements using the Feature Information window
- Create a new attribute using List elements

2.4 Testing to Filter Data

Demonstrates	Filter features into multiple data streams using the TestFilter. Using multiple Test Clauses.
Overall Goal	Filter the parks into groups using Tests, then assign the park category as a new attribute.
Data	Parks (MapInfo TAB)
Start Workspace	none
End Workspace	C:\FMEModularData\Workspaces\Complete\2.04-Attributes-TestingFiltering-Complete.fmw

The Council's Environment Directorate want to categorize the city Parks into 'premier', 'destination' or 'local' - based on their features. We'll use testing to filter the parks into groups, then assign the park category as a new attribute.

2.4.1 Launch FME Workbench and Create new Workspace from Blank

Launch the FME Workbench, if it isn't open already.
Within the Get Started section of the Workbench, click on Blank Workspace.

2.4.2 Add Reader

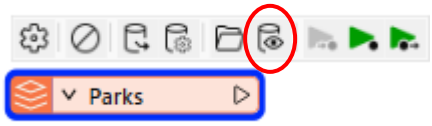
Use either the Reader button, or use Readers > Add Reader… from the menu bar.
The Add Reader dialog will open, in which define the Format and Dataset settings as follows:

Reader Format	Precisely MapInfo TAB (MAPINFO)
Reader Dataset	C:\FMEModularData\Data\Parks\Parks.tab

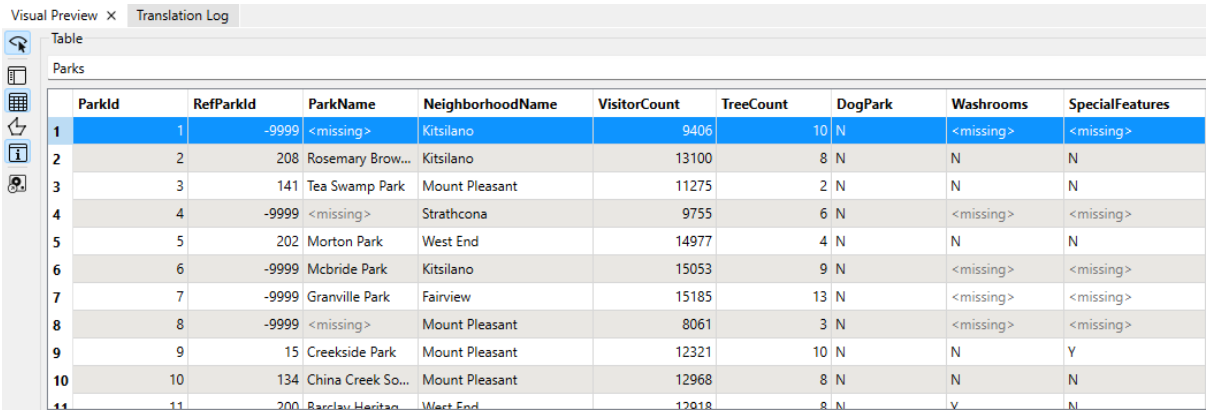
Into the Navigator panel a MapInfo Reader will be added and on the canvas there will be a single reader feature type for Parks.

2.4.3 Examine the Parks Data

Before we start designing our workflow, first examine the parks data. Select the Parks reader feature type on the canvas (so it becomes highlighted in blue) then use the View Source Data button.



Within the Visual Preview panel view the data within a Table view. This will display the Parks attributes and attribute values.



	ParkId	RefParkId	ParkName	NeighborhoodName	VisitorCount	TreeCount	DogPark	Washrooms	SpecialFeatures
1	1	-9999	<missing>	Kitsilano	9406	10	N	<missing>	<missing>
2	2	208	Rosemary Brow...	Kitsilano	13100	8	N	N	N
3	3	141	Tea Swamp Park	Mount Pleasant	11275	2	N	N	N
4	4	-9999	<missing>	Strathcona	9755	6	N	<missing>	<missing>
5	5	202	Morton Park	West End	14977	4	N	N	N
6	6	-9999	Mcbride Park	Kitsilano	15053	9	N	<missing>	<missing>
7	7	-9999	Granville Park	Fairview	15185	13	N	<missing>	<missing>
8	8	-9999	<missing>	Mount Pleasant	8061	3	N	<missing>	<missing>
9	9	15	Creekside Park	Mount Pleasant	12321	10	N	N	Y
10	10	134	China Creek So...	Mount Pleasant	12968	8	N	N	N
11	11	200	Barclay Heritag...	West End	12918	8	N	Y	N

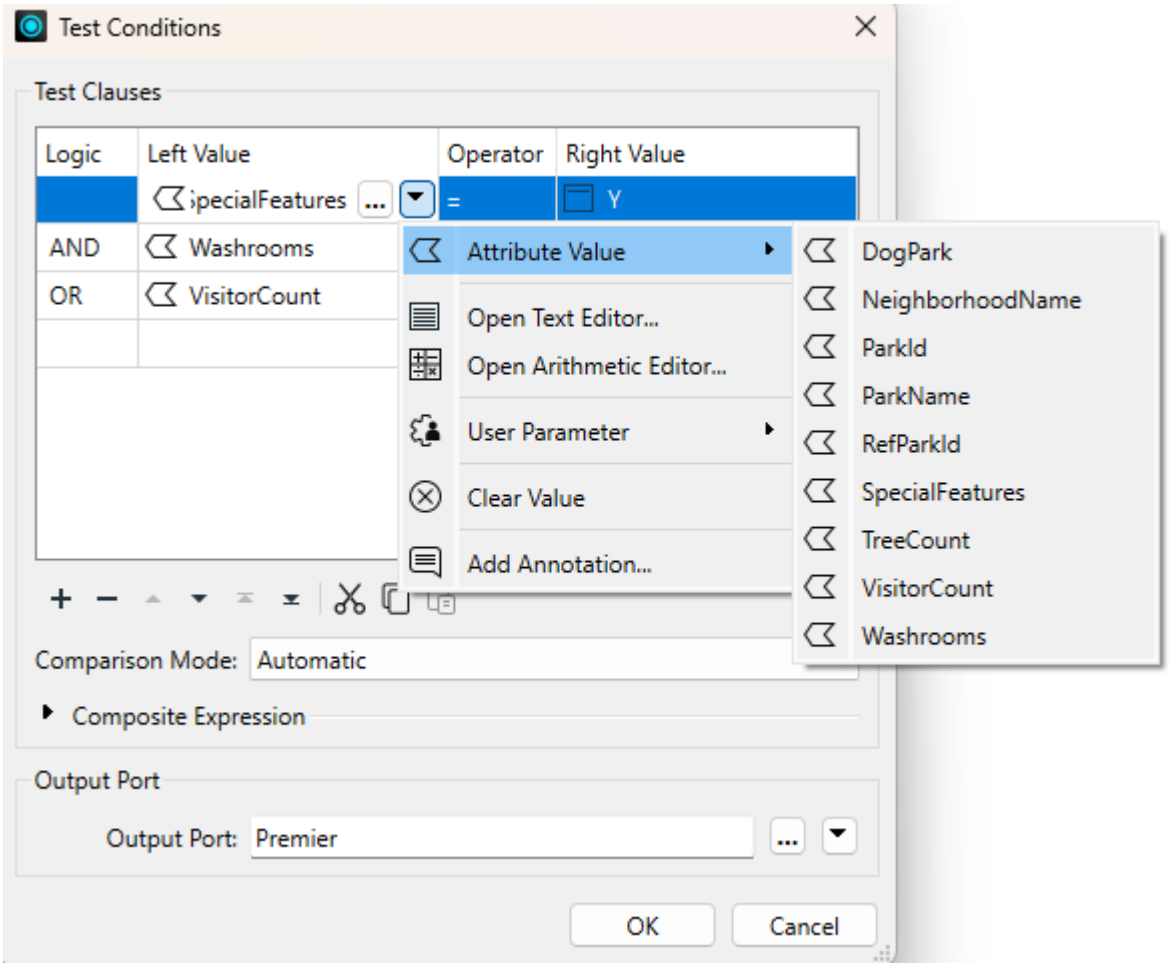
The parks need to be filtered into three groups:

Category	Criteria
Premier	Has Special Features and Washrooms Or the Visitor Count is greater than 20000
Destination	Has Special Features or Washrooms
Local	Everything else

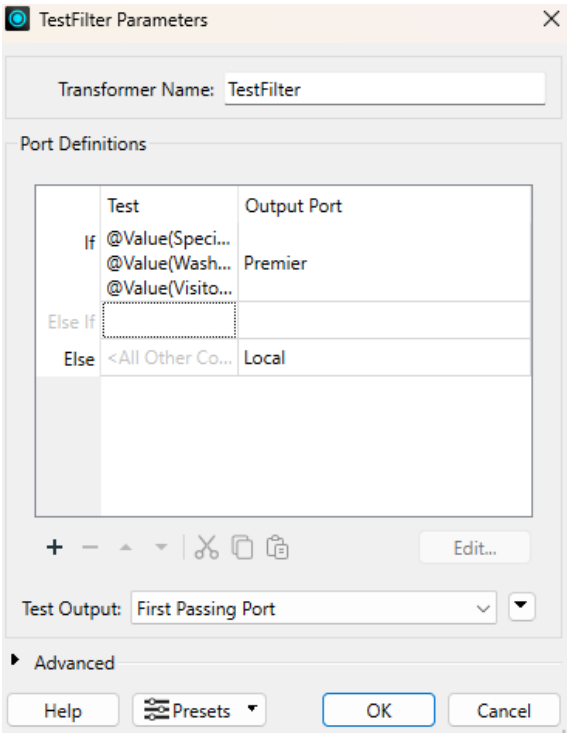
2.4.4 Add a TestFilter Transformer

Add a TestFilter to the canvas, then connect it to the Parks feature type. The TestFilter transformer filters features by test conditions to one or more output ports. We are going to need three output ports.

Go into the TestFilter parameters and set the first Test Condition; double-click on the *If* cell below Test Condition.
The Test Conditions dialog will open. In here we need to set multiple Test Clauses to filter features to meet the criteria for the Premier category. Also set the *Output Port:* name to *Premier*

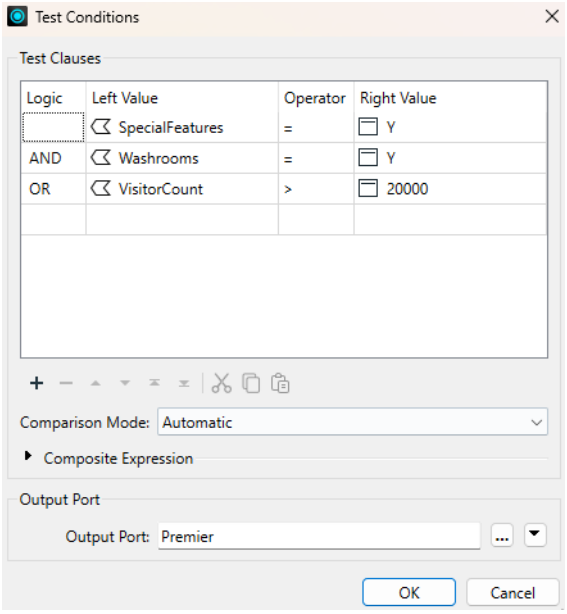


Click OK to return to the TestFilter Parameters dialog.



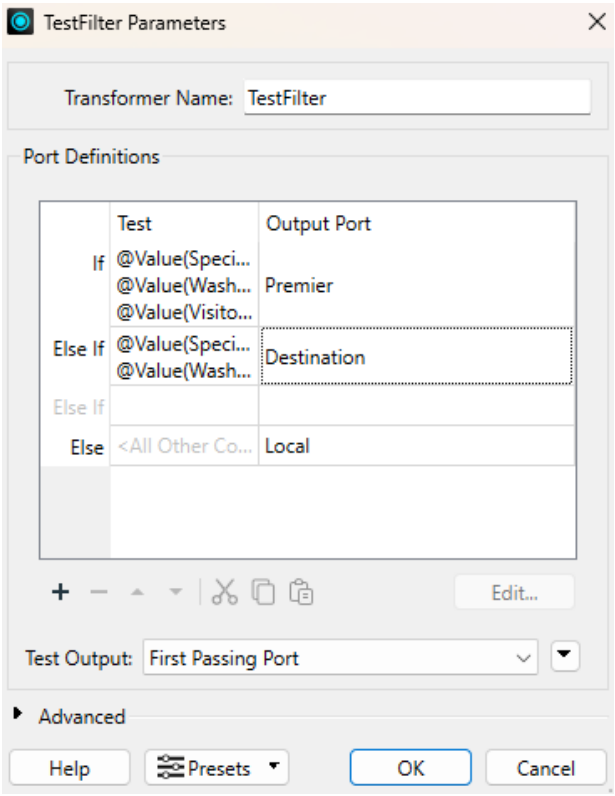
Next we need to set the Test Conditions for the Destination category. Double-click on

the *Else If* cell below our first Test Condition.
The Test Conditions dialog will open. In here we need to set multiple Test Clauses to filter features to meet the criteria for the Destination category. Also set the *Output Port:* name to *Destination*



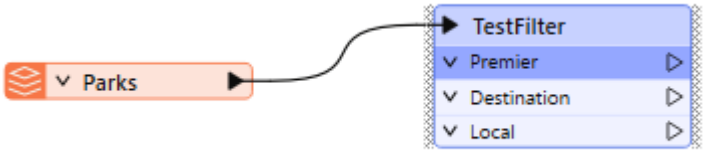
Click OK to return to the TestFilter Parameters dialog.

All other parks will be categorized as Local. Rename the <UNFILTERED> Output Port to *Local.*



Click Ok to return to the canvas.

The transformer output ports have now updated to reflect our TestFilter configuration.

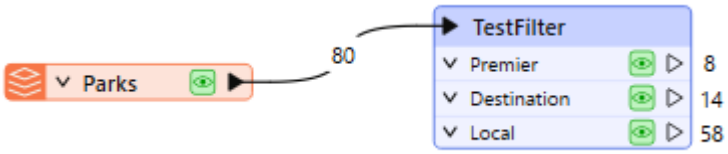


2.4.5 Save and Run the Workspace

Click on the Save button, or use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.
Then run the workspace.

2.4.6 Inspect the TestFilter Feature Caches

There will be a feature cache for each output port of the TestFilter. Examine the features from each output port.



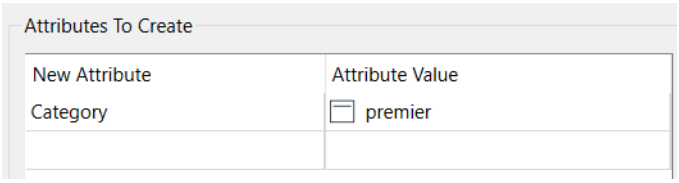
Within the Visual Preview window use the Toggle Table View button to display the table view, where we can then examine the attributes.

The parks features have now been filtered into three groups, based on the test conditions.

2.4.7 Create new Category Attribute

For each data stream we will create the new Category attribute and populate it with the appropriate value; 'premier', 'destination' or 'local'.

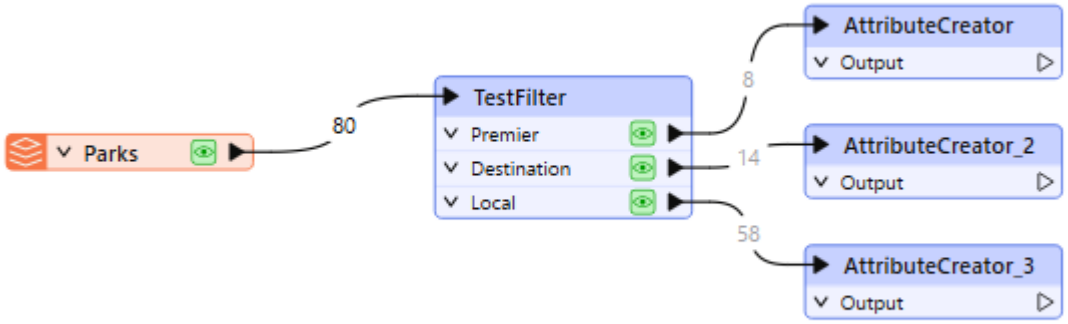
Add an AttributeCreator transformer to the canvas, then connect it to the Premier output port of the TestFilter.
Open the AttributeCreator parameters and configure as follows:
New Attribute Name: *Category*
Attribute Value: *premier*



Add another AttributeCreator transformer to the canvas, then connect it to the Destination output port of the TestFilter.

Open the AttributeCreator parameters and configure as follows:
New Attribute Name: *Category*
Attribute Value: *destination*

Add a third AttributeCreator transformer to the canvas, then connect it to the Local output port of the TestFilter.
Open the AttributeCreator parameters and configure as follows:
New Attribute Name: *Category*
Attribute Value: *local*



The final step is to output the revised parks data to a new MapInfo TAB dataset.

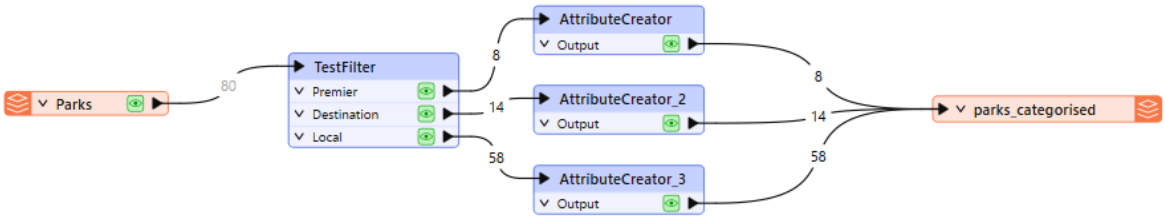
2.4.8 Add a Writer

Use either the Writer button, or use Writers > Add Writer… from the menu bar
The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	Precisely MapInfo TAB (MAPINFO)
Writer Dataset	C:\FMEModularData\Output\Training

Also set the Table Definition: to *Automatic…*
Set the Table Name: to *parks_categorised*

Connect the *parks_categorised* feature type to the three AttributeCreators



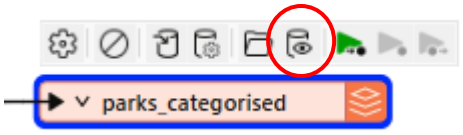
2.4.9 Resave and Run Workspace

Save the workspace, then Run the workspace by clicking the Run button on the toolbar, or by using Run > Run Workspace on the menu bar.
The Translation Parameters dialog may display. Click Run to proceed.

2.4.10 Examine the output

We can examine the output in a few of ways including;

- From the Windows Explorer using the **Data Inspector**
- Or, back in the Workbench using the **View Written Data** button and the **Visual Preview** tools.



Table

parks_categorised

	Category	ParkId	RefParkId	ParkName	NeighborhoodName	VisitorCount	TreeCount	DogPark	Washrooms	SpecialFeatures
1	local	55	48	Willow Park	Fairview	7123	7	N	N	N
2	local	36	32	Wendy Poole P...	Downtown	10250	4	N	N	N
3	local	44	121	Volunteer Park	Kitsilano	15094	11	N	N	N
4	local	34	31	Victory Square	Downtown	8769	5	N	N	N
5	premier	51	120	Vanier Park	Kitsilano	13245	59	N	Y	Y
6	local	20	183	Thornton Park	Strathcona	18735	12	N	N	N
7	local	3	141	Tea Swamp Park	Mount Pleasant	11275	2	N	N	N
8	desination	43	119	Tatlow Park	Kitsilano	12988	7	N	Y	N
9	desination	52	47	Sutcliffe Park	Fairview	18404	11	N	N	Y

FME Lizard

This workspace will generate the desired results, however we have duplicated effort and used multiple AttributeCreator transformers when creating the new Category attribute for each data stream. There is a better way to approach this workspace design; where our TestFilter and three AttributeCreators could all be replaced with a single transformer!

Resulting in a more efficient workspace.

We cover how to do this using Conditional Values on our 'Advanced Attribute Handling and Lists' training module

Congratulations

By Completing this exercise you have learned how to:

- Filter features into multiple data streams using the TestFilter
- Using multiple Test Clauses
- Work with multiple data streams within a workspace
- Set a fixed (constant) attribute value

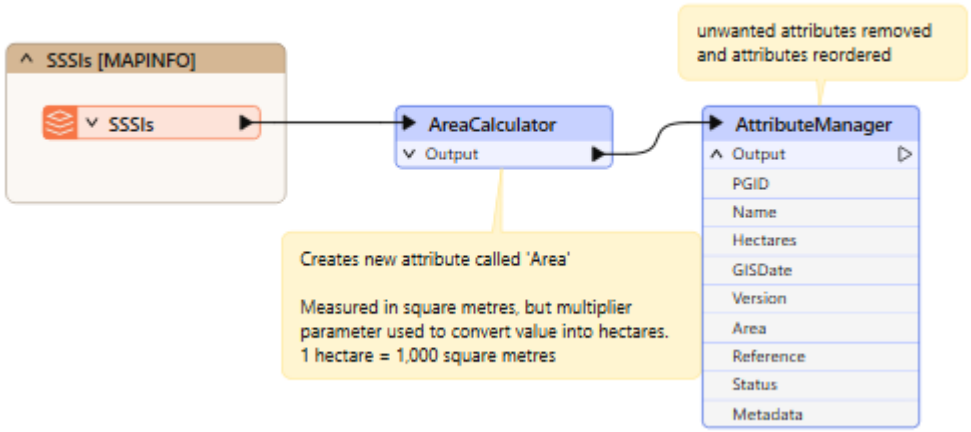
2.5 Testing to QA Data

Demonstrates	Prepare numerical values for direct comparison, using an AttributeRounder Using Tests to filter features Set a fixed (constant) attribute value
Overall Goal	Validate the Hectares attribute, and correct if required. Update and create attributes to document modifications made to the dataset.
Data	SSSI (MapInfo TAB)
Start Workspace	C:\FMEModularData\Workspaces\2.05-Attributes-TestingQA-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\2.05-Attributes-TestingQA-Complete.fmw

We have a shapefile containing data about Sites of Special Scientific Interest, this needs validating before it can be sent to a client.

2.5.1 Launch FME Workbench and Open Existing Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open. Navigate to and open: C:\FMEModularData\Workspaces\2.05-Attributes-TestingQA-Begin.fmw



The workspace includes a MapInfo reader, bringing in a SSSI dataset. The SSSI dataset already contains an attribute called Hectares. However this is a static value on the source data, and we don't know when this was last populated, or if it's still correct.

We are going to compare the Hectare values to the actual size of each polygon. The workspace already has the first step setup for us; an AreaCalculator transformer is being used to measure the size of each SSSI polygon. Then an AttributeManager is removing any unwanted attributes.

2.5.2 Run Workspace and Examine Attributes

Run the workspace.
Then click on the AttributeManager transformer feature cache to examine the features at this point in the workflow.
Within the Visual Preview window use the Toggle Table View button to display the table view, where we can then examine the attributes.

Table

AttributeManager: Output

	PGID	Name	Hectares	GISDate	Version	Area	Reference	Status	Metadata
1	421	Mantles Heath	13.6534401132...	20031204	1	13.6539895539...	SP597552	Notified	<a href="https:...
2	453	Narborough Bog	8.52475675861...	20031204	1	5.76864733564...	SP549978	Notified	<a href="https:...
3	454	Narborough Bog	8.52475675861...	20031204	1	2.75587067430...	SP549978	Notified	<a href="https:...
4	999	Stanford Park	20.4425454035...	20031204	1	20.4429288468...	SP586792	Notified	<a href="https:...
5	1037	Burbage Wood ...	51.1496294473...	20140115	1	51.1501433993...	SP452940	Notified	<a href="https:...
6	1164	Croft and Hunc...	35.2514415808...	20140115	1	35.2510745382...	SP511964	Notified	<a href="https:...
7	1380	Kilby - Foxton C...	32.0928643874...	20140116	1	32.0937426820...	SP652959	Notified	<a href="https:...
8	1369	Stockton Railwa...	24.1395752936...	20140116	1	20.0281952327...	SP442643	Notified	<a href="https:...
9	1370	Stockton Railwa...	24.1395752936...	20140116	1	4.11195156554...	SP442643	Notified	<a href="https:...
10	1717	Misterton Mars...	6.76760704267...	20140527	1	6.76770800299...	SP556851	Notified	<a href="https:...
11	1745	Saddington Res...	19.1216629472...	20140527	1	19.1221710375...	SP663910	Notified	<a href="https:...

We are going to test that the Hectare value matches the newly measured Area value. Before we can compare these values, we first need to round them to the same level of decimal places.

2.5.3 Add and configure an AttributeRounder Transformer

Find and add an AttributeRounder transformer, then connect it to the AttributeManager.
This transformer rounds off attributes with numeric values to the specified number of decimal places.
Open the transformer parameters and configure as follows:
Attributes to Round: Area and *Hectares*
Decimal Places: 2

2.5.4 Save and Run the Workspace

Use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.
Then run the workspace.

2.5.5 Inspect the AttributeRounder Feature Cache

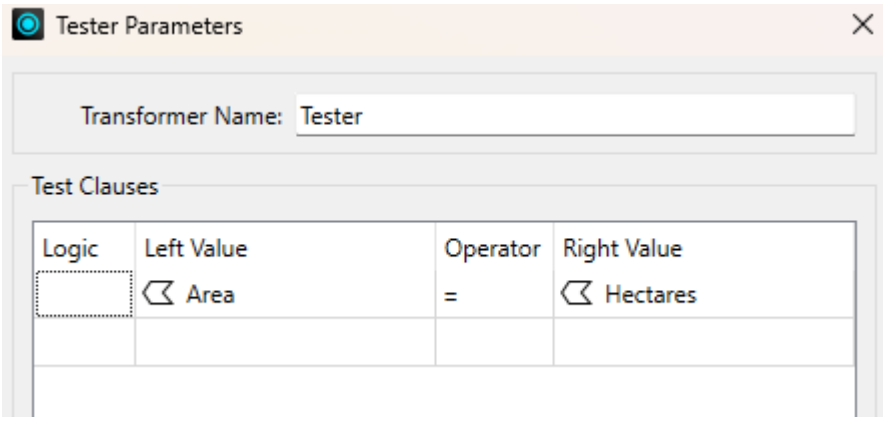
Click on the AttributeRounder transformer feature cache to examine the features at this point in the workflow.

Within the Visual Preview window use the Toggle Table View button to display the table view, where we can then examine the attributes. Both the 'Hectares' and 'Area' attribute values should now be rounded to 2 decimal places – ready for comparison.

2.5.6 Add and configure a Tester Transformer

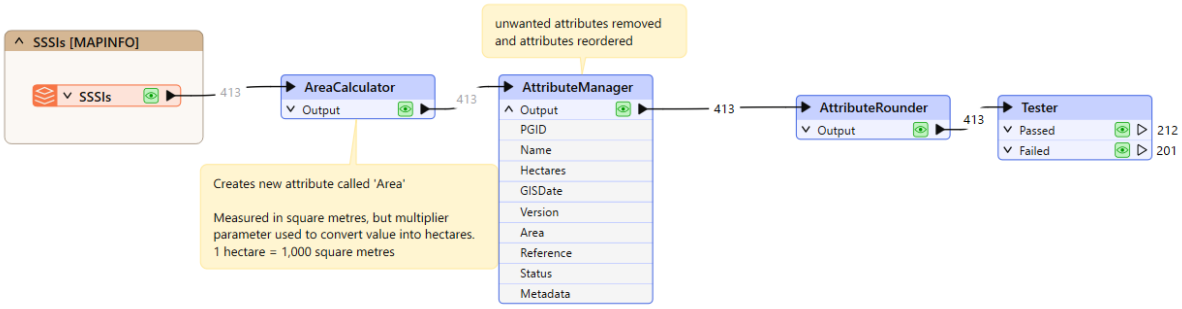
Find and add a Tester transformer, then connect it to the AttributeRounder.
Open the transformer and create a Test Clause in the first row as follows:

Click on the Left Value cell in the first row and use the drop-down arrow to select the first attribute that will be compared – 'Area'.
Move to the Operator cell and left click to define default '=' (or use drop-down arrow to access full list of FME operators).
Go to the Right Value and like before, select from the available attributes list 'Hectares'.



2.5.7 Resave and Rerun the Workspace

Click on the Save button, or use File > Save As on the menu bar.
Then run the workspace.



All the features outputted through the Passed port have Hectare values that match (are equal to) the Area value, and don't require correction. Features that fail this test need more attention. We will have to fix the 'Hectares' values that are wrong and mark all of them using new attribute.

2.5.8 Fix incorrect Hectares values and Manage Attributes

To fix the 'failed' features and complete the validation process, we have to do 3 things:

- update the 'Hectares' values with those calculated by FME (in Area attribute)
- increase the 'Version' number attribute
- and create a new 'Edits' attribute and populate it with the comment "area recalculated"

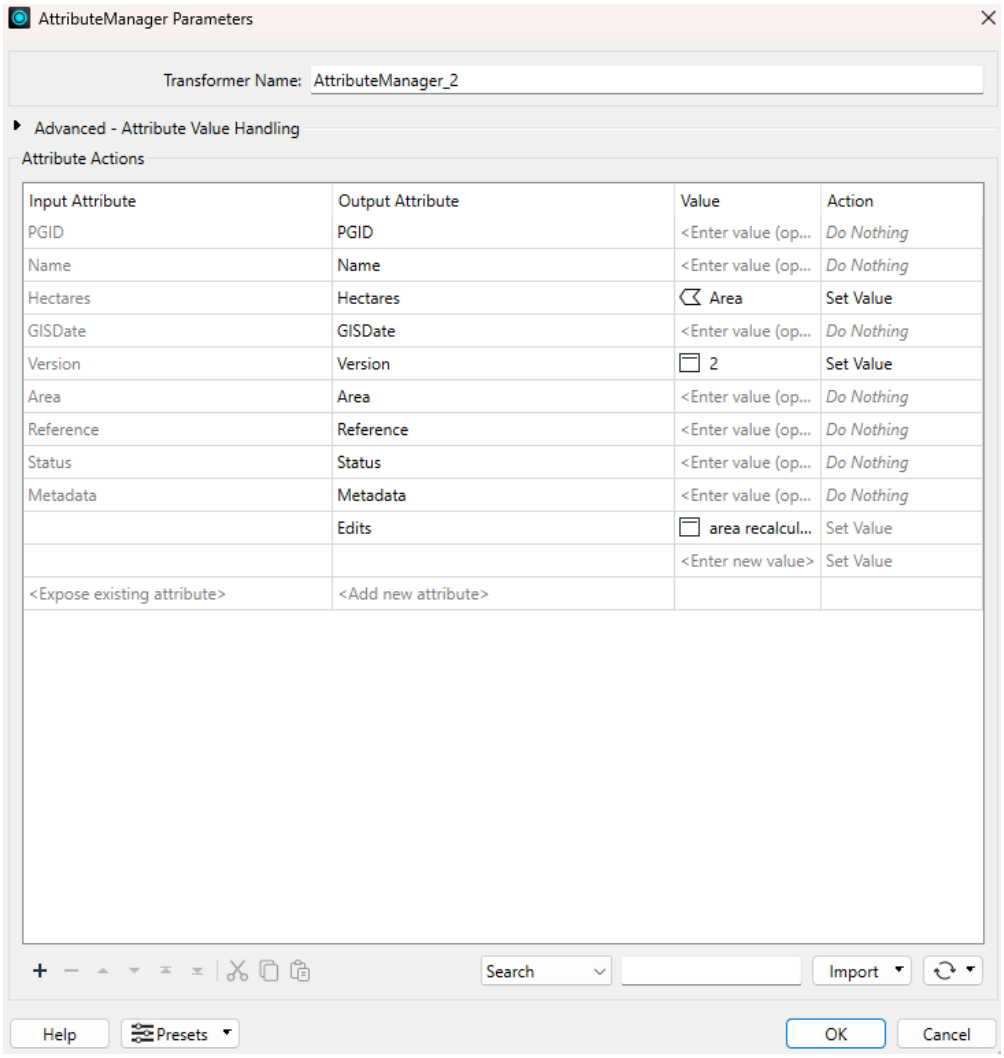
All these tasks can be done at once using a single transformer – an AttributeManager.

Add an AttributeManager transformer and connect it to the Tester Failed port. Within the transformer, make the following changes:

To fix ‘Hectares’ attribute:
Click in the Attribute Value column, of the Hectares row and use drop-down arrow to access the Attribute Value section. There select the ‘Area’ attribute.
The Area value will overwrite the existing Hectares value.

To increase ‘Version’ attribute:
Click in the Attribute Value column, of the Version and enter *2*

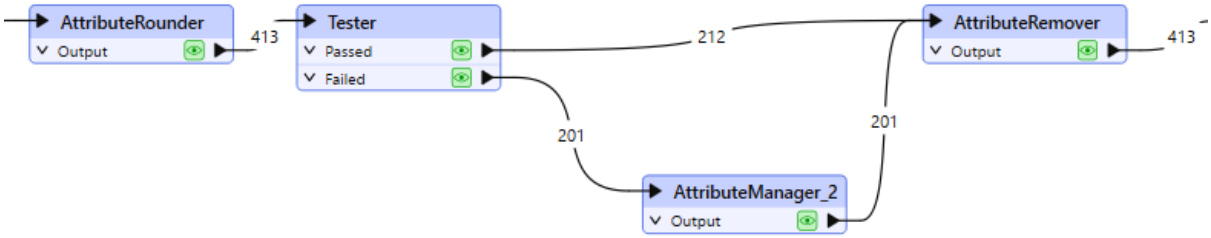
To create the new 'Edits' attribute:
At the bottom of the attributes list click on the Output Attribute column on <Add new attribute> and enter *Edits*
In the Attribute Value cell enter: *area recalculated*



2.5.9 Merge the two data streams and Remove attribute

Now bring the two data streams back together and pass all features through an AttributeRemover, to remove the Area attribute (as we only needed this for validation purposes).

Add an AttributeRemover to the canvas. Then drag a connector line between the AttributeManager and the AttributeRemover. Drag another connector line from the Tester Passed port to the AttributeRemover.



Configure the AttributeRemover to remove the *Area* attribute.

Now let's add a Writer to output the revised data to ESRI Shapefile format.

2.5.10 Add a Writer

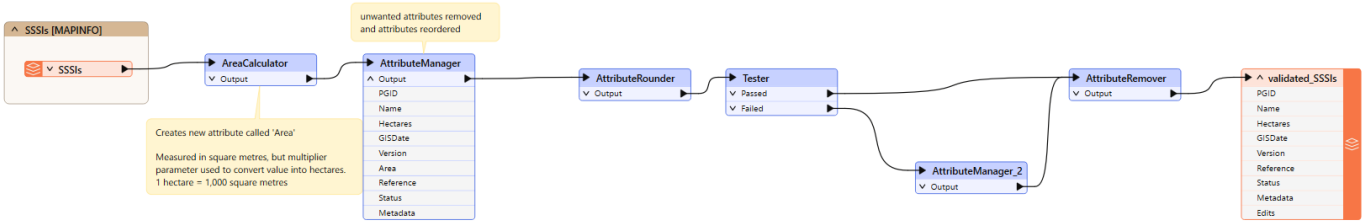
Use either the Writer button, or use Writers > Add Writer… from the menu bar.
The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	Esri Shapefile
Writer Dataset	C:\FMEModularData\Output\Training

Set the Shapefile Definition: to *Automatic*

Click OK. A Feature Type dialog will open. Set the Shapefile Name: to *validated_SSSIs.* Then click OK.

The new Shapefile writer will be added to the Navigator panel, and its feature type will be added to the canvas. Connect it to the AttributeRemover.



2.5.11 Resave and Rerun the Workspace

Click on the Save button, or use File > Save As on the menu bar.
Then run the workspace.

Congratulations

By Completing this exercise you have learned how to:

- Prepare numerical values for direct comparison, using an AttributeRounder
- Using Tests to filter features
- Manage attributes
- Set a fixed (constant) attribute value

2.6 Key-Based Joining Datasets - Voting Analysis

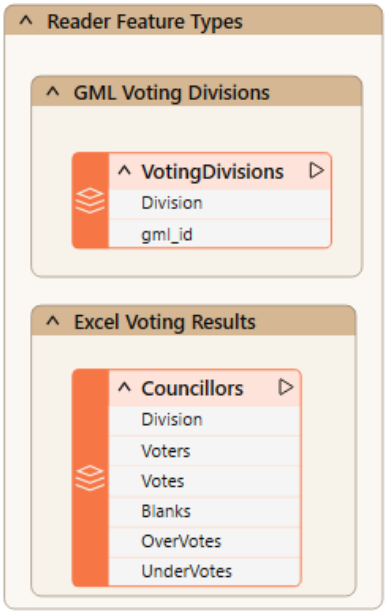
Demonstrates	Performing a Key-Based Join Using the ExpressionEvaluator to perform mathematical calculations and output results as a new attribute
Overall Goal	Generate statistics of voting patterns by performing key-based join and mathematical calculations
Data	ElectionResults (Microsoft Excel) ElectionVoting (OGC GML)
Start Workspace	C:\FMEModularData\Workspaces\2.06-Attributes-Joins-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\2.06-Attributes-Joins-Complete.fmw

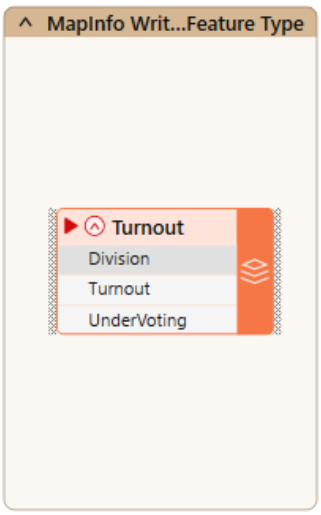
The Electoral Services Officer has asked for help identifying voting divisions that had a low turnout at the last election, or divisions where voters had difficulties understanding the process.
They request that the results be provided in MapInfo TAB format so that they can generate visualizations and reporting.

2.6.1 Launch FME Workbench and Open Existing Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open. Navigate to and open:
C:\FMEModularData\Workspaces\2.06-Attributes-Joins-Begin.fmw

The workspace already has our source datasets (readers) and MapInfo writer configured – included the required attributes on the MapInfo writer feature types.





All we need to do is carry out the transformation, but first, let's inspect the data to know what we are working with:

2.6.2 Examine the input datasets

Use the View Source Data buttons and Visual Preview tools to examine the attributes in both VotingDivisions and Councillors.

Notice that both datasets have a *Division* attribute by which to identify each voting division (area). The Excel data is non-spatial but has a set of other voting attributes:

- Voters: Number of registered voters
- Votes: Number of voters who voted
- Blanks: Number of voters who left a blank or spoiled vote
- OverVotes: Number of voters who voted for too many candidates
- UnderVotes: Number of votes not cast

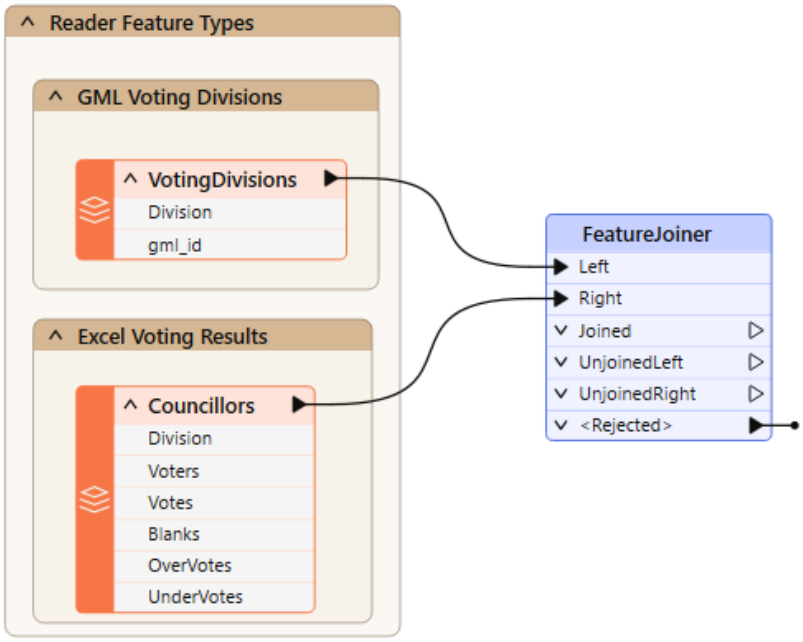
The OverVotes and UnderVotes attributes are an indicator of how well the voting process was understood. Each voter gets to vote for up to 10 candidates (out of 30).
OverVotes are those voters who voted for more than ten candidates. UnderVotes are the number of votes that could have been cast, but were not; for example, the voter only voted for four candidates instead of ten, giving six undervotes.

2.6.3 Add a FeatureJoiner transformer

The first task is to join the statistical election data onto the target features. We'll use a FeatureJoiner transformer to do this. A FeatureJoiner is a way to join or merge features. In this case, we are merging election result records onto election boundary features (Voting Divisions).

Find and add a FeatureJoiner transformer to the canvas. Connect the VotingDivisions

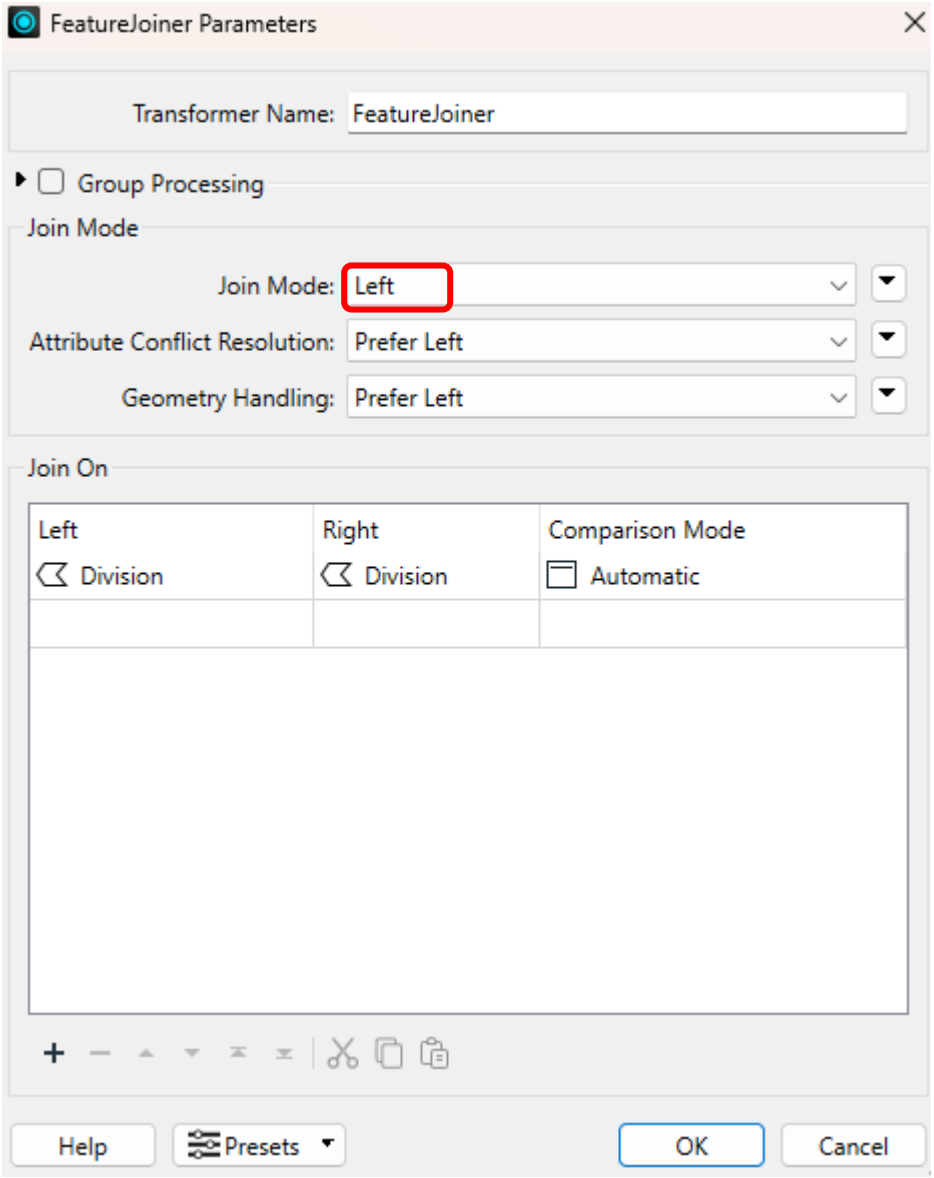
data to the Left port, and the Councillors (result) data to the Right port:



2.6.4 Set the FeatureJoiner Parameters

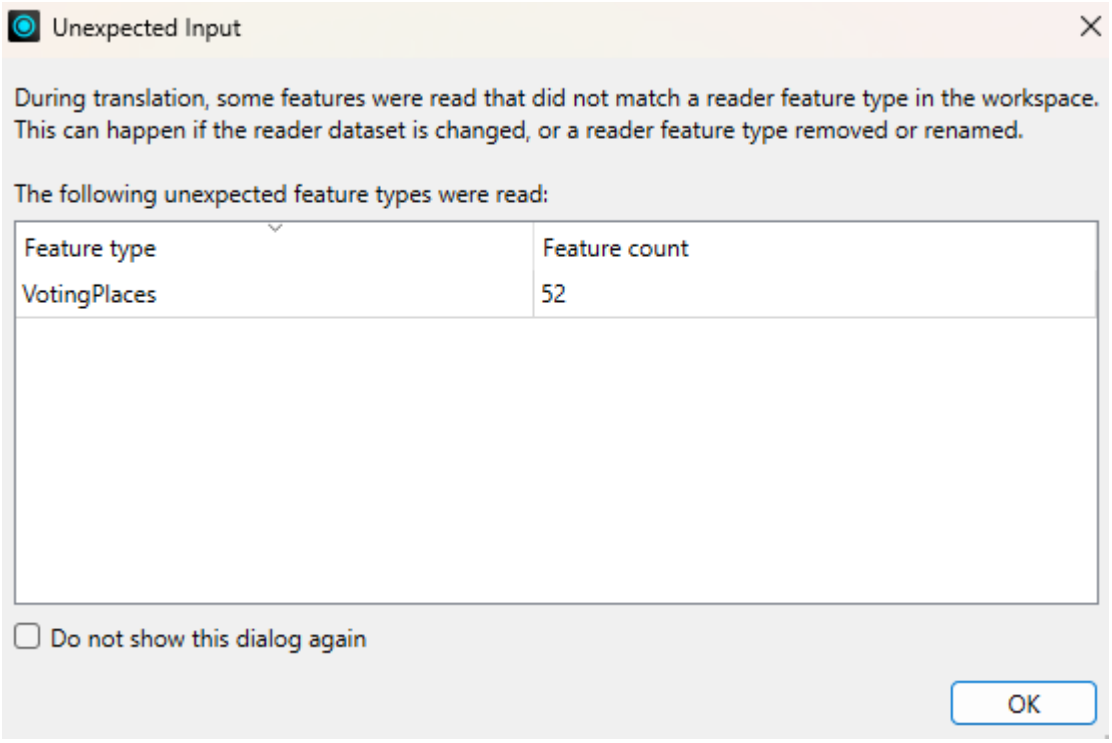
View the FeatureJoiner parameters. Because we want all of the voting division features, we will do a Left join; therefore set the Join Mode to Left.

For both the Left and Right join fields, click in the field and choose the Division attribute from the drop-down list. This attribute is the common key by which we join our data:



2.6.5 Save and Run the Workspace

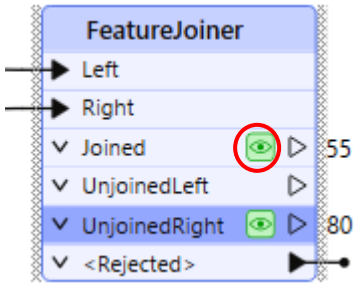
Use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.
Then run the workspace.
There will be an Unexpected Input warning. This is because there is an additional feature type on the GML source data that we are not making use of.



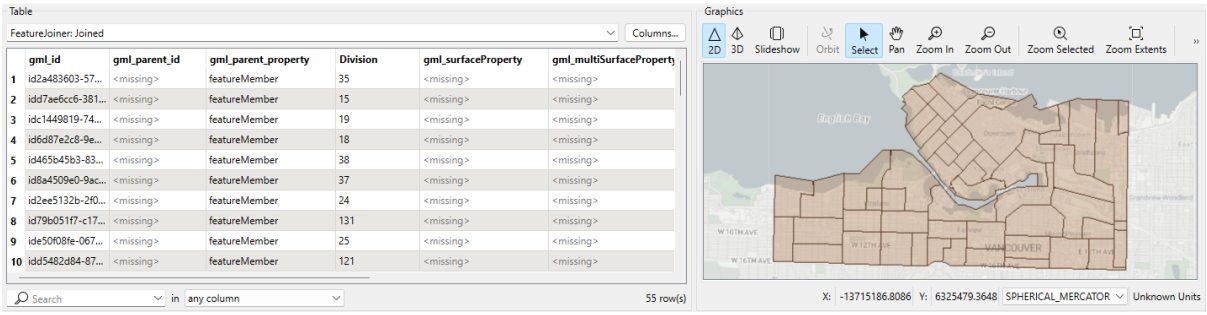
Click OK to ignore this message.

2.6.6 Inspect the FeatureJoiner Feature Cache

Click on the Joined feature cache of the FeatureJoiner to examine the features at this point in the workflow.



Examine the data to ensure all division polygons now include a set of attribute data copied from the Excel spreadsheet:



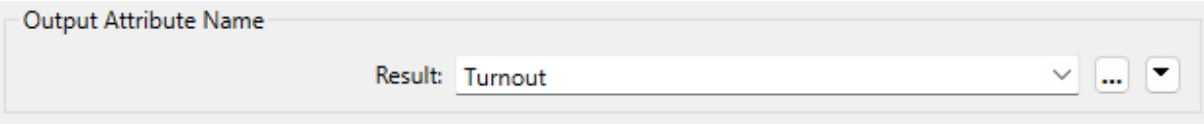
Now that we have the numbers we need, we can start to calculate some statistics.

To do this, we'll use an ExpressionEvaluator transformer first to calculate the voter turnout percentage for each division.

2.6.7 Add an ExpressionEvaluator transformer

Place an ExpressionEvaluator transformer after the FeatureJoiner - connect it to the FeatureJoiner:Joined output port.

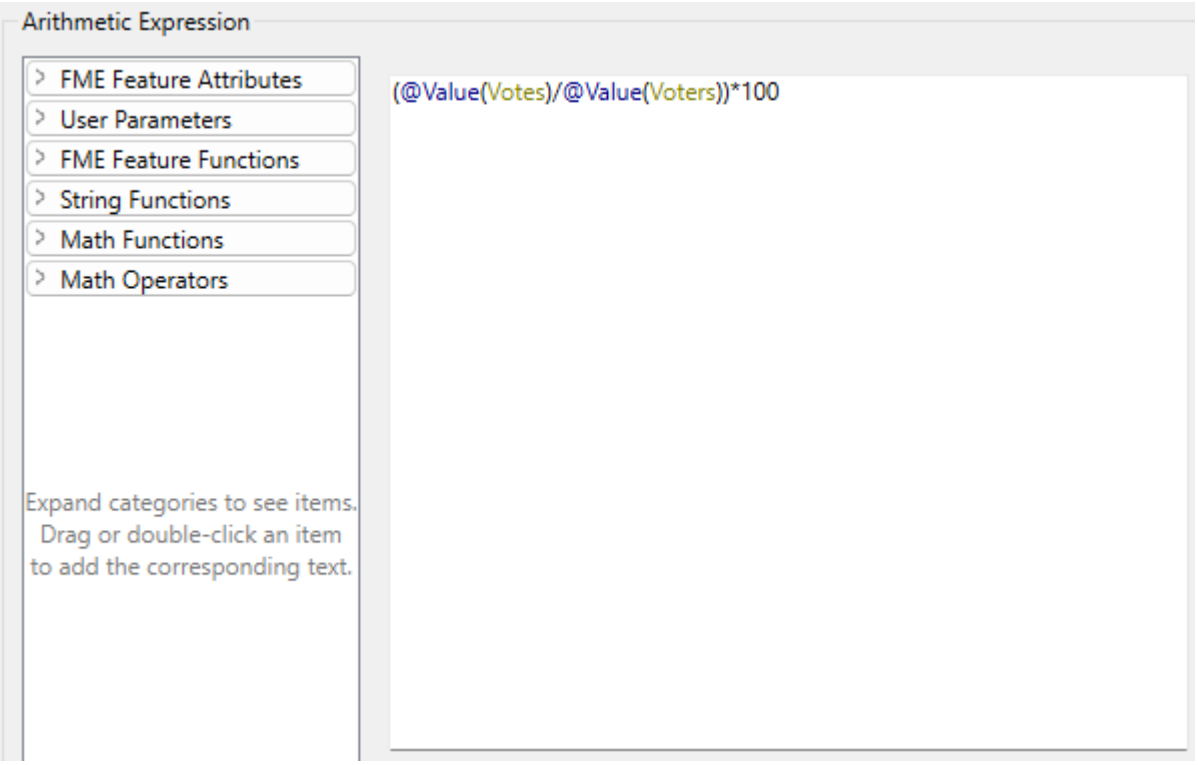
View the transformer's parameters. Set the New Attribute to *Turnout* (to match what we have on the destination schema):



In the Arithmetic Expression section, set the expression to:

```
(@Value(Votes)/@Value(Voters))*100
```

You don't need to type the *@Value(Votes)* and *@Value(Voters)* part in, it can be obtained by double- clicking on the attributes in the list to the left under FME Feature Attributes:



2.6.8 Save and Rerun the Workspace

Click the Save button, then run the workspace.

2.6.9 Inspect the ExpressionEvaluator Feature Cache

Click on the feature cache of the ExpressionEvaluator to examine the features at this

point in the workflow.

Viewing the attributes in a Table View within the Visual Preview window you should now see the new Turnout attribute:

Table

ExpressionEvaluator_3: Output — Columns...

	nl_multiSurfaceProperty	Voters	Votes	Blanks	OverVotes	UnderVotes	Turnout
1	missing>	3195	1129	32	0	1702	35.3364632237...
2	missing>	3997	780	34	0	1160	19.5146359769...
3	missing>	3062	522	35	0	852	17.0476812540...
4	missing>	3880	853	37	1	1463	21.9845360824...
5	missing>	1521	559	12	0	710	36.7521367521...
6	missing>	2703	833	25	0	1378	30.8176100628...
7	missing>	2552	786	20	2	1276	30.7993730407...
8	missing>	2822	974	31	0	1345	34.5145287030...
9	missing>	2987	932	23	0	1480	31.2018747907...
10	missing>	2824	830	25	1	1231	29.3909348441...

2.6.10 Add another ExpressionEvaluator transformer

Using a similar technique, add a second ExpressionEvaluator to calculate the number of UnderVotes per voter and put it in an attribute that matches the output schema which will be *UnderVoting*.
Set the expression to:

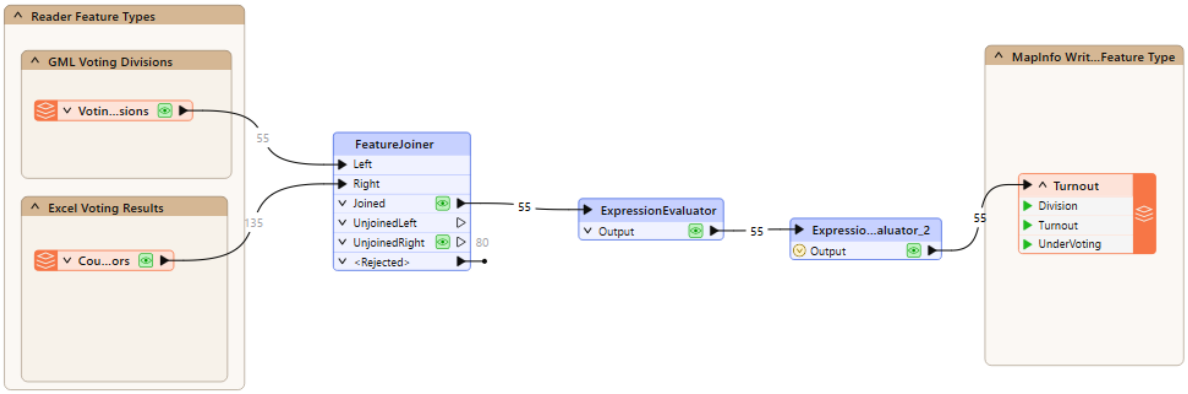
```
@Value(UnderVotes)/@Value(Voters)
```

Note: This isn't a percentage, like the previous calculation.

Now connect the second ExpressionEvaluator to our Turnout feature type (of the MapInfo writer). Notice that the writer schema is only making use of some of the attributes.

2.6.11 Save and Rerun the Workspace

Click the Save button, then run the workspace.



Congratulations

By Completing this exercise you have learned how to:

- Perform a Key-based Join, obtaining attribute values from a supplier (right) dataset
- Use ExpressionEvaluator transformers to perform mathematical calculations and output result as a new attribute
- Use transformer parameters to create attributes that match the writer schema

3 Working with Spatial Data

3.1 Spatial Data Concepts

Demonstrates	Filtering features by geometry type Converting a polygon to a point object Setting coordinate systems on reader/writer Reprojecting data to a different coordinate system Extracting coordinates as attributes Setting the geometry type on a writer feature type
Overall Goal	Translate the Crime Incidents MapInfo TAB data to GeoPackage format, whilst also checking geometry type and reprojecting to the required coordinate system.
Data	Crime Incidents (MapInfo TAB)
Start Workspace	None
End Workspace	C:\FMEModularData\Workspaces\Complete\3.01-Spatial-SpatialDataConcepts-Complete.fmw

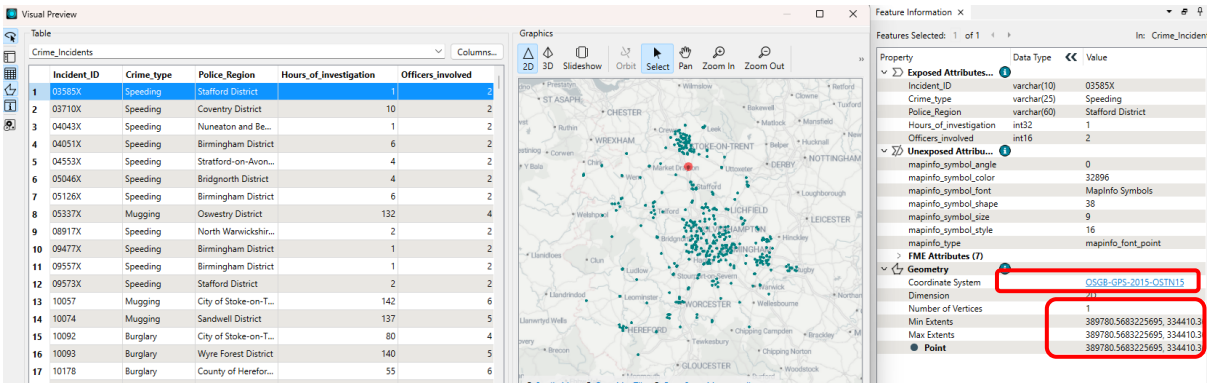
The Community Engagement team has sourced a spatial dataset containing crime incidents and want to load it into one of their systems. However, the data needs to be structured in a certain way before it can be consumed by their system:

- the data has to be in Lat/Long WGS84 coordinate system
- contain only points
- include the coordinates stored as attributes

3.1.1 Use the Data Inspector to inspect the input data

Launch the Data Inspector, then open the crime dataset.
When prompted, fill in the fields inthe Select Dataset dialog as follows:

Reader Format	Precisely MapInfo TAB (MAPINFO)
Reader Dataset	C:\FMEModularData\Data\GB\Crime Incidents.tab



By selecting a feature and viewing the details in Feature Information window, we can see that the coordinate system of the dataset is *OSGB-GPS-2015-OSTN15* (British

National Grid). The Attributes section of the Feature Information window details the attributes and properties of the selected feature. These include *fme_geometry* and *fme_type*; displaying the geometry information for the selected feature.

Many spatial data formats only permit a single geometry type to be contained within a feature type/table/layer. MapInfo TAB is one of the exceptions, and a MapInfo layer can contain a mixture of geometry types. This is an important consideration when translating data from MapInfo TAB to a different spatial format.

On initial inspection the crime incident dataset seems to consist of point objects, but we can't be sure without testing the entire dataset.

We now need to create a workspace to translate the Crime Incidents MapInfo TAB data to GeoPackage format, whilst also checking geometry type and reprojecting to the required coordinate system.

3.1.2 Launch FME Workbench and Create new Workspace from Blank

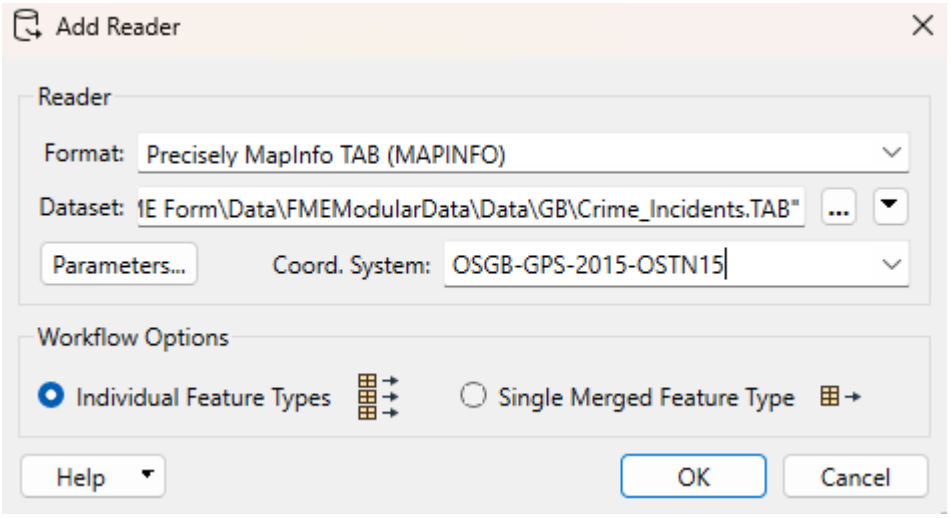
Launch the FME Workbench, if it isn't open already.
Within the Get Started section of the Workbench, click on Blank Workspace.

3.1.3 Add Reader

Use either the Reader button, or use Readers > Add Reader… from the menu bar.
Then define the Format and Dataset settings as follows:

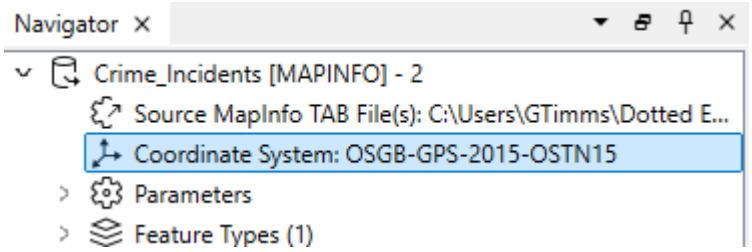
Reader Format	Precisely MapInfo TAB (MAPINFO)
Reader Dataset	C:\FMEModularData\Data\GB\Crime Incidents.tab

Also, specfify the Coordinate System of the dataset: *OSGB-GPS-2015-OSTN15*



Into the Navigator panel a MapInfo Reader will be added and on the canvas there will be a single reader feature type for Crime Incidents.

Notice, the coordinate system parameter on the Reader is populated:



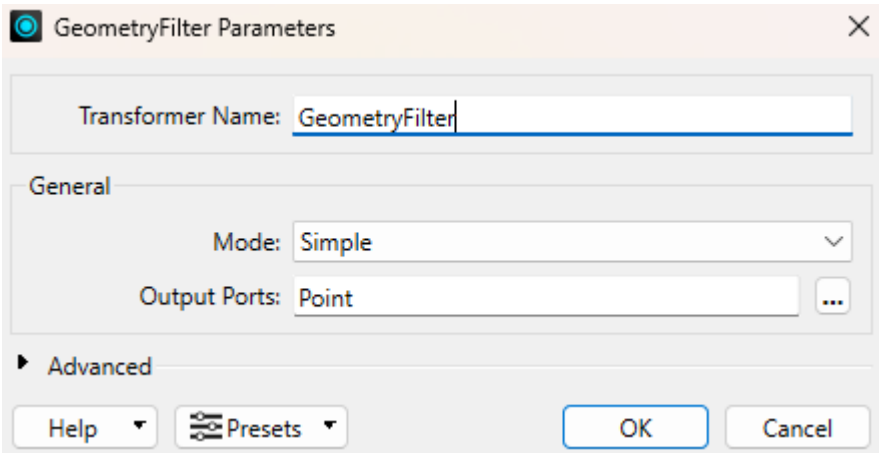
FME Lizard

- *Remember, when a reader's Coordinate System parameter is defined as <not set> FME will automatically try to determine the correct coordinate system from the dataset itself.*
- *When the source dataset is in a format that stores coordinate system information (as it does in this example) you can safely leave the parameter unset.*
  – so the step to populate it in this exercise wasn't necessary, we could have left it <not set>. However it can be useful to users to see the coordinate information, especially in workspaces with multiple inputs with different coordinate systems.
- *You **must** set this parameter when you wish to reproject source data that does not store coordinate system information; otherwise, an error will occur in the translation.*

3.1.4 Add an GeometryFilter transformer

The GeometryFilter lets you select which geometry types to filter incoming features by. It enables you to separate features into different groups based on geometry.

Add and connect a GeometryFilter transformer to the Crime Incidents feature type. Then within the transformer parameters click on the ellipse button for *Output Types*. Then select *Point:*

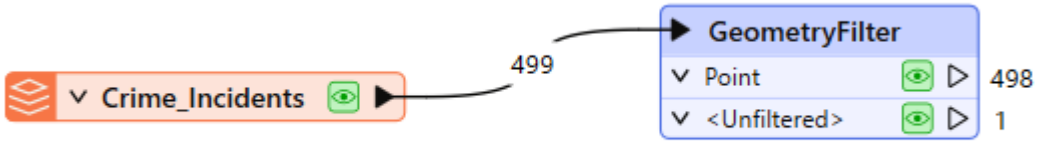


Click OK. An output port for each chosen geometry type is then added to the GeometryFilter. In this example, just one output port; for Point

3.1.5 Save and Run the Workspace

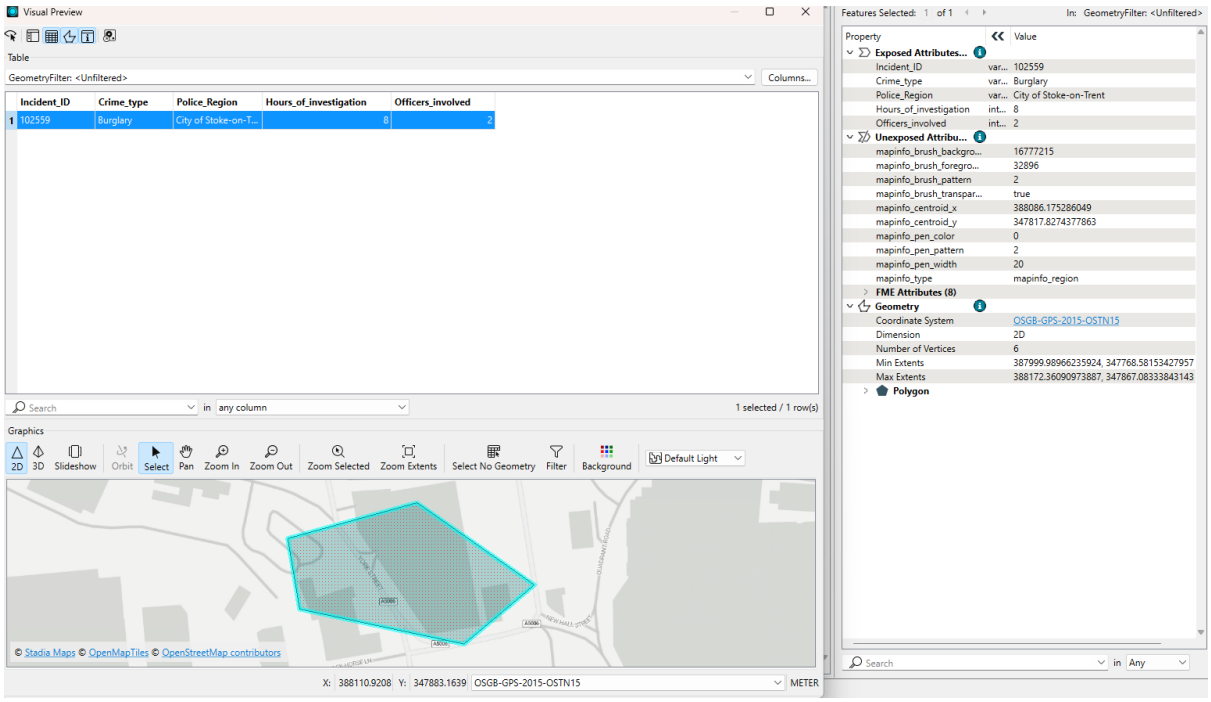
Use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.
Then run the workspace.

The GeometryFilter has separated the 499 crime incidents into two groups; 498 with point geometry and 1 with a different geometry (not point)



3.1.6 Inspect the GeometryFilter Feature Cache

Click on the GeometryFilter *<Unfilitered>* feature cache to examine the feature.



The feature with Incident_ID 102559 has polygon geometry. We will need to replace this with a point object before it can be included in the output!

FME Lizard

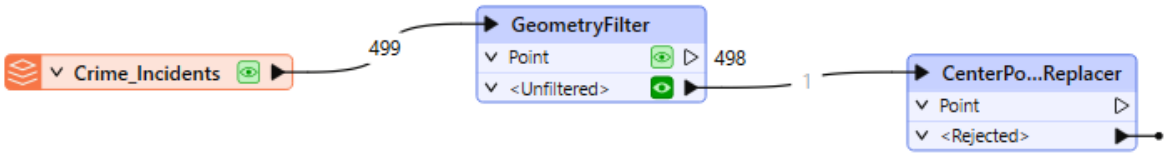
When processing spatial data, it's advisable to first validate the data (e.g., ensuring it contains the expected geometry type). As unexpected or missing geometry may result in spatial-based transformers rejecting features or cause you translation to fail. This is especially important when designing workspaces that will run as scheduled tasks or in a production environment.

Another useful transformer to include before processing or writing spatial data is the GeometryValidator. - especially useful for identifying data quality issues such as polygon self-intersections.

3.1.7 Add a CenterPointReplacer transformer

Add a CenterPointReplacer transformer to the canvas and connect it to the *<Unfiltered>* output port of the GeometryFilter. This transformer replaces the geometry of the feature with a point that is either in the center of the feature's bounding box, at the centre of mass of the feature, or somewhere guaranteed to be inside the feature's area.

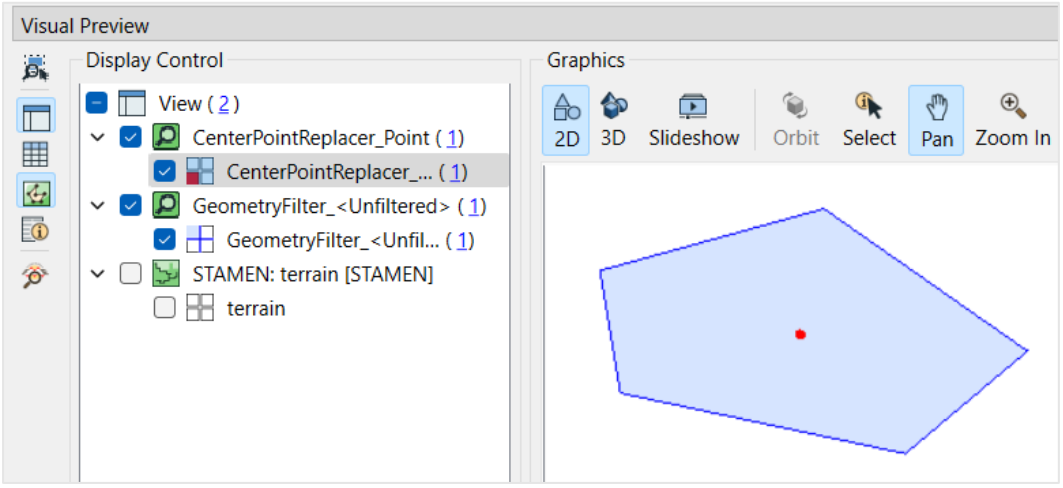
Within the CenterPointReplacer parameters, set the Mode: *to Center of Gravity Point.*



Then run the workspace. (or use *Run From This* on the CenterPointReplacer)

3.1.8 Examine the CenterPointReplacer Feature Cache

Click on the CenterPointReplacer feature cache to examine the feature.
You can also display the original polygon in the same view, by using the Shift key and selecting also the <Unfiltered> cache.



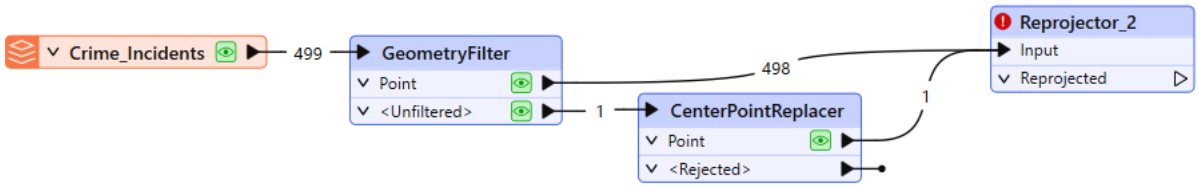
Now that the polygon of Incident_ID 102559 has been replaced by a point, we can add it

back into the main dataset at the next step of the workflow.

3.1.9 Add a Reprojector transformer

The coordinate system of the dataset is British National Grid (OSGB-GPS-2015-OSTN15), but it needs to be in Lat/Long WGS84.

Add a Reprojector transformer to the canvas and connect it to both the GeometryFilter *Point* output port and CenterPointReplacer *Point* output port:

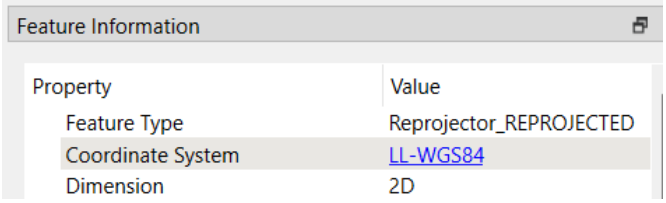


Set the Reprojector transformer parameters to:
Source Coordinate System:
Destination Coordinate System:

Source Coordinate System:	OSGB-GPS-2015-OSTN15	EPSG: 27700
Destination Coordinate System:	LL-WGS84	EPSG: 4326

Then run the workspace. (or use *Run From This* on the Reprojector)

Examine the Reprojector feature cache. Select a feature and check the Coordinate System value within the Feature Information window – it should be LL-WGS84

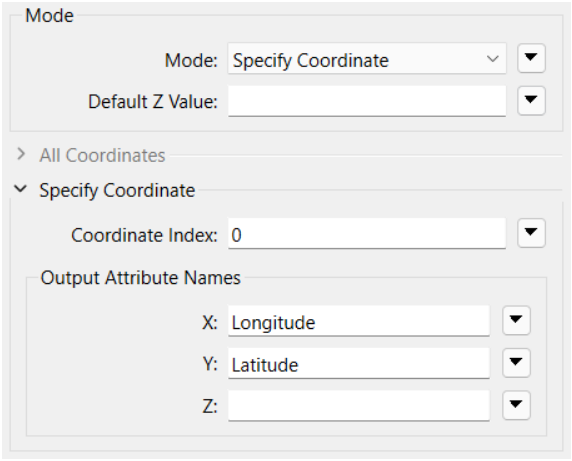


Finally, the last requirement is that the coordinates of each point are extracted and added as attributes to the dataset.

3.1.10 Add a CoordinateExtractor transformer

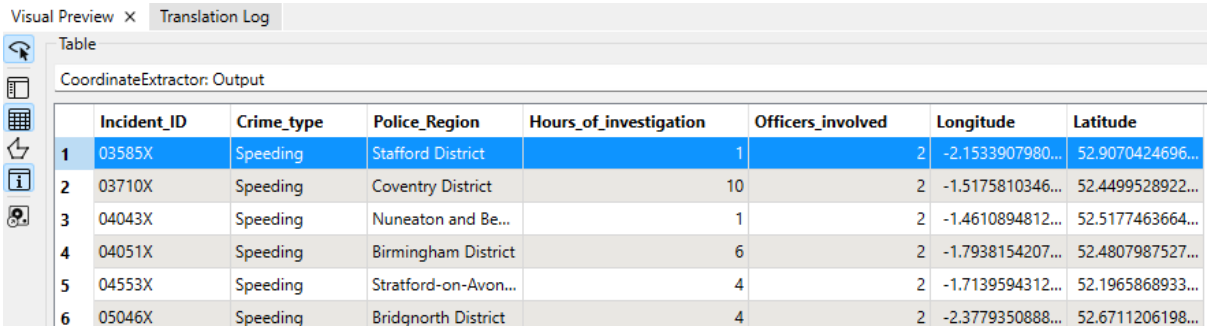
Add a CoordinateExtractor transformer to the canvas and connect it to the Reprojector. This transformer retrieves either specified individual coordinates or all coordinate values from geometry, adding them to the feature as attributes.

Within the transformer parameters set the Mode: to *Specify Coordinate*
Set the *X* coordinate attribute to be named: Longitude
Set the *Y* coordinate attribute to be named: Latitude
We don't need a Z value, so blank that option (delete the *_z*)



Then run the workspace. (or use *Run From This* on the CoordinateExtractor)

Examine the CoordinateExtractor feature cache within a Table View. The new attributes of Longitutde and Latitude have been added, and populated with the coordinates of each point.



	Incident_ID	Crime_type	Police_Region	Hours_of_investigation	Officers_involved	Longitude	Latitude
1	03585X	Speeding	Stafford District	1	2	-2.1533907980...	52.9070424696...
2	03710X	Speeding	Coventry District	10	2	-1.5175810346...	52.4499528922...
3	04043X	Speeding	Nuneaton and Be...	1	2	-1.4610894812...	52.5177463664...
4	04051X	Speeding	Birmingham District	6	2	-1.7938154207...	52.4807987527...
5	04553X	Speeding	Stratford-on-Avon...	4	2	-1.7139594312...	52.1965868933...
6	05046X	Speeding	Bridgnorth District	4	2	-2.3779350888...	52.6711206198...

3.1.11 Add a Writer

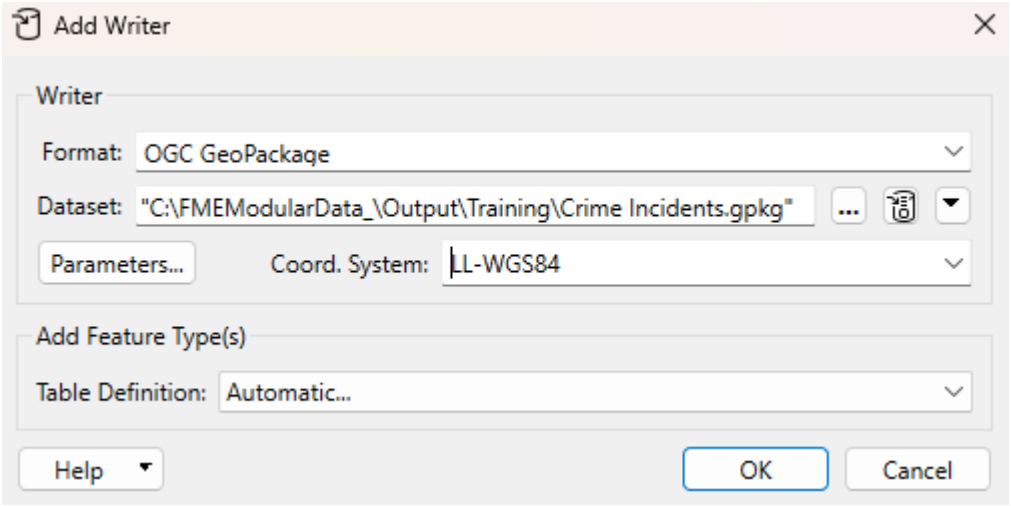
The Community Engagement team want the data suppling in GeoPackage format.

Some destination formats only permit features of a specific geometry type to be written to a single feature type. For example, a Personal Geodatabase Feature Type (Esri Feature Class) can hold polygons or polylines, but not both.
This is the case with GeoPackage format, so as part of adding a writer we will need to define that the specific geometry type to be written is point.

Use either the Add Writer button or use Writers > Add Writer… from the menu bar.
Define the Format and Dataset settings as follows:

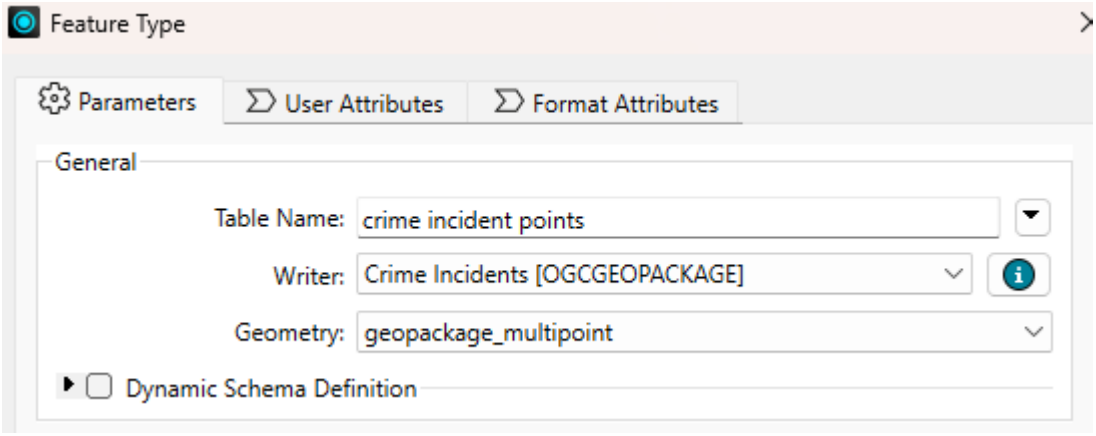
Writer Format	OGC GeoPackage
Writer Dataset	C:\FMEModularData_\Output\Training\Crime Incidents.gpkg

Also, set the Coordinate System to *LL-WGS84*
And set the Table Definition: to *Automatic…*



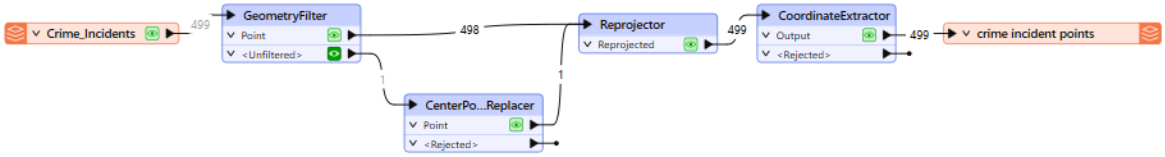
Click OK

Within the Feature Type dialog set the following parameters:
Table Name: *crime incident points*
Geometry: geopackage_multipoint



Click OK.

Then connect the CoordinateExtractor Output to the new *crime incidents points* feature type.



3.1.12 Resave and Rerun the Workspace

Click on the Save button to resave the workspace.
To ensure that we aren't working with any incomplete caches (which can occur when making changes to transformers and the workflow design), rerun the workspace using the

menu command *Run > Rerun Entire Workspace.*

3.1.13 Inspect the Output

Open the newly reprojected crime GeoPackage dataset and select a feature. The Feature Information window should report that the data is now in LL-WGS84.

FME Lizard

Watch out! GeoPackage writers by default append to an existing dataset, they don't over write/replace them. Unlike MapInfo TAB or ESRI Shapefile writers, which over write datasets of the same name.

So if you rerun your workspace multiple times when testing/designing your workspace, be sure to either disable the GeoPackage writer or delete the unwanted GeoPackage files each time. Or change the parameters on the Writer.

Congratulations

By Completing this exercise you have learned how to:

- *Filter features by geometry type*
- *Convert a polygon to a point object*
- *Set the coordinate systems on reader/writer*
- *Reproject data to a different coordinate system*
- *Extract coordinates and add them as attributes*
- *Set a geometry type on a writer feature type*

3.2 Spatial Joins

Demonstrates	Perform a spatial join using the SpatialRelator Create the spatial join result attribute of count/overlap Use transformers to set a symbology/style for output features (KML)
Overall Goal	Create a new version of the local authority boundaries that includes the required fly-tipping incident counts for each local authority
Data	Fly-Tipping (MapInfo TAB), Local Authority Boundaries (GeoPackage)
Start Workspace	C:\FMEModularData\Workspaces\3.02-Spatial-SpatialJoins-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\3.02-Spatial-SpatialJoins-Complete.fmw

You have been given the task of preparing a dataset for the Corporate Strategy and Environmental Services teams. They have requested a breakdown of regional fly-tipping incidents to local authority level.

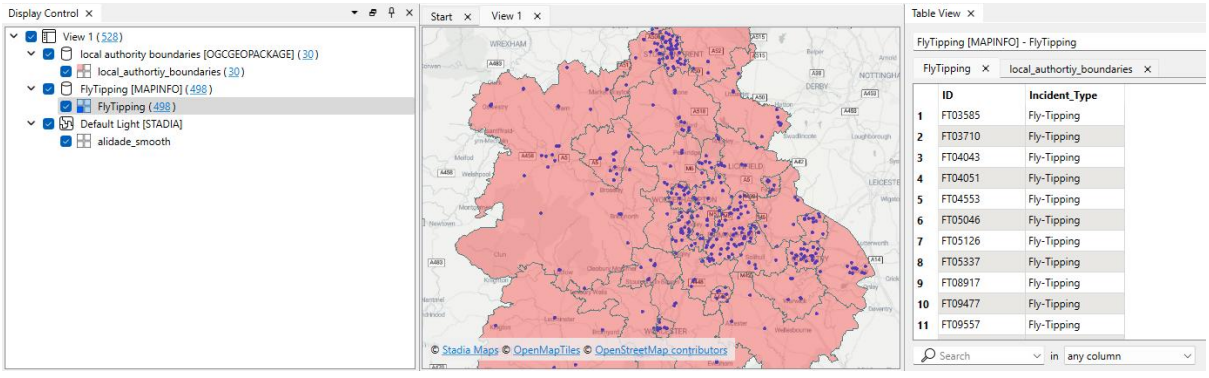
The task is to join two spatial datasets and generate a count of how many fly-tipping incidents occurred within each local authority area.

We will supply the resulting dataset as styled KML for use in Google Earth.

3.2.1 Inspect the Source Datasets

First, familiarize yourself with the data. To do this, open the following datasets within the FME Data Inspector. (Open the first dataset, then use the *Add* button to bring in the second dataset):

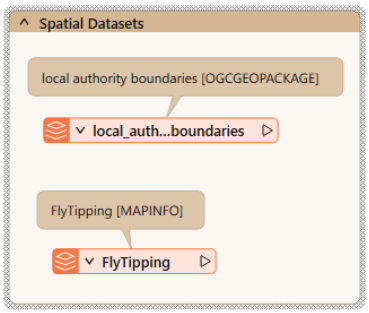
Reader Format	Precisely MapInfo TAB (MAPINFO)
Reader Dataset	C:\FMEModularData\Data\GB\FlyTipping.tab
Reader Format	OGC GeoPackage
Reader Dataset	C:\FMEModularData\Data\GB\local authority boundaries.gpkg

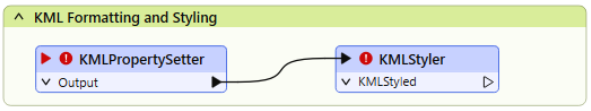


3.2.2 Launch FME Workbench and Open Workspace

A workspace already exists with the first part of our required workflow already setup.

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open Workspace. Navigate to and open C:\FMEModularData\Workspaces\3.02-Spatial-SpatialJoins-Begin.fmw





3.2.3 Add a SpatialRelator transformer

To perform the spatial join and transfer attributes from one feature to another based on their spatial relationship, add the SpatialRelator transformer onto the canvas.

FME Lizard

We could use a PointOnAreaOverlayer transformer instead of the SpatialRelator. But this gives us chance to practice our spatial test usage. We'll take a look at using a PointOnAreaOverlayer transformer in a later exercise.

An advanced use of these transformers could be to also return all the fly-tipping IDs of all the incidents within each local authority. This would be achieved by generating a List attribute, which could then be concatenated into a single new attribute on the local authority features.

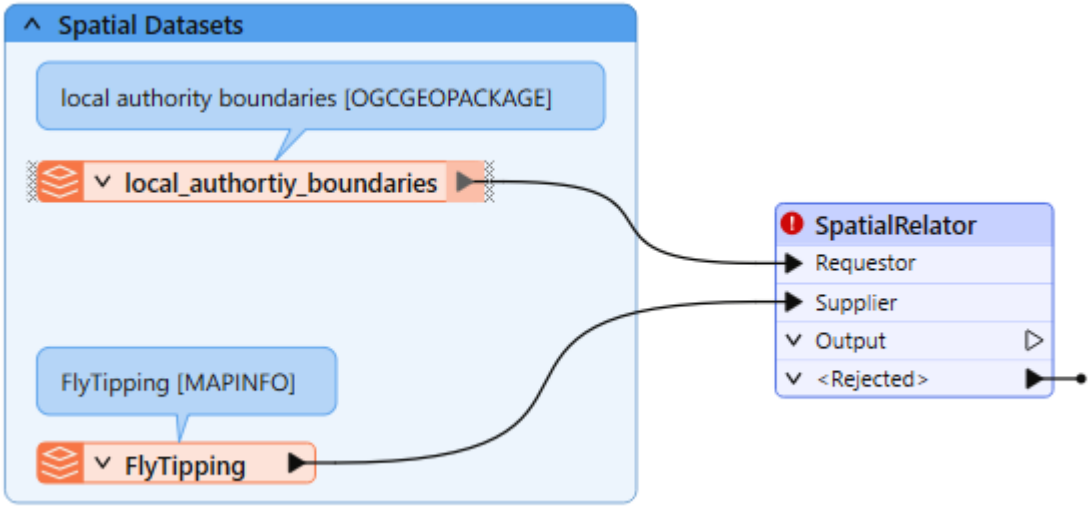
Local_Authorities_Output

	name	flytipping_incident_count	FlyTipping_IDs
14	Telford and Wrekin...	14	FT10196, FT10394, FT119901, FT11993, FT12203, FT12254, FT146618, FT2
15	Newcastle-under-L...	19	FT102159, FT102410, FT102487, FT102556, FT102589, FT102782, FT1028(
16	Wyre Forest District	8	FT10093, FT11671, FT11842, FT11843, FT11855, FT119129, FT142, FT923(
17	Warwick District	7	FT12107, FT121193, FT220, FT44263, FT45391, FT53041, FT53096
18	Bromsgrove District	5	FT10907, FT118555, FT128075, FT129294, FT52456
19	Lichfield District	8	FT10581, FT10886, FT119592, FT122561, FT128042, FT128177, FT40495, F

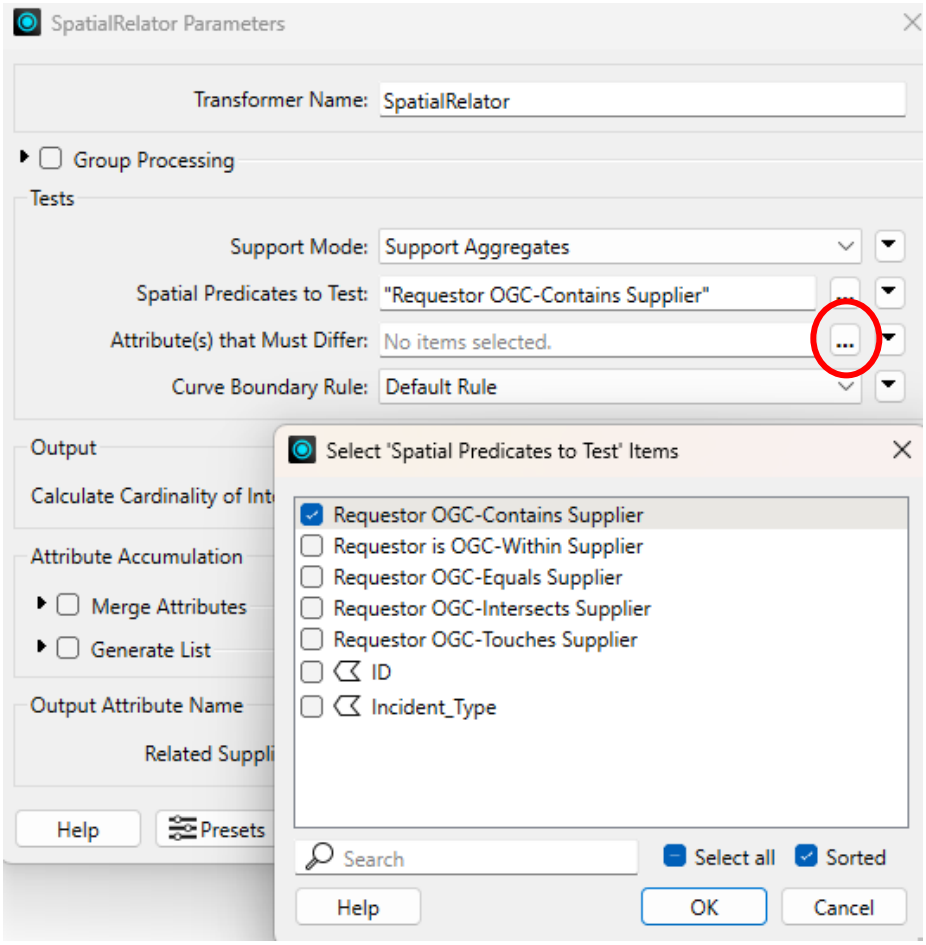
We cover List attributes and the techniques required to achieve the above on our 'Advanced Attribute Handling and Lists' training module.

The SpatialRelator has two input ports; Requester and Supplier. After performing the spatial join the local authority boundaries are our features of interest, so these are our 'requestor' features. Connect the FlyTipping feature type to the *Supplier* input port and

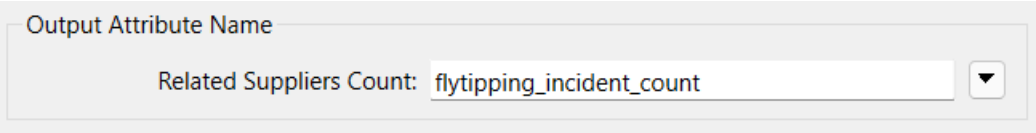
connect local_authorty_boundaries to the *Requestor* input port:



Set the parameters of the transformer as follows:
Spatial Predicates to Test:
Click on the ellipse button to access the spatial tests to perform. In this scenario our requestor dataset is polygons and the supplier dataset are points. So, the appropriate spatial test for this scenario is *Requestor is OGC-Contains Supplier*



Related Suppliers Count:
The default result attribute name would be *_related_suppliers,* lets change this to *flytipping_incident_count*



3.2.4 Save and Run workspace

Click on the Save button, or use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name, and Click Save.

Then run the workspace.

3.2.5 Inspect the SpatialRelator Feature Cache

Click on the SpatialRelator transformer feature cache to examine the features in a Table View.

Table

SpatialRelator_Output

	la_census_code	name	area_description	type	hectares	population_all_ages	fid	flytipping_incident_count
1	E08000025	Birmingham Dis...	Metropolitan District	Metrop...	26779.106	1141816	1	75
2	E08000028	Sandwell Distric...	Metropolitan District	Metrop...	8555.893	328450	2	22
3	E08000031	City of Wolverh...	Metropolitan District	Metrop...	6943.671	263357	3	19
4	E08000026	Coventry Distric...	Metropolitan District	Metrop...	9863.906	371521	4	28
5	E08000027	Dudley District ...	Metropolitan District	Metrop...	9795.824	321596	5	32

The spatial join has been performed and the new count attribute has been populated with the number of fly-tipping points found within each local authority polygon.

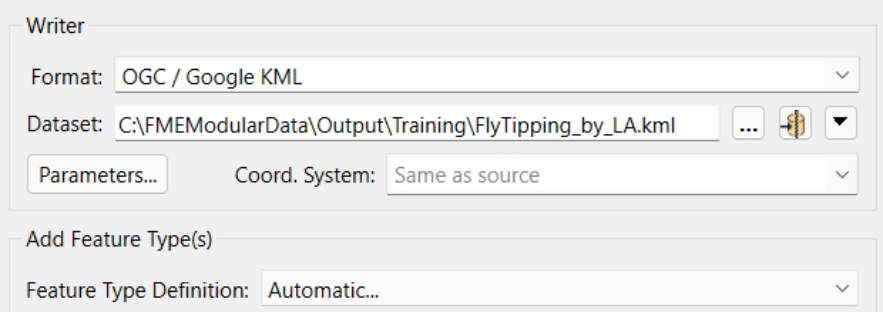
3.2.6 Add a KML Writer

We will provide the output in KML format, so that it can be easily loaded into Google Earth by the end-users.

Use either the Writer button, or use Writers > Add Writer… from the menu bar.
The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	OGC / Google KML
Writer Dataset	C:\FMEModularData\Output\Training

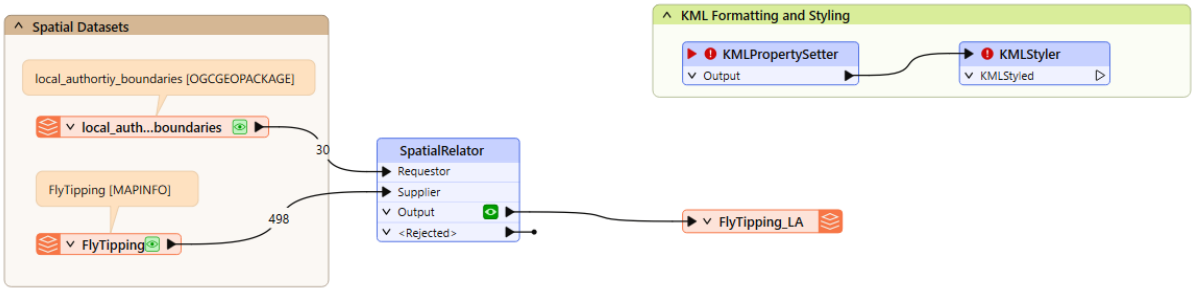
Set the KML file name: *FlyTipping_by_LA.kml*
Set the Feature Type Definition: to *Automatic*



Click OK. A Feature Type dialog will open. Set the Feature Type Name: to *FlyTipping_LA.* Then click OK.

The new KML writer will be added to the Navigator panel, and its feature type will be added to the canvas. Connect the *FlyTipping_LA feature type* to the SpatialRelator.

Note: we aren't yet using the KML formatting and styling transformers, we want to first see what the default output looks like:



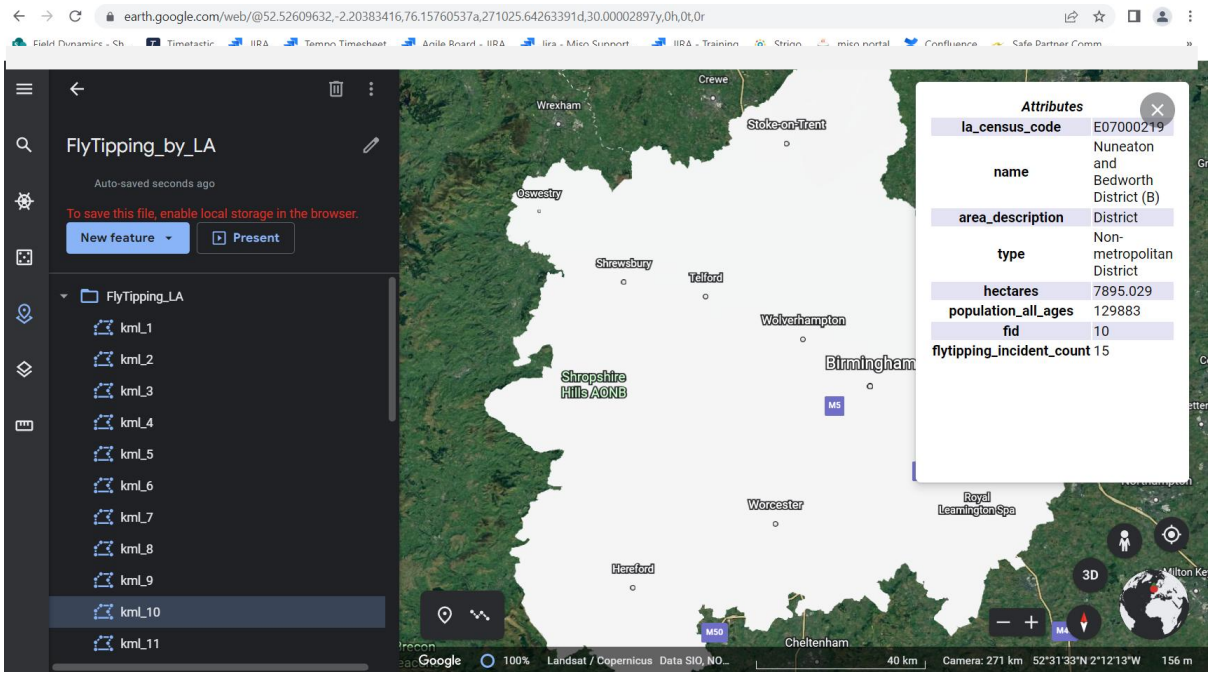
3.2.7 Resave and Rerun the Workspace

Click on the Save button to resave the workspace, then run the workspace.

The workspace will run to completion and output a KML file. We will now examine the output in Google Earth to prove it has the correct attributes and is in the correct location.

3.2.8 Examine the KML dataset in Google Earth

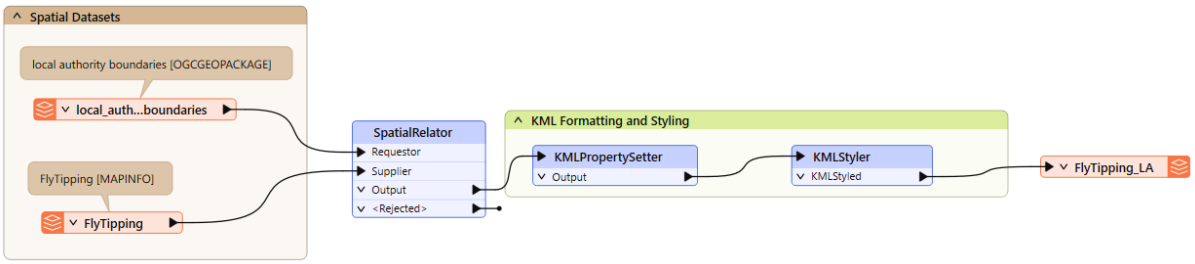
Use either the Google Earth Pro shortcut on the desktop (if there), or open Google Chrome and go to https://earth.google.co.uk/
Within Google Earth, load your KML file (normally via the Open/Import option within Google Earth > Projects)



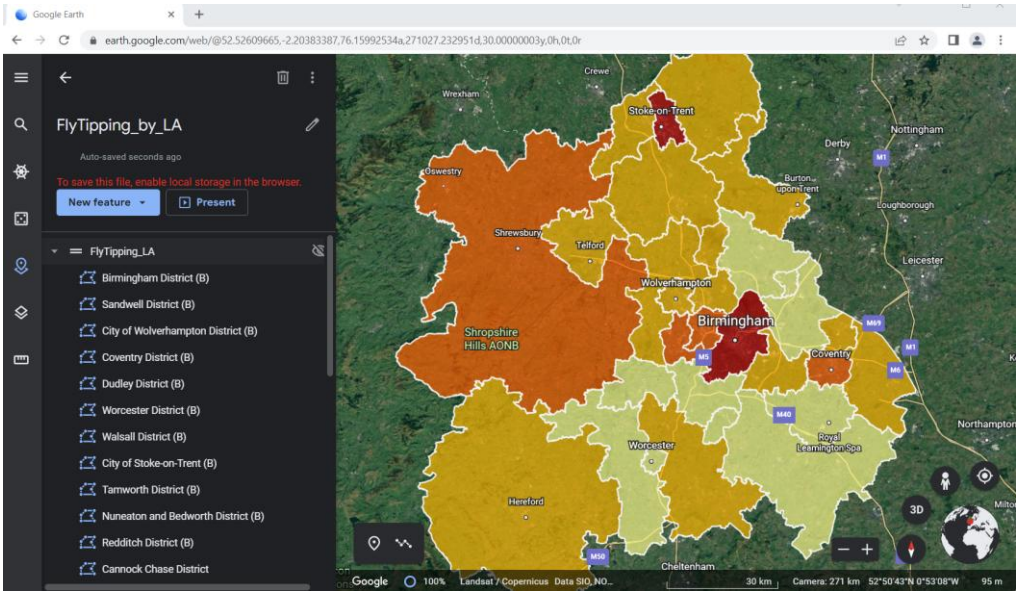
The output is very plain. It would be much better to improve the look of the results and there are several ways to do this with KML. We could colour the local authorities differently according to their fly-tipping incident counts.

The focus of this exercise is spatial Joins, so to save time the workspace already contains KML styling and formatting transfer pre-configured ready for us to use. But if you're interested in their parameters, take a look within each transformer.

Connect the KMLPropertySetter and KMLStyler between the SpatialRelator transformer and KML writer feature type:



Rerun the workspace. Then load the new (and formatted) KML file into Google Earth.



Congratulations

By Completing this exercise you have learned how to:

- Perform a spatial join using the SpatialRelator
- Create the spatial join result attribute of count/overlap
- Use transformers to set a symbology/style for output features (KML)

3.3 Analyse Spatial Data

Demonstrates	Generation of Buffers on spatial data Use of the Clipper transformer to filter spatial data Performing a spatial join using an Overlayer transformer Performing a point to raster overlay to extract values from a DEM Use the Group By parameter in and Inspector to view results of spatial analysis
Overall Goal	Use spatial analysis to identify addresses within the city that are at risk of flooding
Data	Addresses (ESRI Geodatabase), Land Boundary (ESRI Shapefile), Coastline Vancouver (ESRI Shapefile) Canadian Digital Elevation Data (CDED)
Start Workspace	C:\FMEModularData\Workspaces\3.03-Spatial-AnalyzeSpatialData-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\3.03-Spatial-AnalyzeSpatialData-Complete.fmw

The council has asked you to identify addresses within the city that are at risk of flooding.

For this exercise you will model risk as a combination of closeness to the shoreline and elevation above sea level. Flood risk is on a scale from one to five (1-5), and you calculate it using this table:

		Elevation (metres above sea level)		
		1-10m	10-25m	25-60m
Distance from Shoreline (metres)	100m	1	2	3
	200m	2	3	4
	300m	3	4	5

First, you have spatial data of elevation (raster DEM), address points, land boundary polygon, and the coastline. You need to use spatial analysis techniques on your starting datasets to calculate the flood risk. You have several steps in mind:

- Use a **buffer** to create three zones measuring the distance to the coastline: 0-100 m, 101-200 m, and 201-300 m.
- Use **spatial joining and overlaying** to add an elevation and distance to the shoreline attribute to each address.
- Use **attributes** to calculate the final flood risk.

We will show you two methods of doing this, one is more efficient but higher skill requirement than the other.

Let's get started!

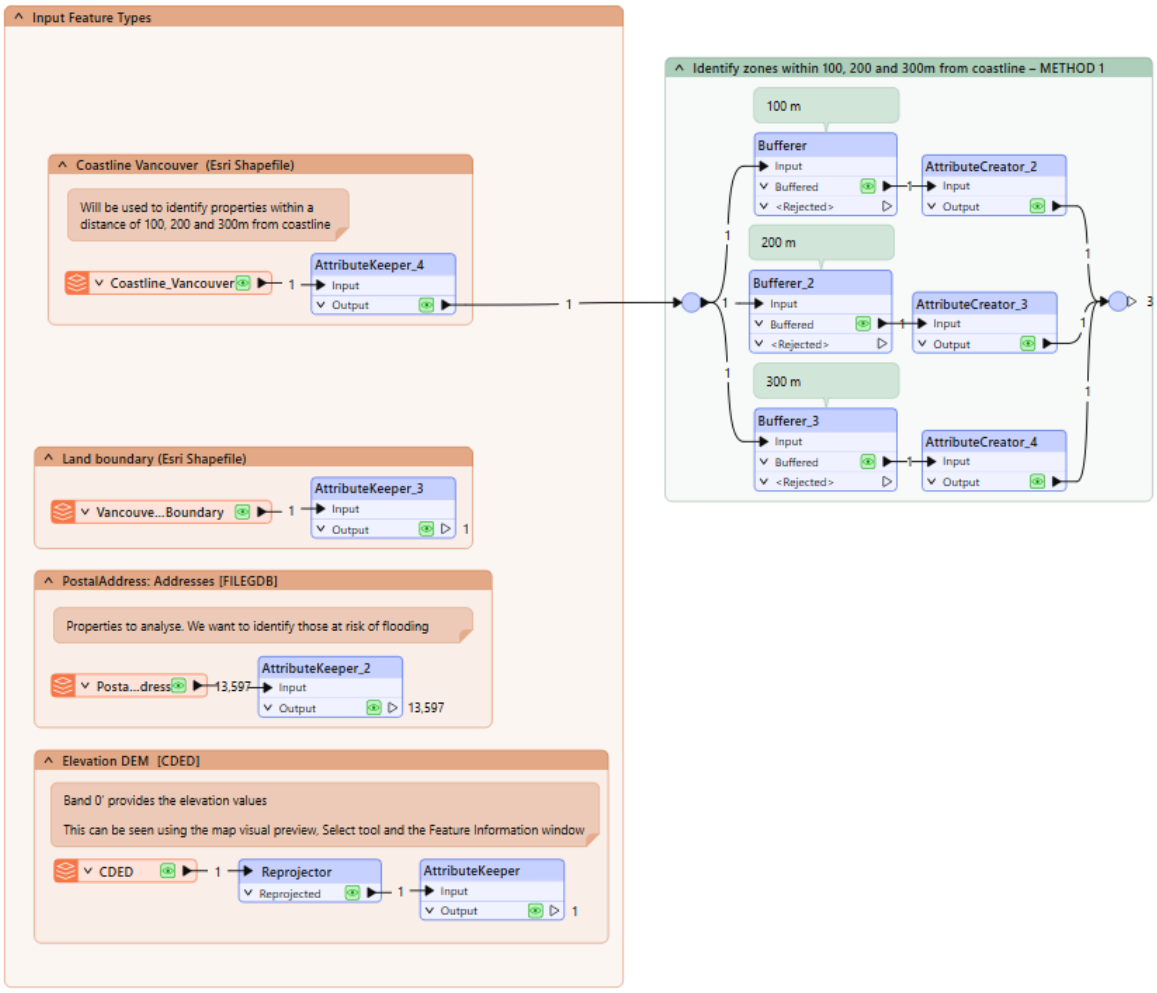
3.3.1 Launch FME Workbench and Open Existing Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open. Navigate to and open:
C:\FMEModularData\Workspaces\3.03-Spatial-AnalyzeSpatialData-Begin.fmw

This workspace contains the datasets you will be working with: a Digital Elevation Model (DEM) file for the elevation, an Esri ArcGIS Geodatabase for the addresses, an Esri shapefile of the land boundary, and an ESRI Shapefile for the coastline. Note that the DEM is an example of raster data, while the other datasets are vector data.

We have AttributeKeepers connected to the inputs, which lets us choose which attributes to keep and which ones to drop. We only want to keep the attributes that we need to work with to perform the analysis.

Turn on feature caching and run the workspace, this is what it should look like:



Inspect the results of each dataset to get familiar with your data.
You should notice that each separate input dataset contains a piece of information we need for our final result.

We'll use spatial filtering and joining to combine them into the resulting dataset that we need.

3.3.2 Create Buffer polygons and Distance Attribute

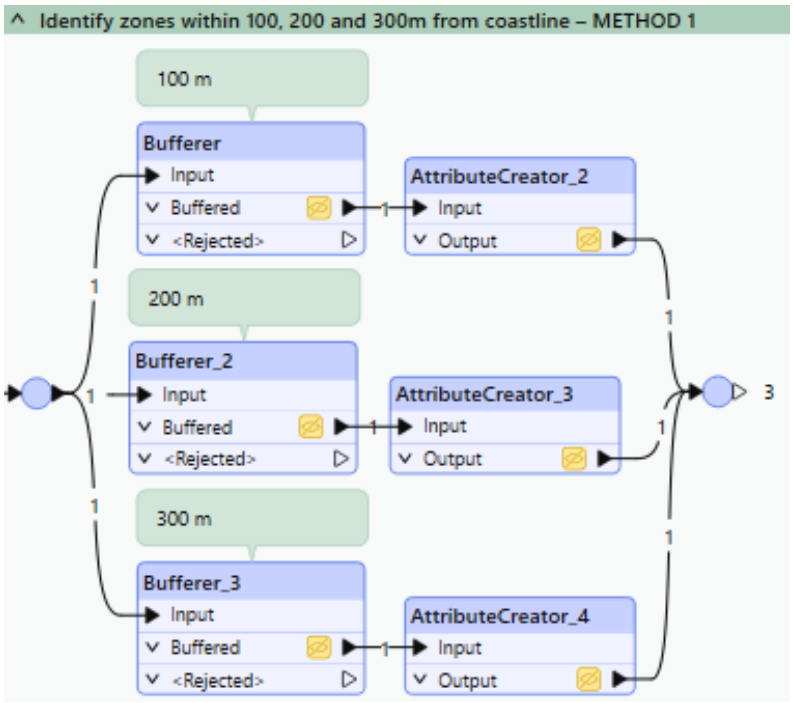
During our analysis we need to identify properties that are within a distance of 100, 200 or 300 metres of the coastline. We will achieve this using buffer polygons generated from the coastline, each with a Distance attribute identifying if the zone is representing 100, 200 or 300m from the coastline.

We want to create three buffer zones, one for each distance band (100, 200, and 300).

The workspace already includes 2 methods for generating the buffer zone polygons that we require. We will be using the buffer polygons from Method 1. We'll later explore the Method 2.

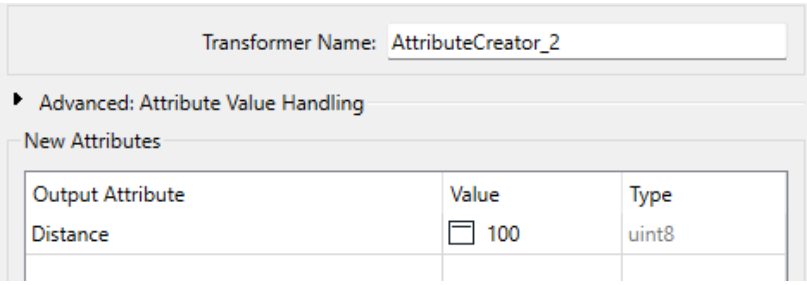
Method 1

You will see in the green bookmark that there is a set of Bufferers created for you with AttributeCreators linked to them.

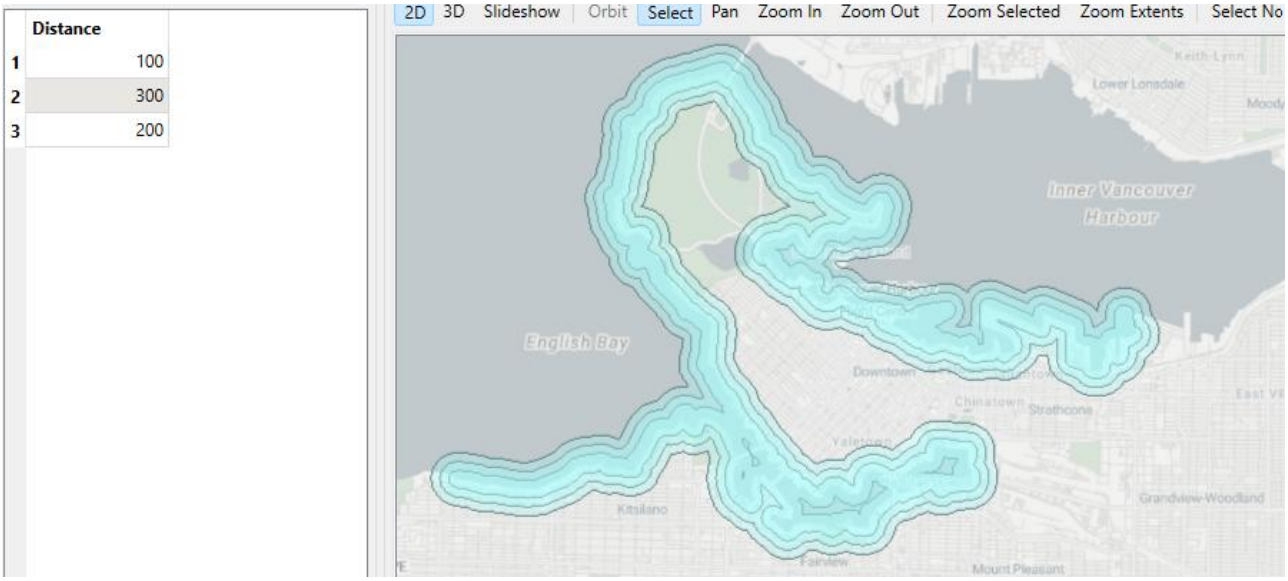


These transformers have been set up to create three buffer polygons around the coastline. Each one a different size: 100, 200 and 300m.

The AttributeCreators that are linked to each Bufferer transformer are creating a Distance attribute which is set to the corresponding distance for that Bufferer.



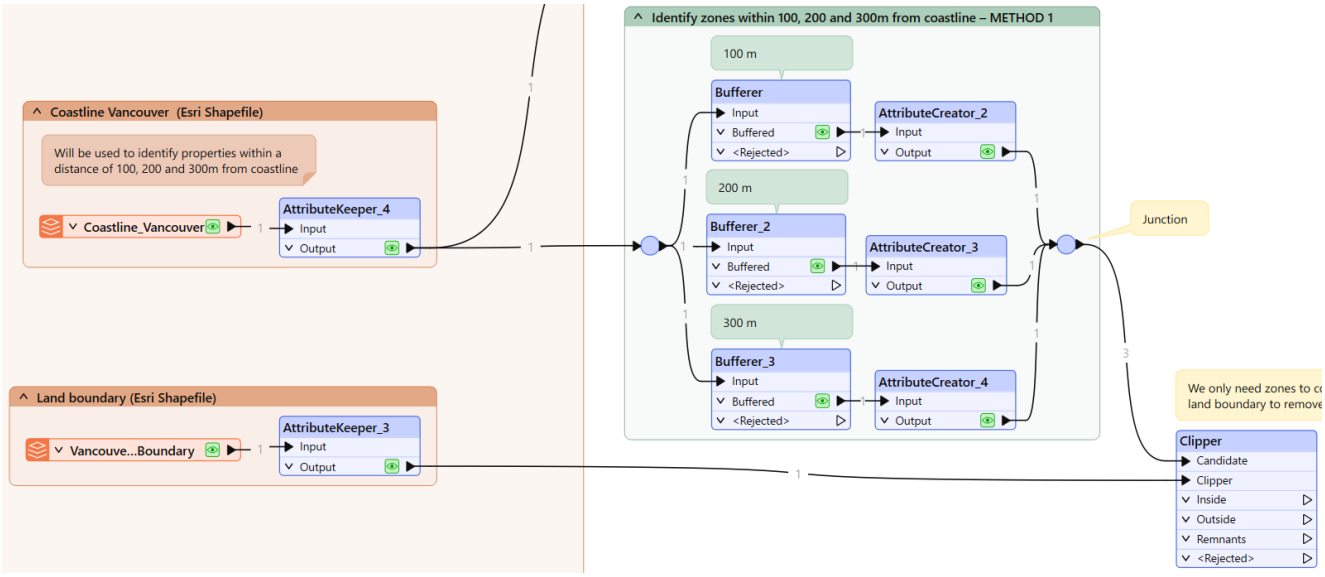
This should create three buffer polygons showing 100, 200 and 300m from the coastline. And include a ‘Distance’ attribute.



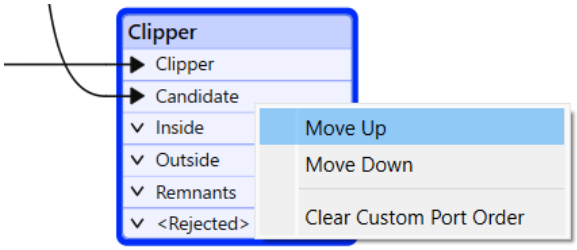
You now need to continue building the workspace. The next step is to tidy these buffer polygons, by clipping them to the land boundary.

3.3.3 Add a Clipper transformer

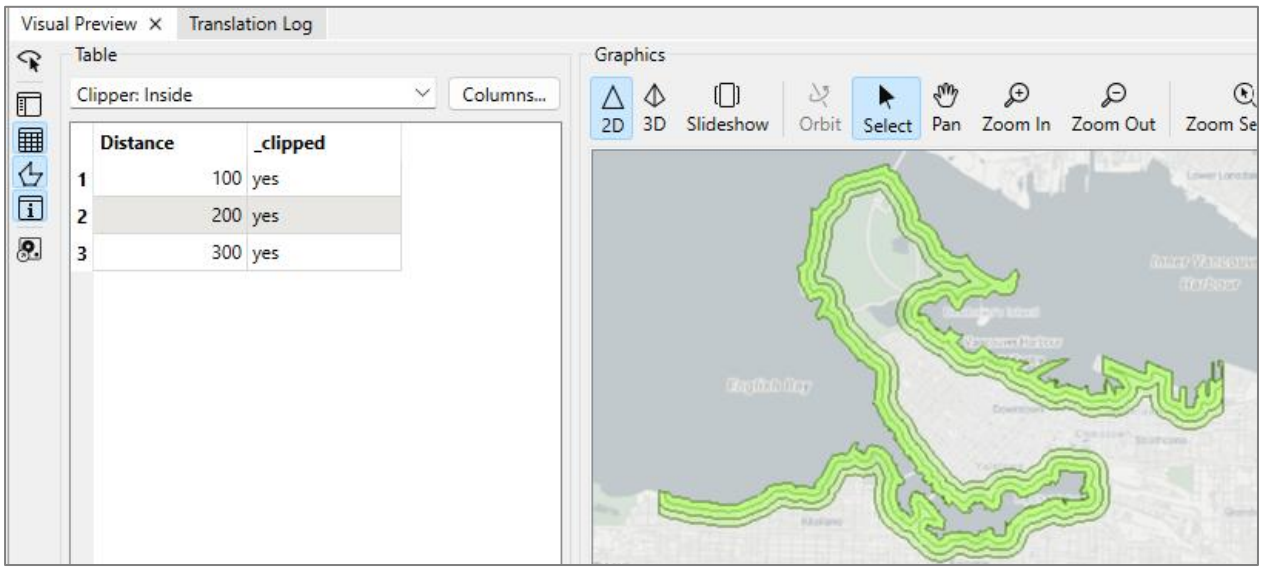
Add a Clipper after the Bufferer section.
Connect the VancouverLandBoundary feature type to the ‘Clipper’ port and the Buffer polygons (from the Junction) to the ‘Candidate’ port.



Watch-out for criss-crossing connector lines. Tidy them up by right-clicking on ‘Candidate’ and choose ‘Move Up’



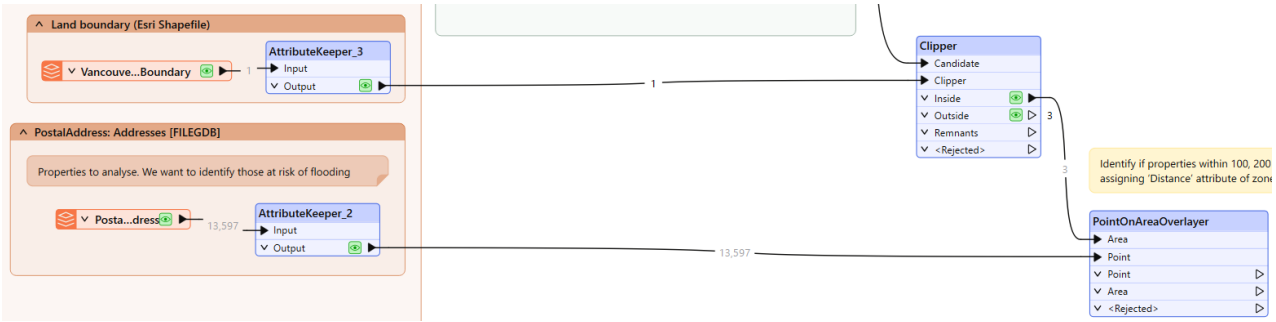
Run the Clipper and inspect the Clipper's *Inside* feature cache. The buffer zones from this port only cover the land area:



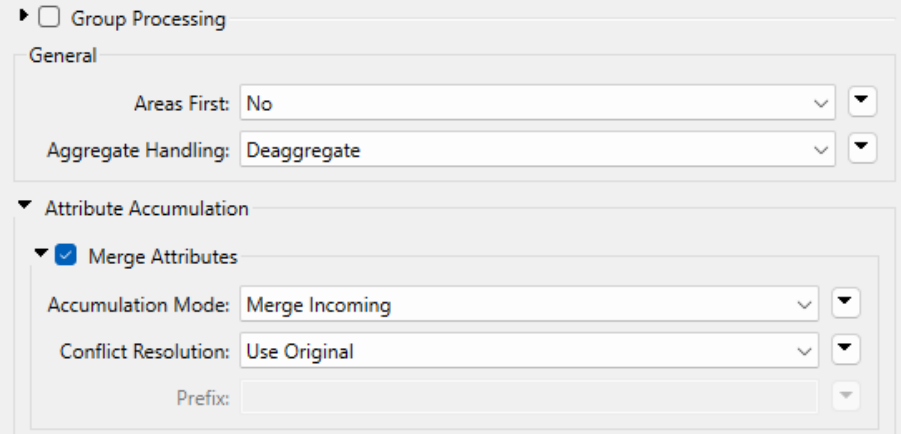
Now we are ready to overlay the address (properties) points on the distance buffer zones. We will add the Distance attribute to our points that records which coastal distance band they overlap.

3.3.4 Add a PointOnAreaOverlayer transformer

Add a PointOnAreaOverlayer and to the 'Area' port connect the Clipper 'Inside'. Then connect the Postal Addresses feature type to the 'Point' PointOnAreaOverlayer port.



Now double-click the PointOnAreaOverlayer to edit its parameters.
Expand the *Attribute Accumulation* section and enable the *Merge Attributes* parameter.



With this parameter enabled, the PointOnAreaOverlayer adds attributes from the areas to the points and vice versa.

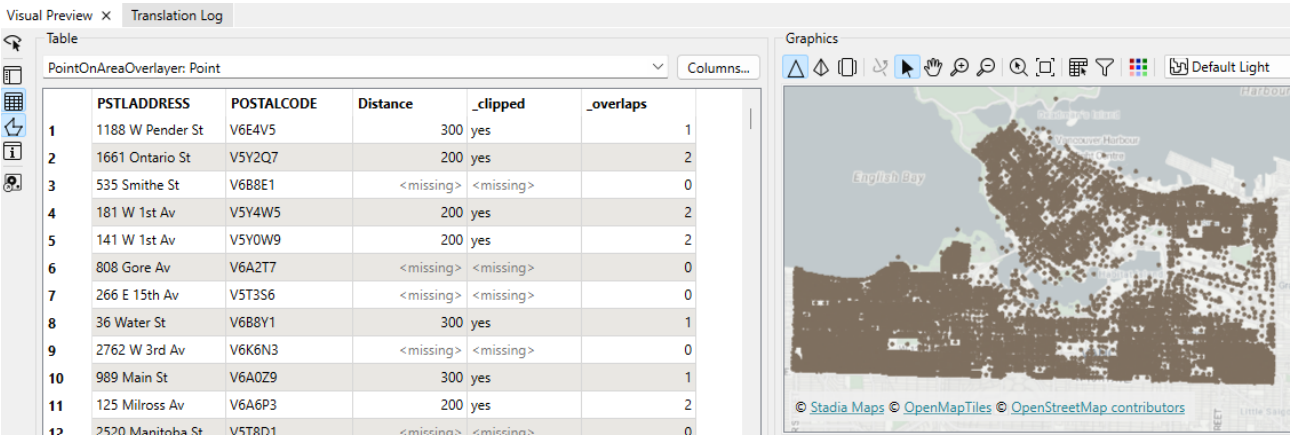
Click OK to accept your changes.

This will identify if properties are within 100, 200 and 300m from coastline, by assigning a 'Distance' attribute from buffer polygon it falls within.

3.3.5 Save and Run the Workspace

Let's save this workspace, as we haven't done that yet. Use *File* > *Save As* on the menu bar. Then navigate to where you'd like to save your workspace file:
C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.

Then run the workspace.
Inspect the Point feature cache results of the PointOnAreaOverlayer. You can now see the attributes from the buffer has been added to each address point, including the *Distance* attribute.



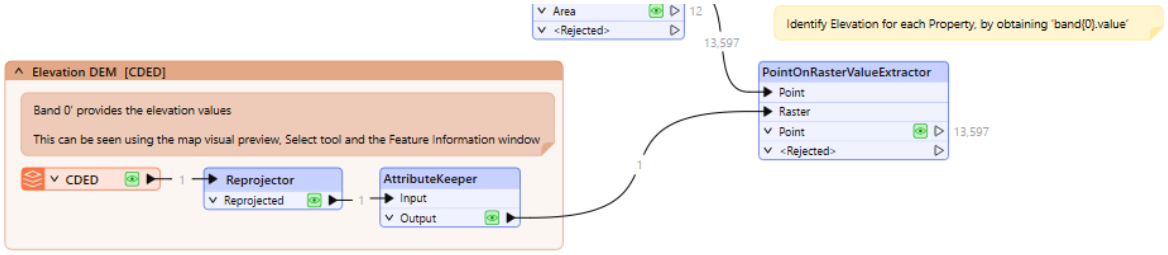
	PSTLADDRESS	POSTALCODE	Distance	_clipped	_overlaps
1	1188 W Pender St	V6E4V5	300	yes	1
2	1661 Ontario St	V5Y2Q7	200	yes	2
3	535 Smithe St	V6B8E1	<missing>	<missing>	0
4	181 W 1st Av	V5Y4W5	200	yes	2
5	141 W 1st Av	V5Y0W9	200	yes	2
6	808 Gore Av	V6A2T7	<missing>	<missing>	0
7	266 E 15th Av	V5T3S6	<missing>	<missing>	0
8	36 Water St	V6B8Y1	300	yes	1
9	2762 W 3rd Av	V6K6N3	<missing>	<missing>	0
10	989 Main St	V6A0Z9	300	yes	1
11	125 Milross Av	V6A6P3	200	yes	2
12	2520 Manitoba St	V5T8D1	<missing>	<missing>	0

The next step is to add an elevation attribute to each point.

3.3.6 Add a PointOnRasterValueExtractor transformer

Add a PointOnRasterValueExtractor. This transformer is similar to the PointOnAreaOverlayer, but instead of vector areas, it takes raster data as input.

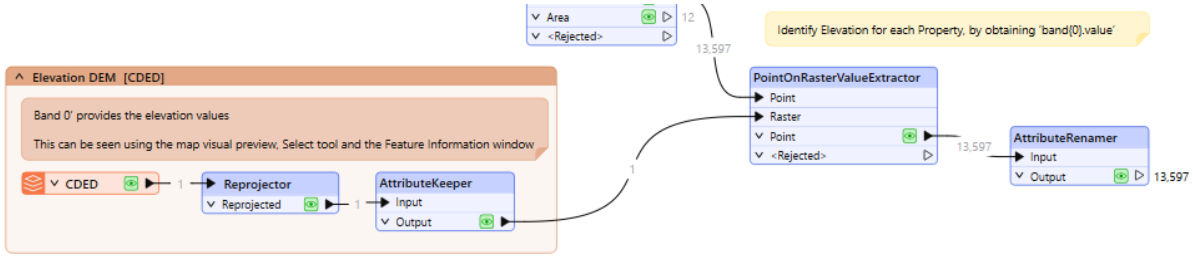
Connect the Projector transformer to the *Raster* port of the PointOnRasterExtractor. Connect the *Points* output port of PointOnAreaOverlayer to the *Point* port of the PointOnRasterExtractor.



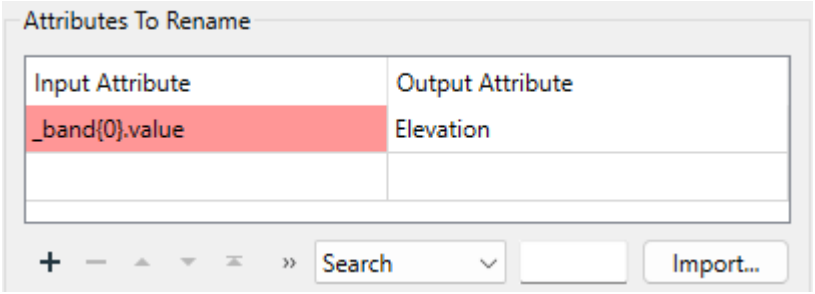
Because raster data can have multiple bands, you will not immediately see an Elevation attribute. Instead the elevation information obtained for each address point from the DEM is in the form of a List attribute called *_band{}.value*. This can be viewed using the Feature Information window.

3.3.7 Add an AttributeRenamer transformer

We can the elevation list value into an exposed attribute with this transformer. Add an AttributeRenamer transformer and connect it to the PointOnRasterValueExtractor.



For Input Attribute enter *_band{0}.value*, and for Output Attribute, enter *Elevation*



Click OK and then Run To This on the AttributeRenamer.

Inspect the feature cache to confirm you now have an Elevation attribute:

Table

AttributeRenamer: Output

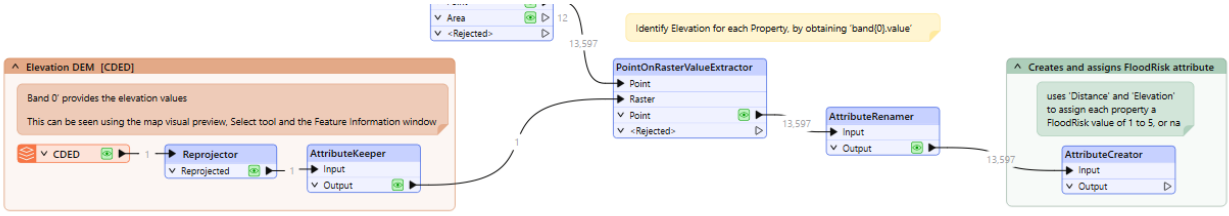
	Elevation	cded_units	PSTLADDRESS
1	22	meters	1188 W Pender St
2	6	meters	1661 Ontario St
3	29	meters	535 Smithe St
4	6	meters	181 W 1st Av
5	7	meters	141 W 1st Av
6	11	meters	808 Gore Av
7	51	meters	266 E 15th Av

Now that our address points have both a distance and an elevation value, we can assign them to a flood risk zone, 1-5. This has been done using an AttributeCreator and 'Inspector' which have been created for you in the workspace

3.3.8 Connect AttributeCreator transformer

The transformer we will use is an AttributeCreator.
Because this module's focus is spatial analysis, not attribute manipulation, we've provided this transformer in the starting workspace and already configured it using Conditional Values.

Connect the AttributeManager transformer to the AttributeCreator.



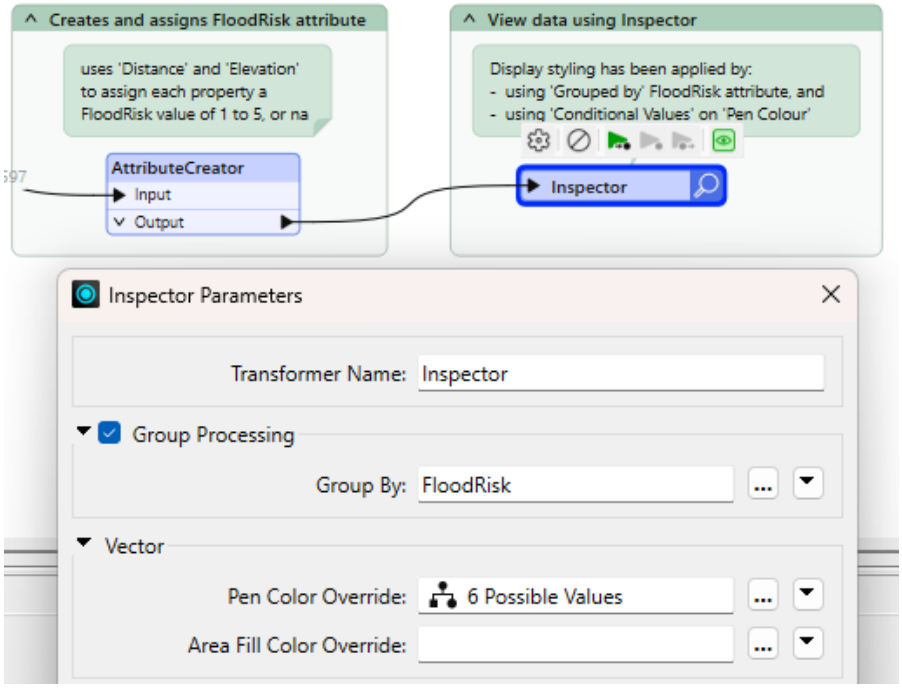
We've already configured the transformer – using a series of Test Conditions, viewable by clicking 6 Possible Values > ellipsis [. . .] button. These tests will provide a value to FloodRisk 1-5 based on the combination of Zone and Elevation. These Conditional Values translate the table above into a form FME can understand.

3.3.9 Connect Inspector

Inspecting feature caches data doesn't allow you to quickly inspect your data grouped by an attribute. However, an Inspector transformer will enable us to do that.

We have provided you with an Inspector transformer, if it is not already, connect the AttributeCreator output to it.

We have already preconfigured the inspector for you, if you open the parameters dialog and under 'Group Processing', you should see we have selected the newly created attribute called *FloodRisk*. The use of the Group Processing allows us to style the address points based on their assigned attribute value (1 to 5).

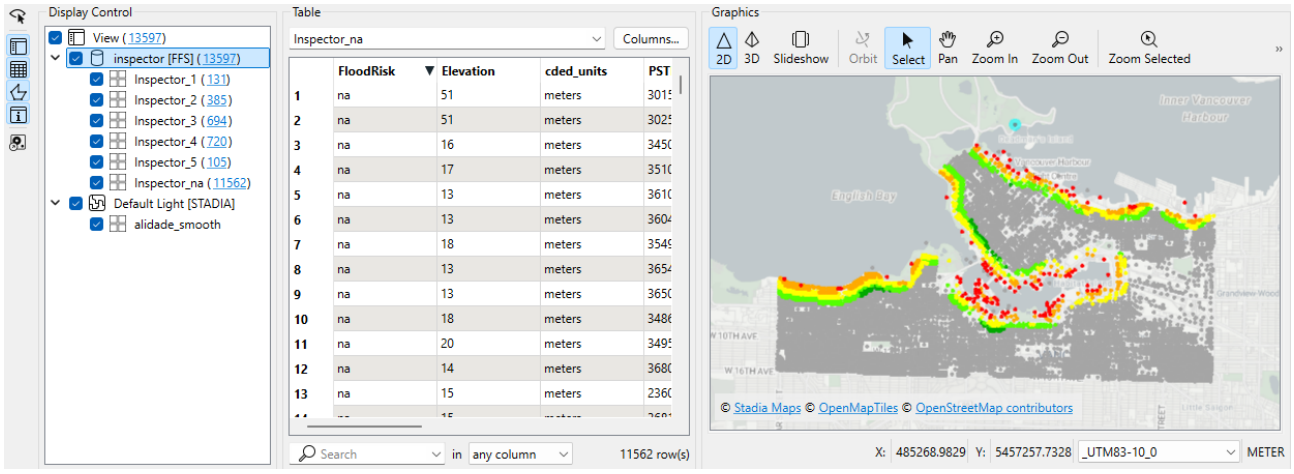


3.3.10 Resave and Rerun the Workspace

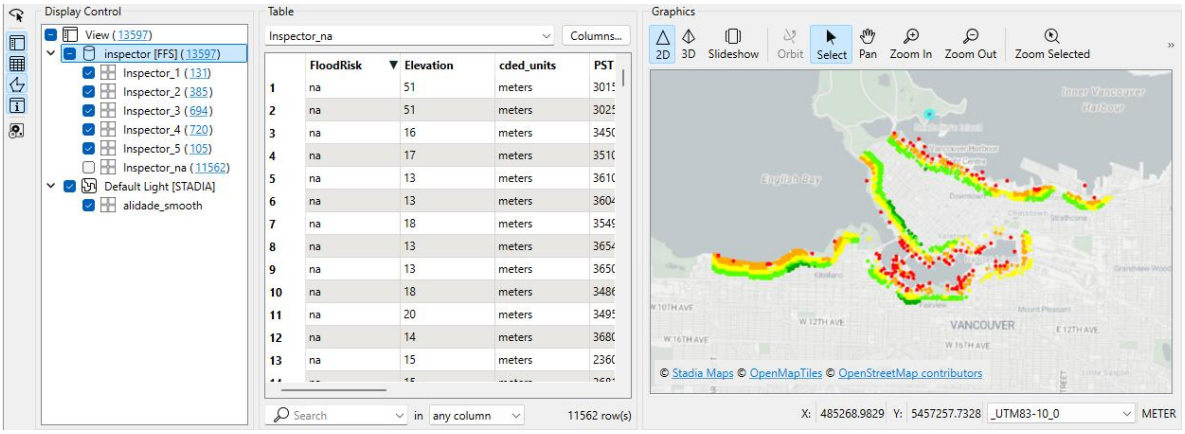
Click on the Save button, or use *File* > *Save As* on the menu bar. Then run the workspace.

The features can be examined within the Visual Preview Map Window. Using the Display Control to assist layer visibility and styling.

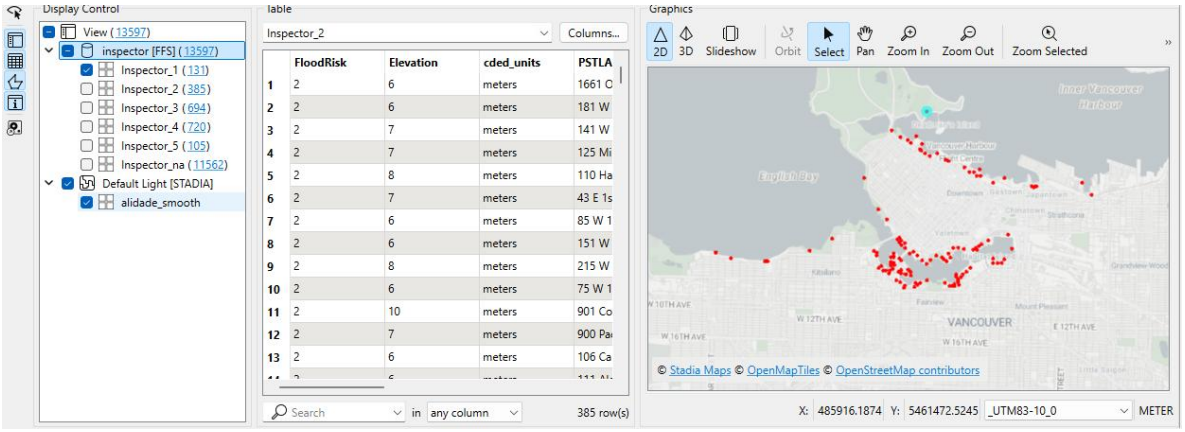
The 11,562 features in the '*Inspector_'* results layer are addresses that are not within a flood risk zone. Below that you should see the addresses separated out into groups based on their flood risk (1 to 5).

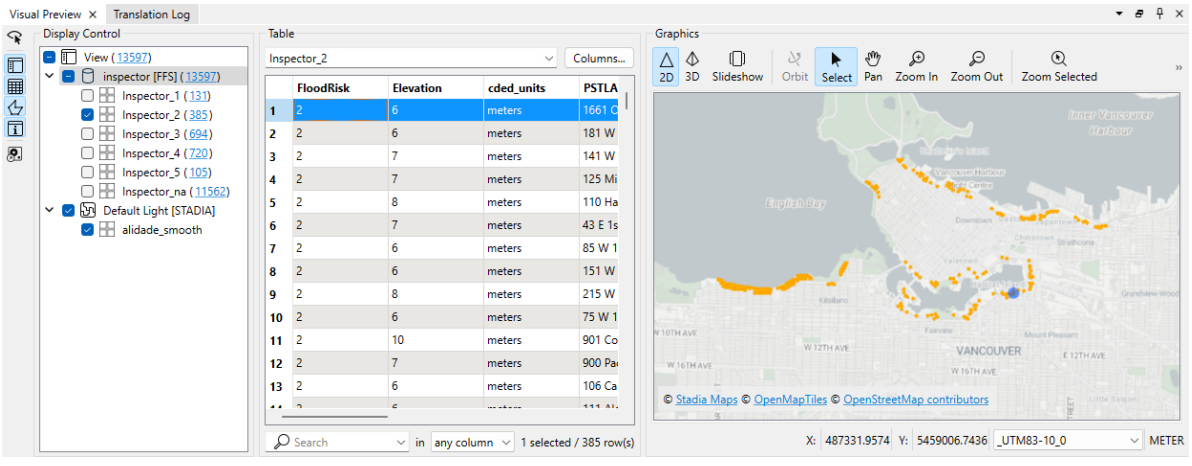


You can disable the 'N/A' category on the left hand side, this will clear some of the unnecessary data and produce a clearer result, like so:



You can also turn off zones one at a time to see which addresses are the most or least at risk.





Going through these will only show the properties given that respective flood risk rating.

3.3.11 Additional Exercise

Method 2 – Generating Multiple Buffers

If you have time, let's revisit our Buffering methods.

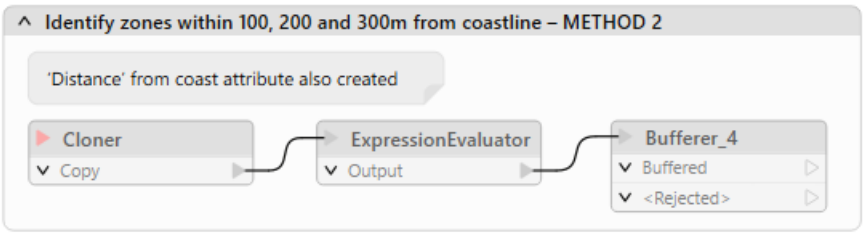
The workspace includes 2 methods for generating the multiple buffer zone polygons that we require. We have been using Method 1, but let's now explore Method 2.

Both methods could be used, and will generate the desired results. Whilst Method 1 is the easiest, Method 2 is more efficient!

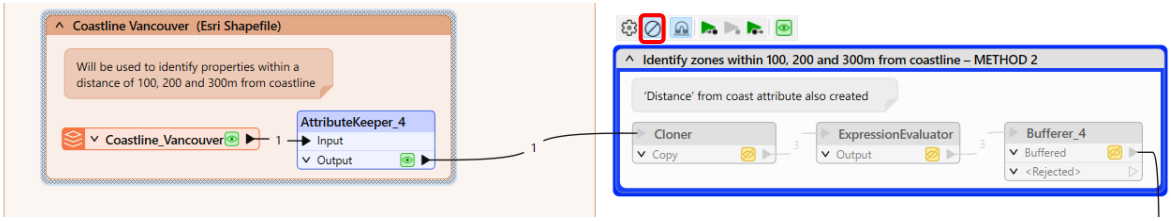
FME Lizard

We could use three separate Bufferer transformers, but there is a trick for improving your workflows in FME when you need to conduct multiple steps. Often users want to use a loop as you would in a programming language. However, you can accomplish the vast majority of tasks in FME without using loops. One way is to use what we call the 'Cloner-Loop technique'.

Method 2 is currently disabled in your workspace:

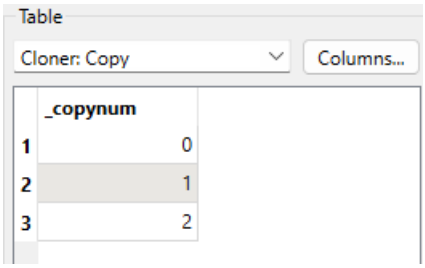


We can enable all of the content at once by clicking on the bookmark and deselecting the disable option. Then connect it to the workflow.



A Cloner transformer creates a set Number of Copies of ingoing features. In this case three coastline features will be output from the Cloner transformer.

The Cloner also adds a new attribute called '_copynum'

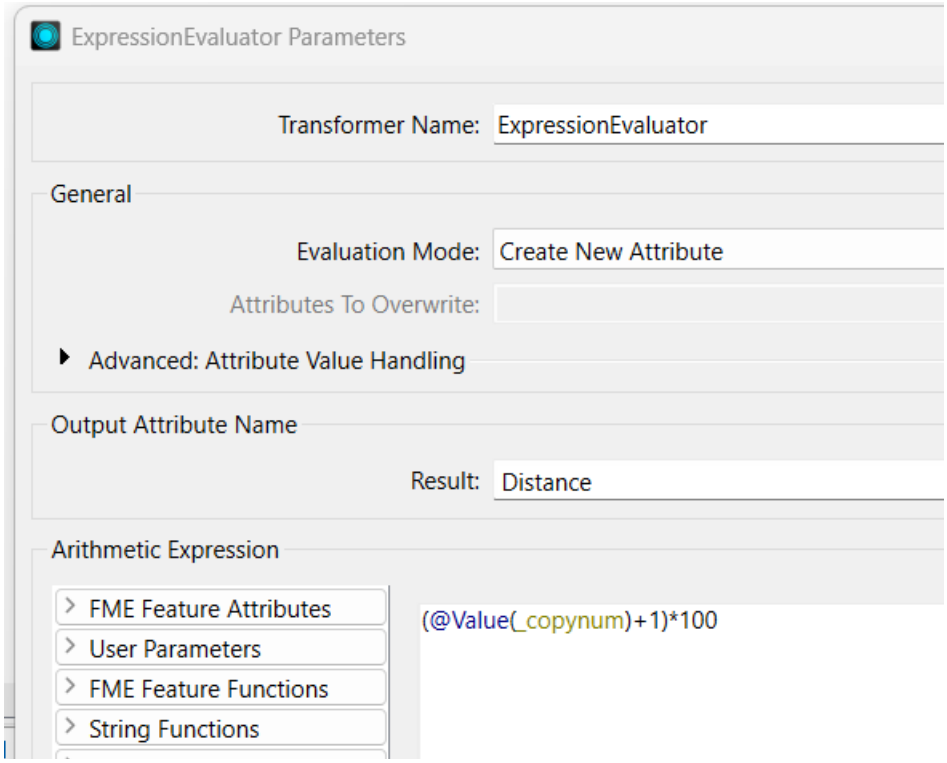


We now need to give each feature a different value to use for the buffer, thereby creating three buffer zones of different sizes (100, 200 and 300 metres)

This is achieved using an ExpressionEvaluator, and storing the buffer distance in a new attribute called 'Distance'

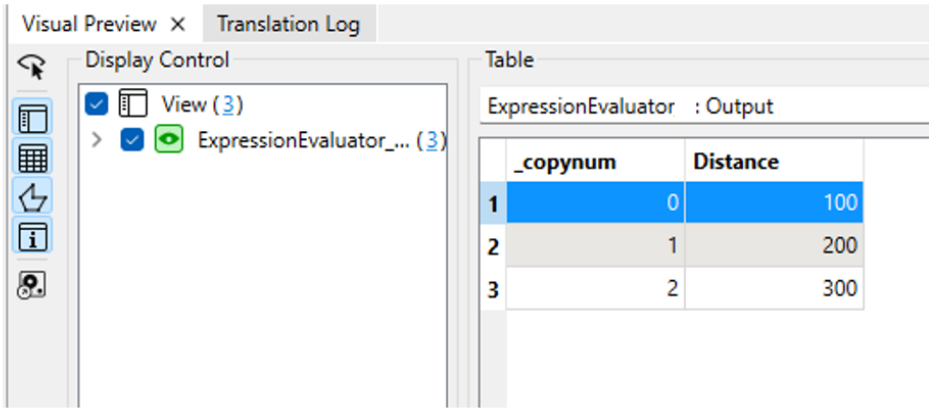
The '_copynum' attribute can be utilised to create the Distance value:

```
(@Value(_copynum)+1)*100
```

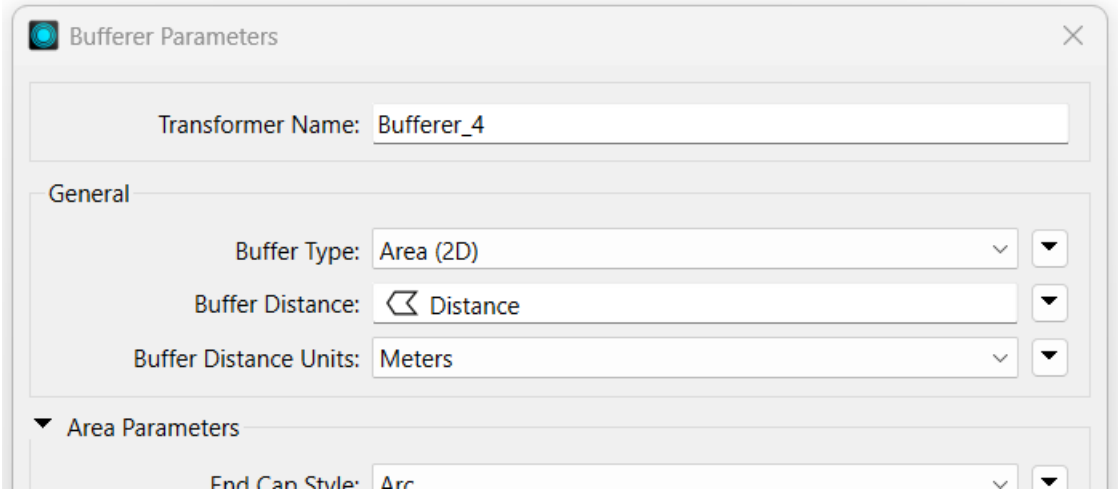


This expression will add 1 to the copy number coming from the Cloner (as counts from 0-2 instead of 1-3) and then multiply it by 100. The result is values of 100, 200, and 300, the buffer lengths we want.

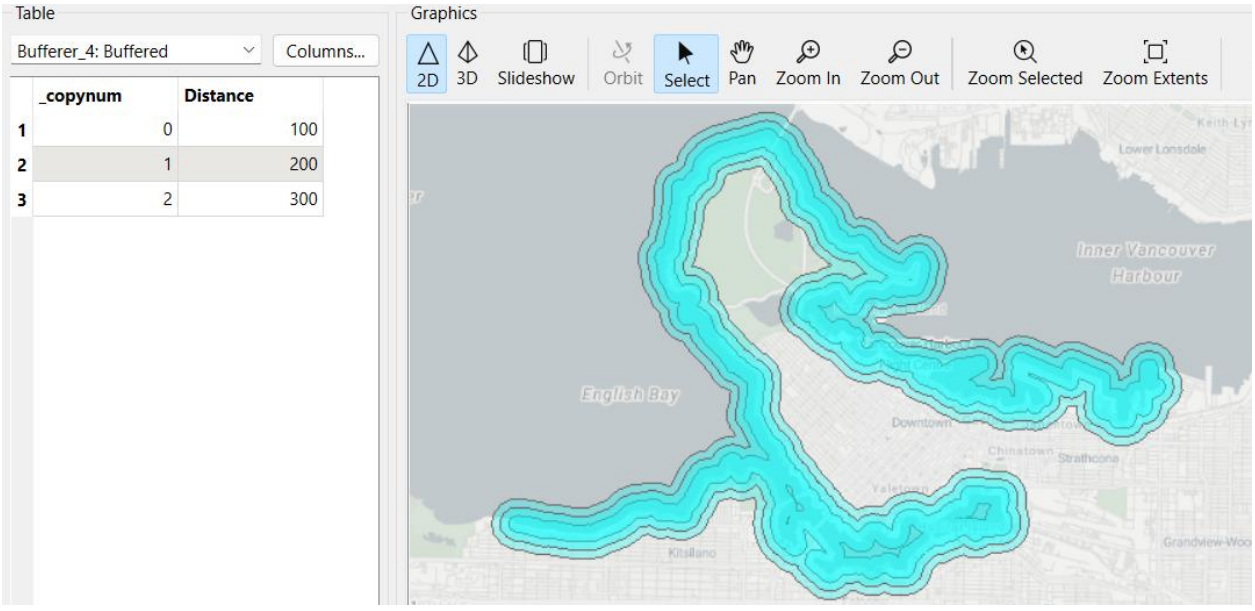
You can confirm this by clicking the ExpressionEvaluator, clicking *Run To This*, and inspecting the Feature Cache of the ExpressionEvaluator output:



Within the Bufferer transformer, the attribute of ‘Distance’ is used to specifiy the ‘Buffer Distance’ to generate around each feature.



This method, like Method 1, will create three buffer polygons showing 100, 200 and 300m from the coastline. And include a ‘Distance’ attribute.



Method 2 uses less transformers and undertakes the buffering in a single Bufferer transformer, rather than 3 Bufferer transformers. Likewise, the creation of the 'Distance' attribute is achieved using a single ExpressionEvaluator transformer, rather than 3 AttributeCreators.

Compare workspace Run time and peak memory usage for the two methods of multiple Buffer generation.

You should find that the both methods give the same results, but looking at the Translation Log you should see a difference in both workspace run duration and memory usage.

Example run logs *(your values may vary from the examples below):*

Method 1

```
Translation was SUCCESSFUL with 9 warning(s) (0 feature(s) output)
FME Session Duration: 2.2 seconds. (CPU: 0.9s user, 0.2s system)
END - ProcessID: 38220, peak process memory usage: 166232 kB, current process memory usage: 113584 kB
Translation was SUCCESSFUL
```

Method 2

```
Translation was SUCCESSFUL with 9 warning(s) (0 feature(s) output)
FME Session Duration: 1.7 seconds. (CPU: 0.7s user, 0.2s system)
END - ProcessID: 33372, peak process memory usage: 131280 kB, current process memory usage: 114236 kB
Translation was SUCCESSFUL
```

Congratulations

By Completing this exercise you have learned how to:

- use the Bufferer to buffer your spatial data
- use the Clipper to filter your spatial data
- use the PointOnAreaOverlayer to conduct a point-to-polygon spatial join
- use the PointOnRasterValueExtractor to conduct a point-to-raster overlay
- view the results of spatial analysis using an Inspector and Group By parameter

You may also have seen:

- an alternative method for generating multiple buffers using a Cloner

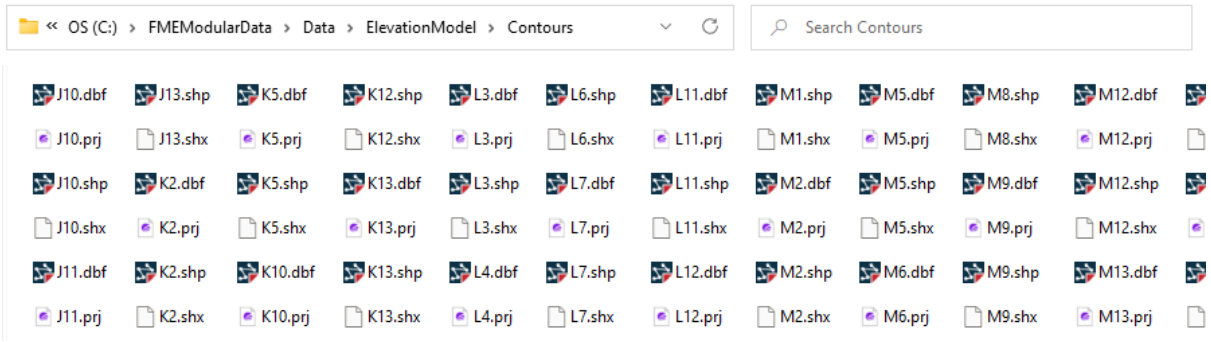
4 Workflow Design

4.1 Reading Multiple Input Files and Splitting Output into Multiple Datasets

Demonstrates	Reading multiple input files using the wildcard. Using the Single Merged Feature Type Workflow Option. Fanout on writer feature type to divide output into subsets based on an attribute. Writing data into a Zip file
Overall Goal	Read in multiple tiles of contour data using wildcard and single merged feature type, creating a single holding. Then output into subsets using fanout. Writing the results into a Zip file.
Data	Contours (ESRI Shapefile)
Start Workspace	C:\FMEModularData\Workspaces\4.01-Workflow-MultipleFiles-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\4.01-Workflow-MultipleFiles-Complete.fmw

You have been given the task of combining multiple tiles of contour data into a single coverage. Which then needs to be divided into subsets for each elevation 'band'
The multiple output files need to be Zipped.

The Contours data is currently split over numerous tiles:



Examine a few of these tiles before creating the workspace to process them.

4.1.1 Use the FME Data Inspector to Examine the input data

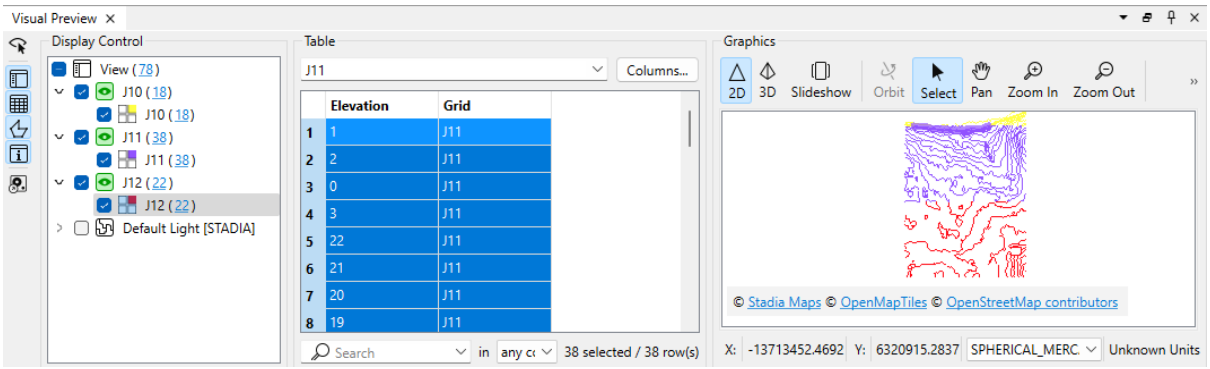
Start the FME Data Inspector by selecting it from the Windows start menu. You'll find it under Start > FME Desktop > FME Data Inspector

Use File > Open Dataset from the menu bar OR use the Open button

When prompted, fill in the fields inthe Select Dataset dialog as follows:

Reader Format	Esri Shapefile
Reader Dataset	C:\FMEModularData\Data\ElevationModel\Contours\J10.shp C:\FMEModularData\Data\ElevationModel\Contours\J11.shp C:\FMEModularData\Data\ElevationModel\Contours\J12.shp

The tiled data will look similar to this. (styling has been applied in the Data Inspector Display Control tree):

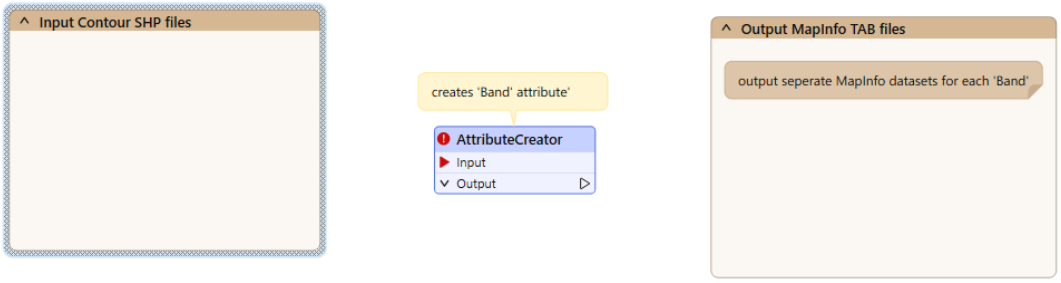


Notice the attributes are the same for all the datasets; Elevation and Grid.This will be important later.

4.1.2 Launch FME Workbench and Open Existing Workspace

Launch FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open. Navigate to and open:
C:\FMEModularData\Workspaces\4.01-Workflow-MultipleFiles-Begin.fmw

The workspace has some empty bookmarks ready for us to fill with a reader and writer. It also contains an AttributeCreator which is generating a new attribute called 'Band'.



4.1.3 Add Reader

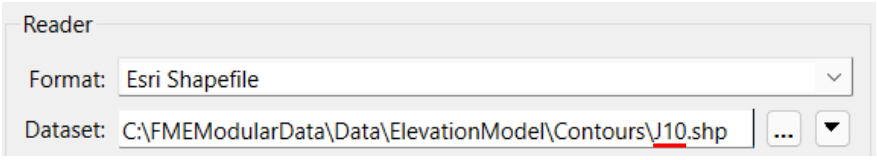
Use either the Reader button, or use Readers > Add Reader… from the menu bar.
The Add Reader dialog will open, in which define the Format and Dataset settings as follows:

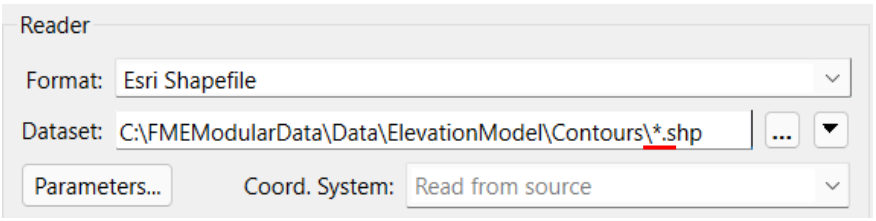
Reader Format	Esri Shapefile

We are going to use a Wildcard to read in all the .shp files within the Contours folder.
Click the ellipse button for *Dataset:* and navigate to C:\FMEModularData\Data\ElevationModel\Contours

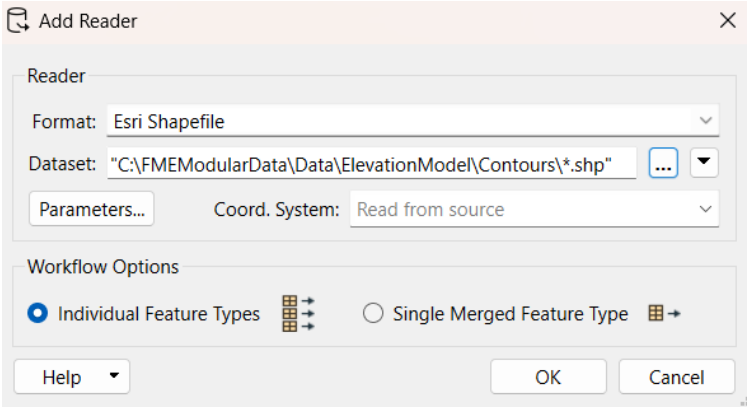
Select just one of the Shapefiles.

Back in the Add Reader dialog replace the file name (not extension) with a wildcard *





Finally, set the **Workflow Option** to Single Merged Feature Type



Click OK. Into the Navigator panel an ESRI Shapefile Reader will be added and on the canvas there will be one reader feature type called <All>

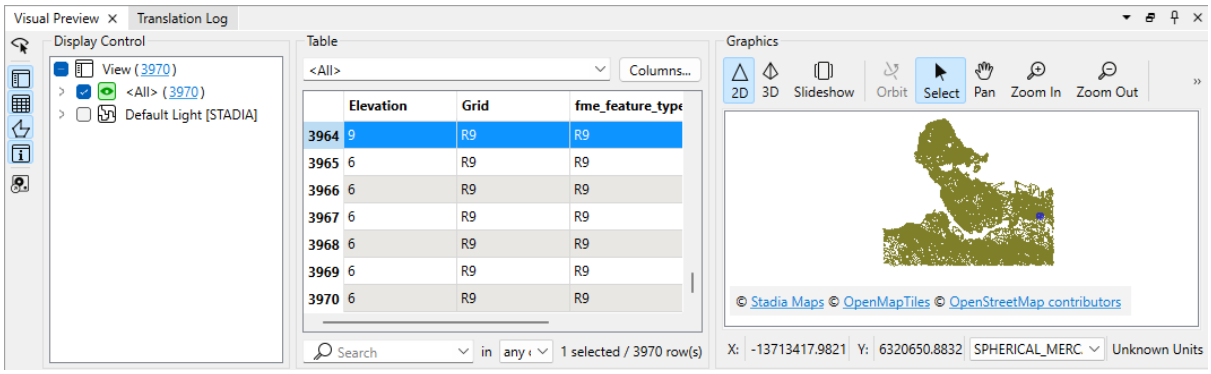
The reader is now configured to read in all files with the extension .shp within the Contours folder, and all features from all the files will route into the workflow via the Single Merged Feature Type.
Connect to reader feature type to the AttributeCreator transformer.

4.1.4 Save and Run the Workspace

Use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.
Then run the workspace.

4.1.5 Inspect the AttributeCreator Feature Cache

Click on the AttributeCreator transformer feature cache to examine the features at this point in the workflow.



We now have a single coverage of contour data – instead of split into separate tiles. Each contour feature has been assigned a *Band* value.

We need to output the contour data to separate MapInfo datasets for each *Band* of elevation. Which also need to be in a Zip file.

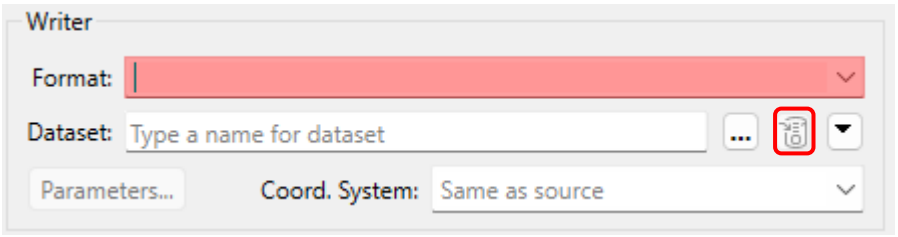
4.1.6 Add a Writer

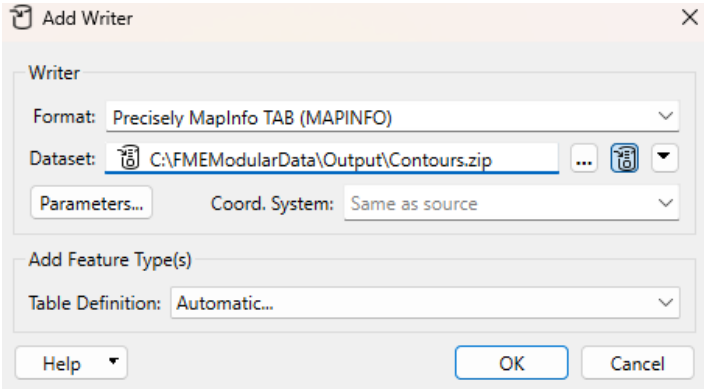
Use either the Writer button, or use Writers > Add Writer… from the menu bar.
The Add Writer dialog will open, in which define the Format and Dataset settings as follows:

Writer Format	Precisely MapInfo TAB (MAPINFO)

Click the ellipse button for *Dataset:* and navigate to: *C:\FMEModularData\Output*
then click on the *Zip output* button
Change the name of the Output.zip to *Contours.zip*

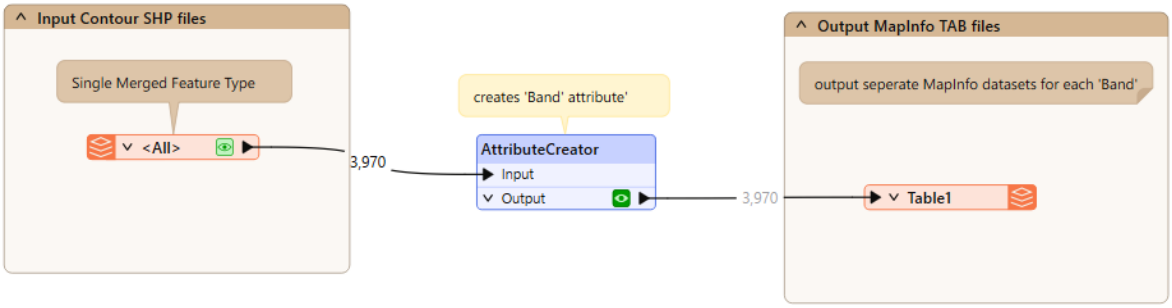
Set the Table Definition: to *Automatic*





Click OK.
A Feature Type dialog will open. Don't worry about setting a Table name, we'll deal with this in the next steps. Click OK and the MapInfo writer feature type will be added to the canvas.

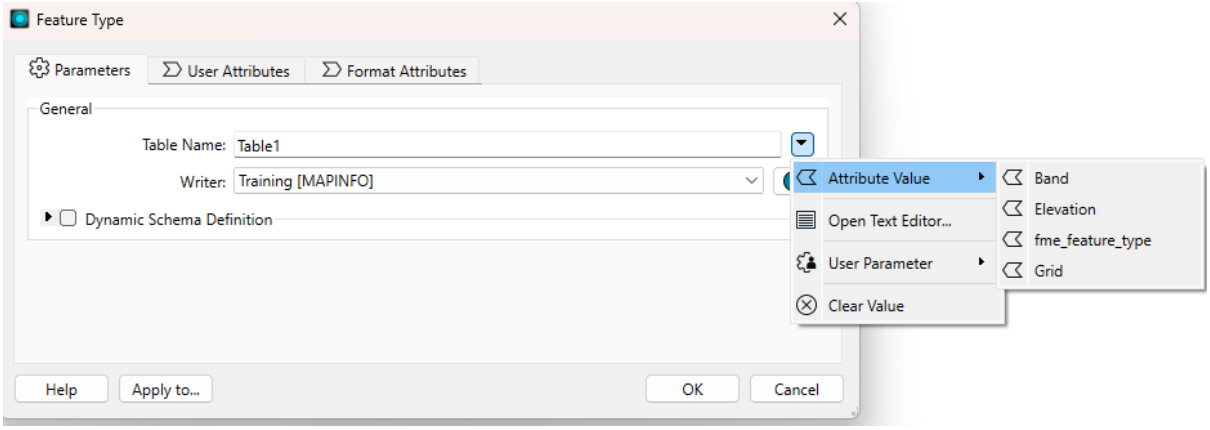
Connect it to the AttributeCreator.



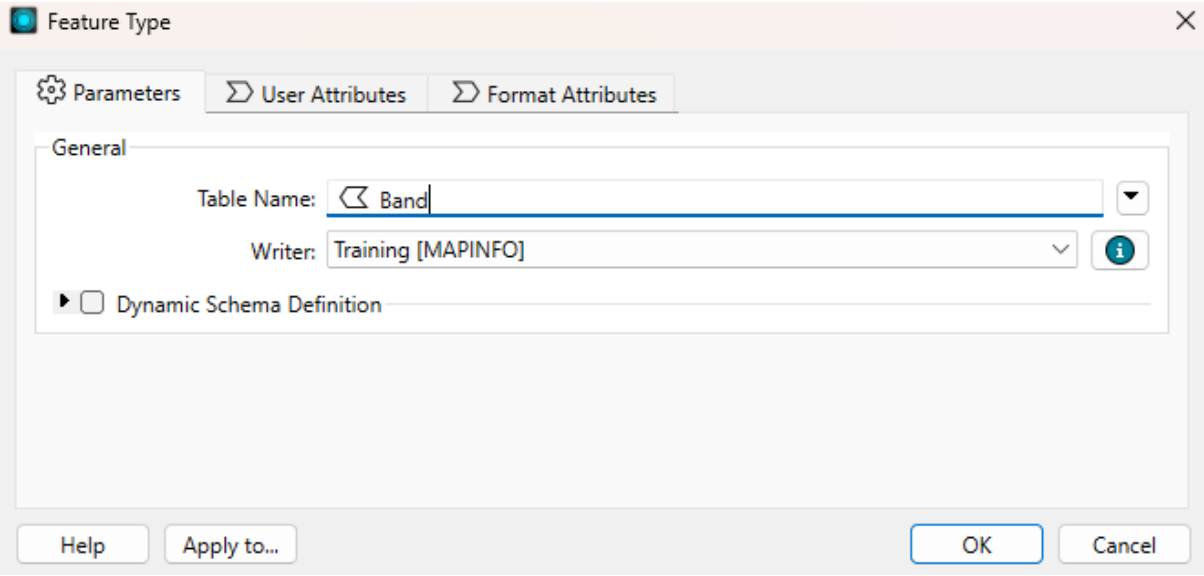
If we run the workspace now it would write all the features out to a single MapInfo dataset. Instead, we want separate MapInfo datasets for each Band category.

4.1.7 Fanout on Feature Type

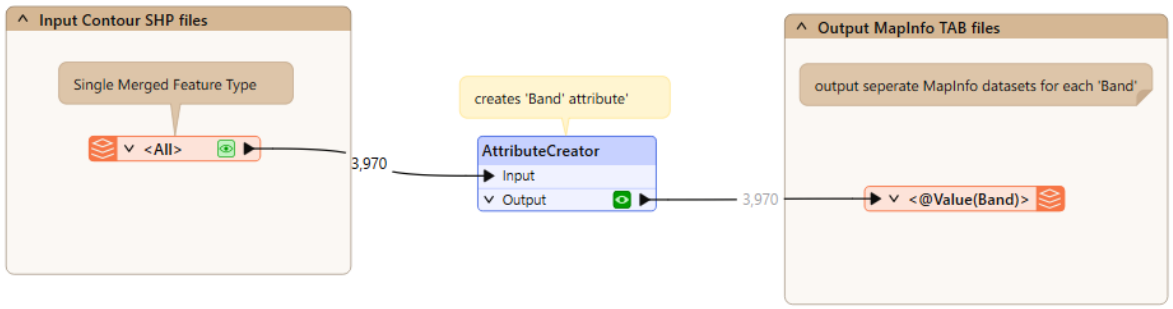
Click on the parameters cog of the MapInfo writer feature type. Use the dropdown menu arrow to the right of *Table Name:*



From the attributes list, select *Band*



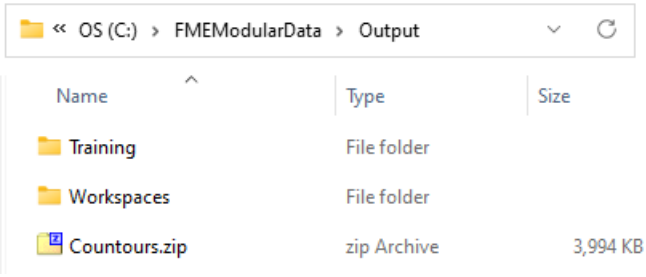
We have now applied Fanout on the feature type, using the *Band* attribute.



4.1.8 Resave and Rerun the Workspace

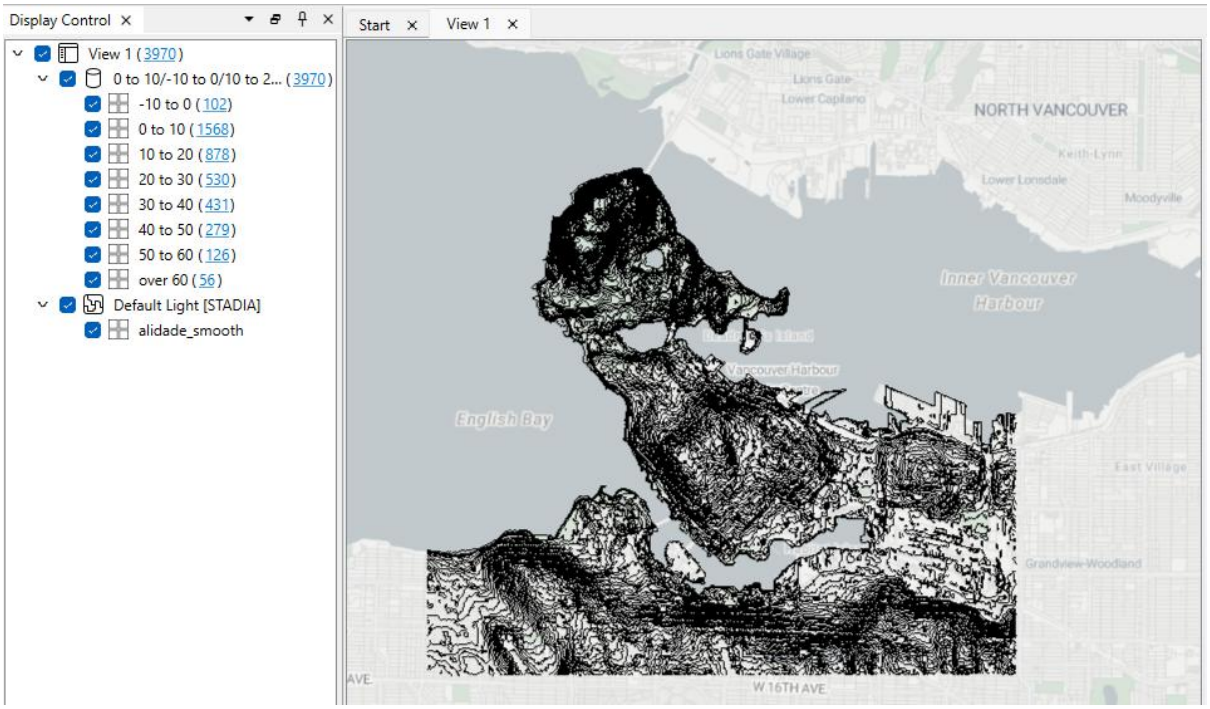
Click on the Save button, or use File > Save As on the menu bar.
Then run the workspace.

The results will be in a Zip file within the Output folder:



4.1.9 Examine the Output

Extract the content of the Zip file.
Inside there will be separate MapInfo datasets for each contour band. Use the Data Inspector to examine them.



Congratulations

By Completing this exercise you have learned how to:

- Reading multiple input files using the wildcard
- Using the Single Merged Feature Type Workflow Option
- Fanout on writer feature type to divide output into subsets based on an attribute
- Write data into a Zip file

4.2 Building Flexible Workflows

Demonstrates	Identify when unexpected Input is encountered. Handle unexpected input using Import Feature Types. Design a flexible workspace to automatically handle changing feature type names in source data (preventing unexpected input).
Overall Goal	Create a flexible workspace to handle changing feature type names in source data
Data	CommunityPlan2019 (Bentley MicroStation Design V8) CommunityPlan2020 (Bentley MicroStation Design V8)
Start Workspace	C:\FMEModularData\Workspaces\4.02-Workflow-Flexible-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\4.02-Workflow-Flexible-Complete.fmw"

We have an ongoing need to convert community plans from DGN to ESRI Shapefiles every year. It would be good if we could handle updated schemas without having to rebuild our workspace.

We have an existing workspace which processed the 2019 CommunityPlan data. Let's take a look at the original 2019 data and the workspace.

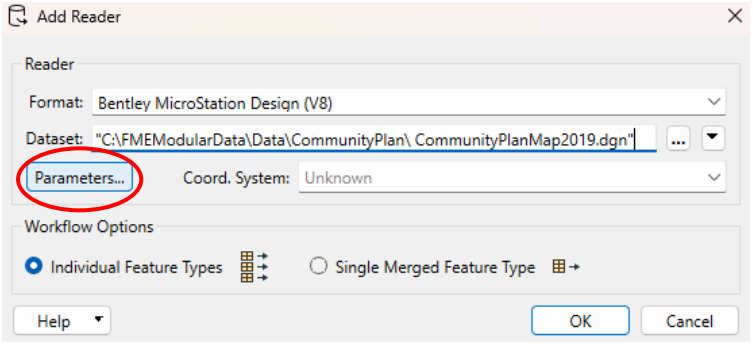
4.2.1 Open the Bentley file into the Data Inspector

Launch the Data Inspector, if it's not already open.

Use File > Open Dataset from the menu bar OR use the Open button

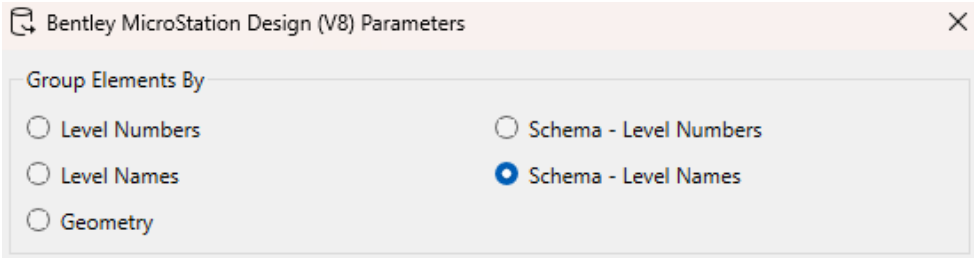
When prompted, fill in the fields inthe Select Dataset dialog as follows:

Reader Format	Bentley MicroStation Design (V8)
Reader Dataset	C:\FMEModularData\Data\CommunityPlan\ CommunityPlanMap2019.dgn



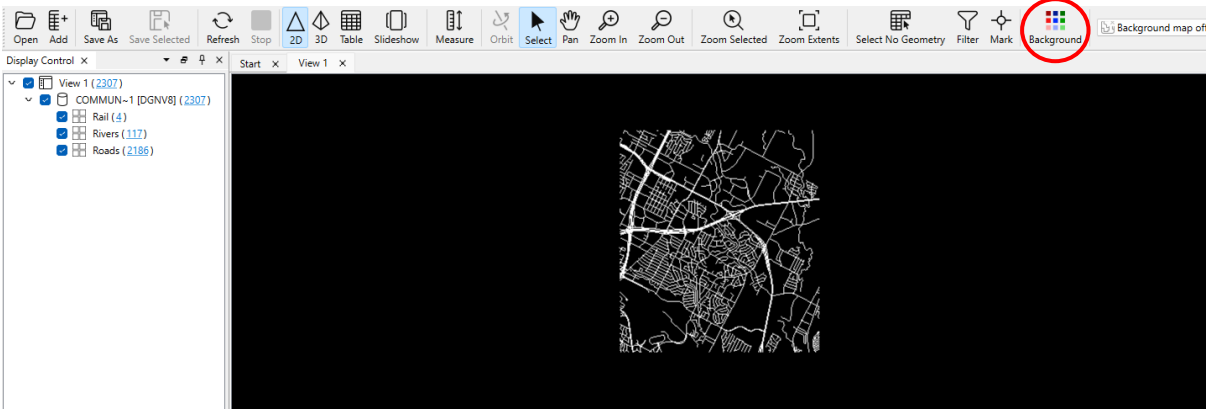
Also, click the **Parameters…** button
Bentley DGN files, like many formats, contain multiple layers of data. When working with Bentley and AutoCAD data it's important to expose the Schema Level Names (so that the layer names are available to us when working with the data).

In the Group Elements By section, choose Schema – Level Names.

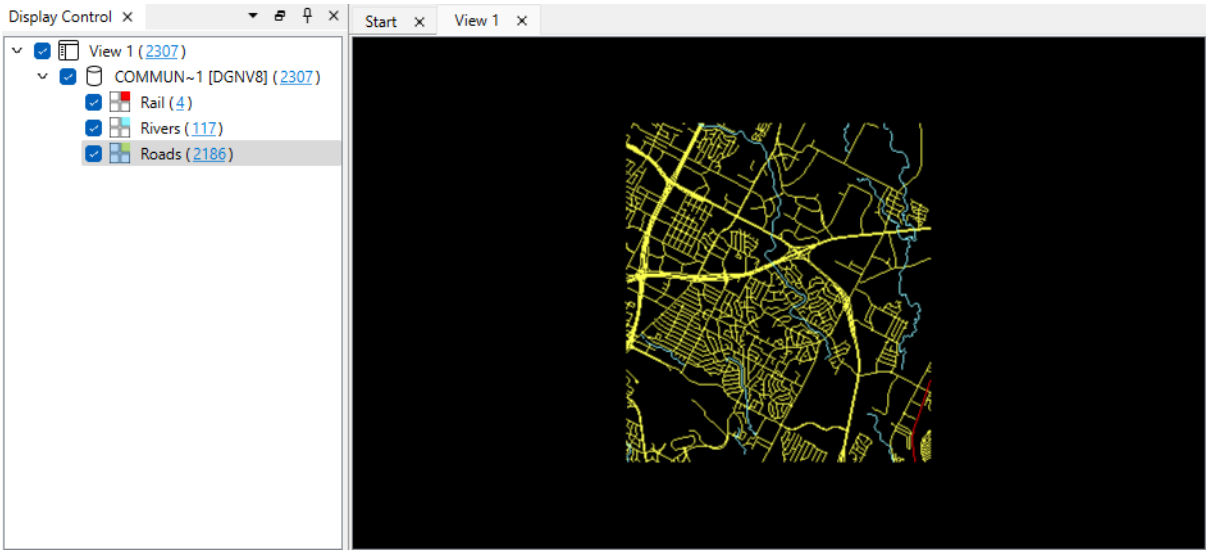


Click OK, then click OK again to read the data into the Data Inspector.

The Bentley DGN dataset contains line features that are styled white in colour. Therefore in order to see the features within the map view you'll need to either change the Background colour:



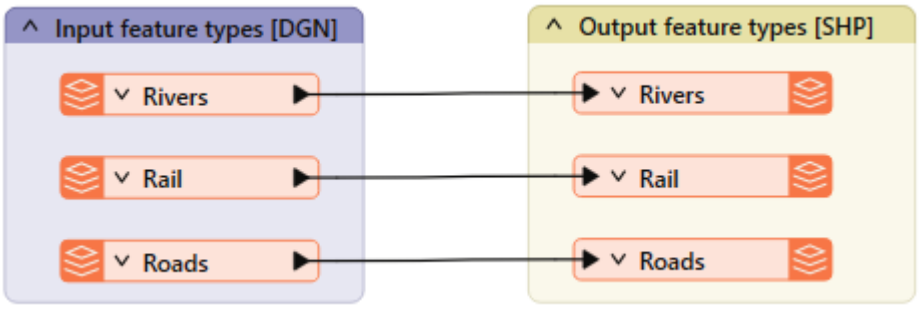
Or change the map layer styles within the Display Control:



There are 3 Feature Types in this dataset; Rail (with 4 Features), Rivers (with 117 Features) and Roads (with 2186 Features).

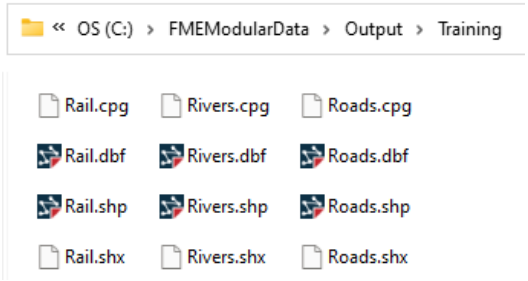
4.2.2 Launch FME Workbench and Open Existing Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open. Navigate to and open:
C:\FMEModularData\Workspaces\4.02-Workflow-Flexible-Begin.fmw



4.2.3 Run the Workspace and Examine output

Run the workspace. Then navigate to C:\FMEModularData\Output\Training to view the written data.
There are separate ESRI Shapefiles for Rail, Rivers and Roads.

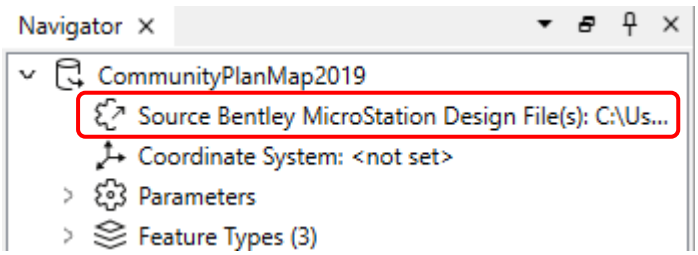


We now need to process the 2020 version of the CommunityPlan data.

4.2.4 Modify Input Data Source on DGN Reader

Within Navigator Panel go to the parameters of the Bentley DGN Reader and change the Dataset Source to instead point to the new version (2020) of CommunityPlan data - which now needs to be processed.

To do this double click on the purple cog:

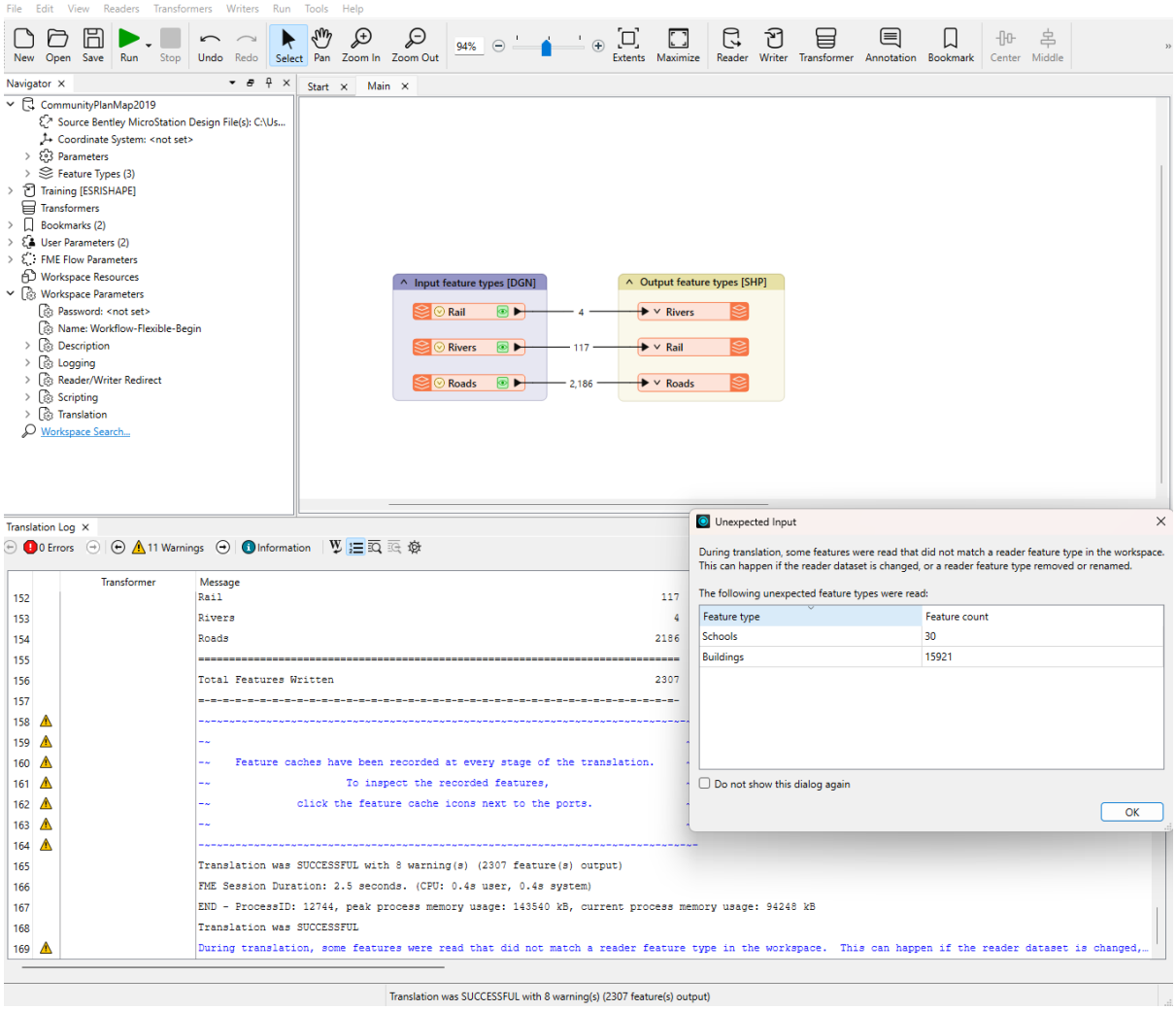


Then navigate to and select the latest DGN file:
C:\FMEModularData\Data\CommunityPlan\CommunityPlanMap2020.dgn

4.2.5 Run the Workspace

Now Rerun the workspace to process the new DGN file.

The translation reports as being successful but warning appears in the Translation Log (you may also get a pop up): 'During translation, some features were read that did not match a reader feature type in the workspace. This can happen if the reader dataset is changed, or a reader feature type removed or renamed.'



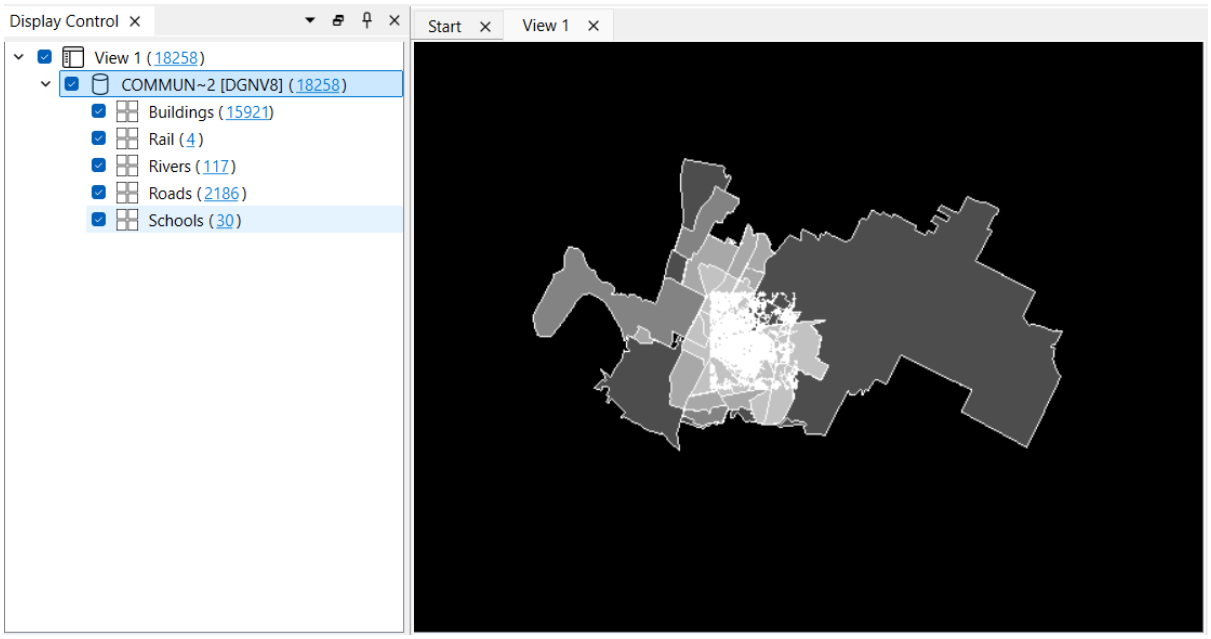
Every time FME reads a dataset, it checks the feature types inside that dataset to ensure that they are all defined within the workspace schema. If there are feature types that exist in the dataset, but do not exist in the workspace, then features are classed as "unexpected" and filtered out by a function called the Unexpected Input Remover.

The actions of the Unexpected Input Remover are reported in the Log file and also through a dialog that opens at the end of a translation – we have just seen that!

4.2.6 Examine CommunityPlanMap2020.dgn

Examine the CommunityPlanMap2020.dgn input data using the Data Inspector, it its clear what has happened.

In addition to the expected feature types of Road, River and Rail – there is now Buildings and Schools. Our workspace is not design to handle these!

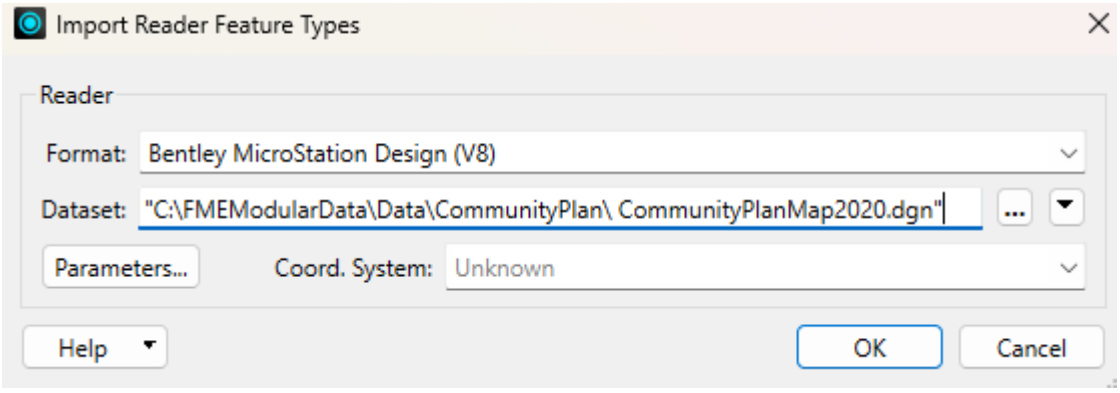


We could modify our workspace to accommodate Buildings and Schools in two different ways. We'll look at both now.

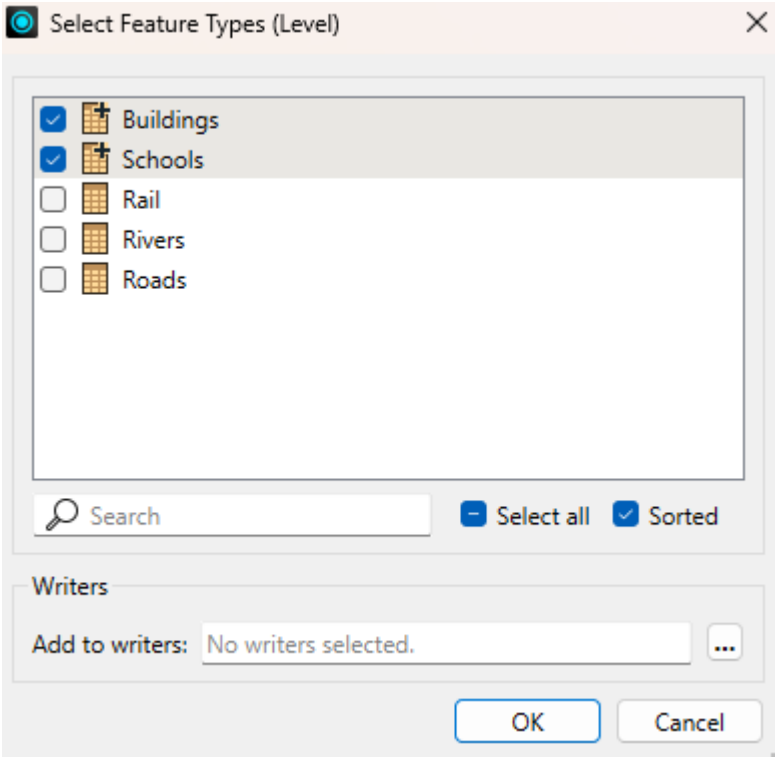
4.2.7 Import Feature Types onto Reader

Add the two missing feature types to the reader. Using *Readers > Import Feature Types..* from the menu bar, then select the CommunityPlanMap2020.dgn dataset.

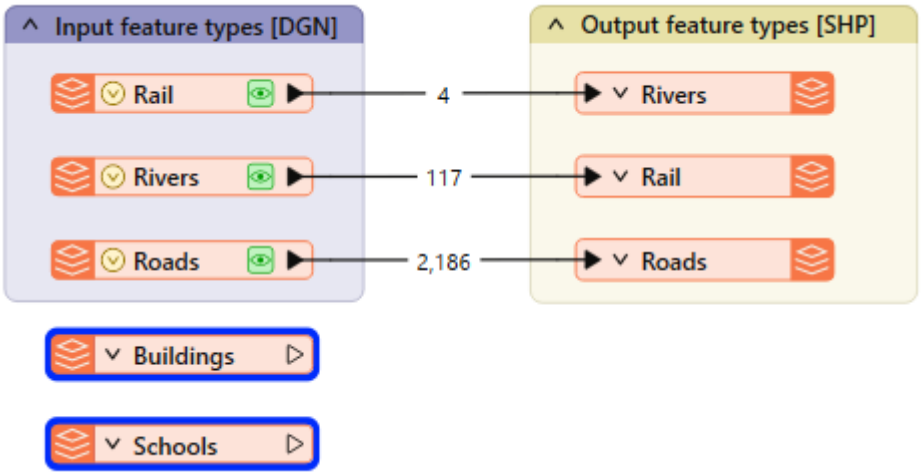
Don't forget to set *Schema -Level Names* within the Parameters…



The choose dataset will be examined and its feature types will be listed. Select the two missing feature types that we require.



The two extra feature types will be added to the canvas.



The workspace will now handle incoming feature types called Roads, Rivers, Rail, Buildings or Schools. This enables us to process the DGN 2020 file now but doesn't solve the problem for next year when different feature types could be encountered again!

Instead, lets change the design of this workspace and make it a flexible workflow, that can be easily reused again and again.

4.2.8 Replace the DGN Reader

Delete the existing DGN reader (either from the Navigator Panel) or by selecting all the DGN reader feature types on the canvas and then deleting them (also choosing to delete the reader when prompted).

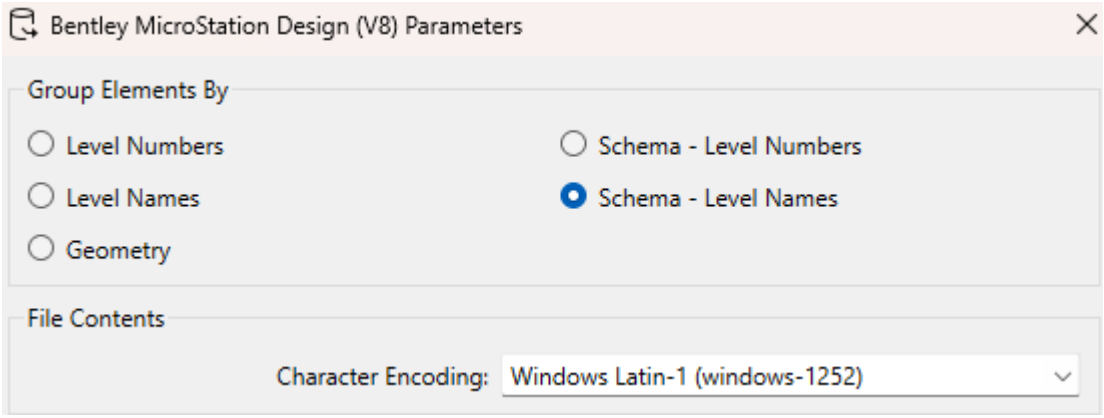
Add a new reader using either the Reader button, or use Readers > Add Reader… from the menu bar. We'll use the CommunityPlanMap2019.dgn as the input dataset to use.

We will use the Workflow Option of Single Merged Feature Type this time.

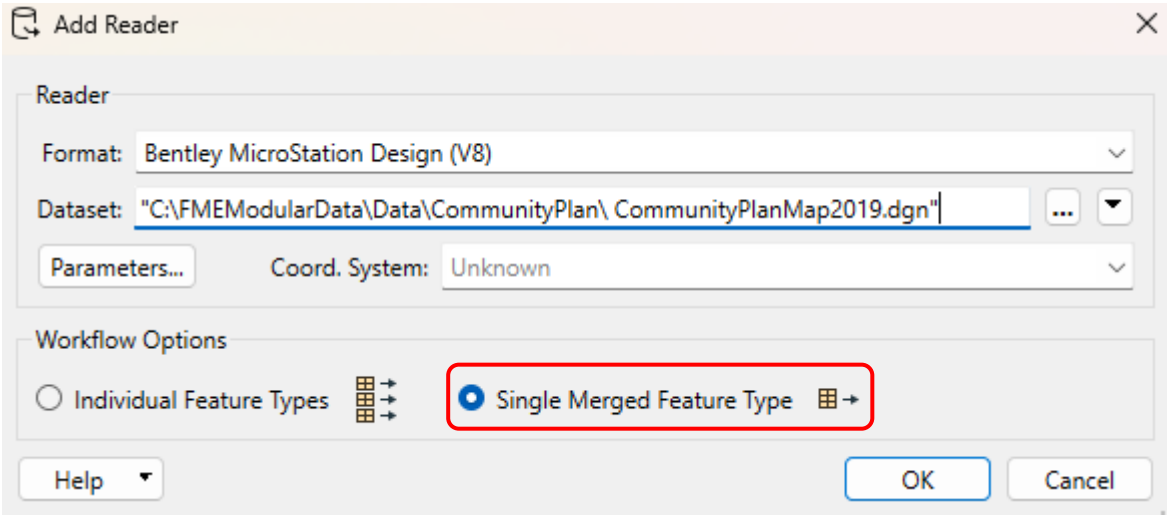
The Add Reader dialog will open, in which define the Format and Dataset settings as follows:

Reader Format	Bentley MicroStation Design (V8)
Reader Dataset	C:\FMEModularData\Data\CommunityPlan\ CommunityPlanMap**2019**.dgn

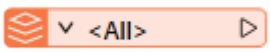
Click the **Parameters…** button. In the Group Elements By section, choose *Schema – Level Names*.



Finally, set the *Workflow Option* to *Single Merged Feature Type*



The new DGN reader has been added and will be listed in the Navigator Panel, it will have one feature type on the canvas – as single merged feature type:



Our DGN reader is no longer hard-coded to handle feature types of a specific name – thus

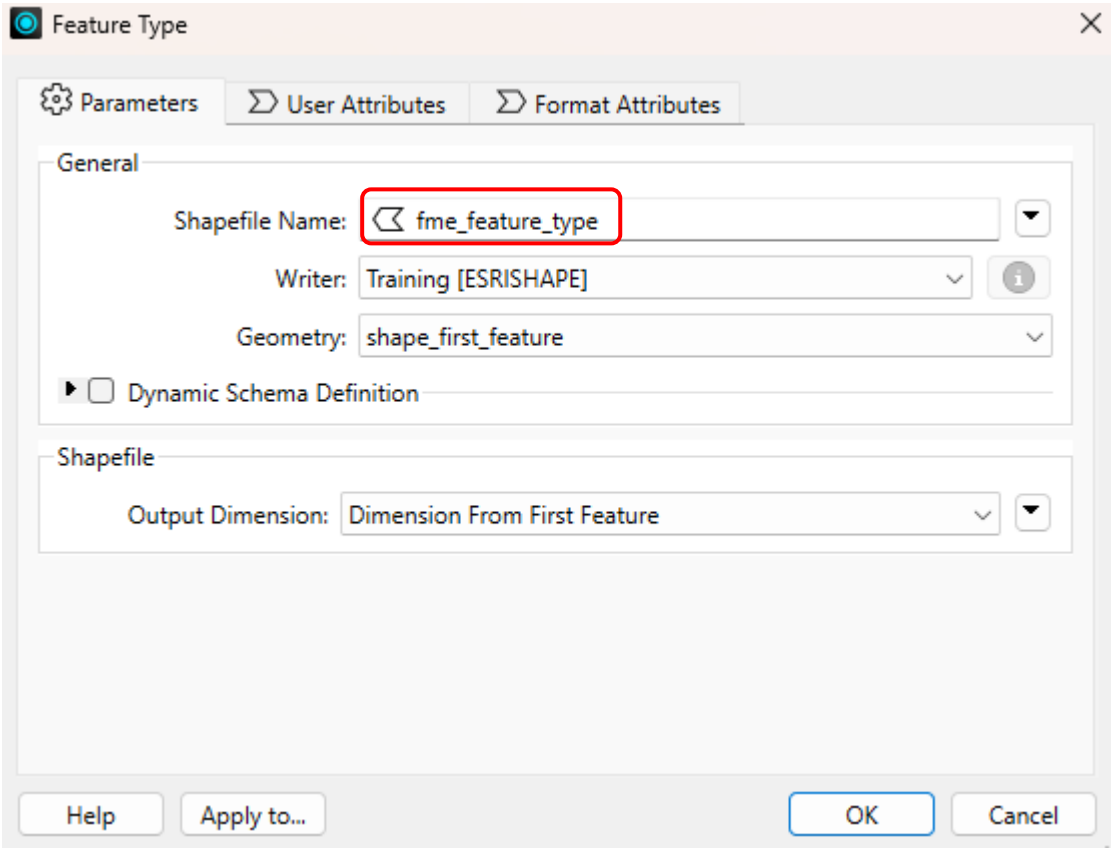
making our workspace more flexible and reusable.

We now need to turn our attention to the other end of the workspace, the writer. Again, the original workspace design is configured to only write feature types called Roads, Rail and Rivers. If we have differing incoming feature types (such as schools and buildings) we need the writer to automatically output datasets appropriately. For this will employ the technique of Fanout.

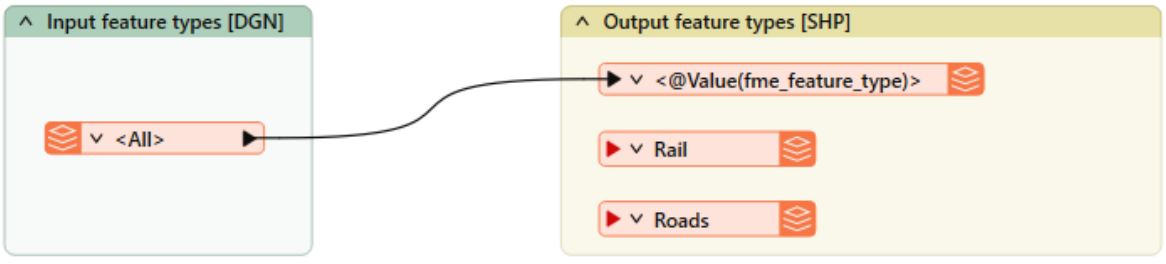
4.2.9 Fanout on Feature Type

Double-click on one of the SHP feature types (Rivers, Roads or Rail). Use the dropdown menu arrow to the right of *Shapefile Name:*

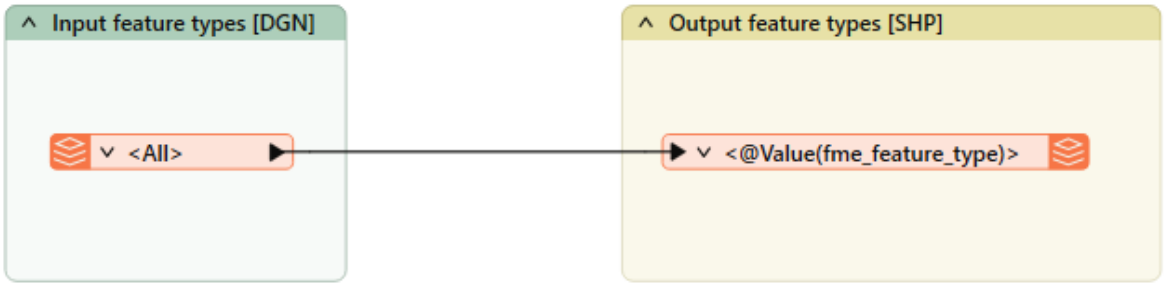
From the attributes list select *fme_feature_type*



Connect the DGN reader feature type *<All>* to the modified SHP writer feature type:



Delete the two remaining SHP writer feature types, as we no longer require them.



4.2.10 Save and Run the Workspace

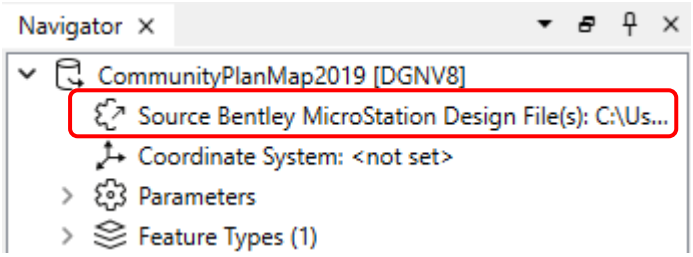
Click on the Save button or use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.
Then run the workspace.

Let's now try to process the new version (2020) of CommunityPlan data again.

4.2.11 Modify Input Data Source on DGN Reader

Within Navigator Panel go to the parameters of the Bentley DGN Reader and change the Dataset Source to instead point to the new version (2020) of CommunityPlan data - which now needs to be processed.

To do this double click on the purple cog:

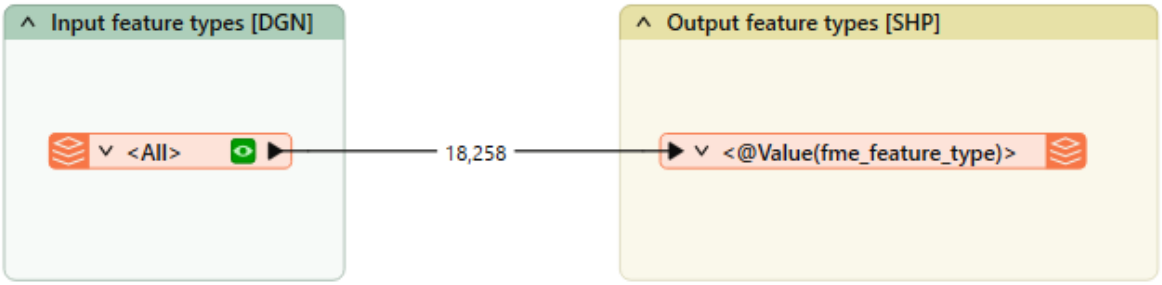


Then navigate to and select the latest DGN file:
C:\FMEModularData\Data\CommunityPlan\CommunityPlanMap2020.dgn

4.2.12 Run the Workspace

Now Rerun the workspace to process the new DGN file.

Hurrah! We now have the data we want entering the workspace and we successfully translated to ESRI Shapefiles. And more importantly, if future CommunityPlanMap DGN files contain different feature types our workspace will automatedly handle them.



In the output folder, you will find 5 Shapefiles that are holding Features from 5 different DGN levels.

You have just created a flexible workflow!

How did we do it?

- First of all, Single Merged Feature Types – a Reader Workflow Option that we used to replace individual Feature Type boxes.

- Secondly, using Fanout on the Shapefile Writer feature type to output separate Shapefile datasets for each value in the 'fme_feature_type' attribute.

Congratulations

By Completing this exercise you have learned how to:

- Identify when Unexpected Input is encountered
- Handle unexpected input using Import Feature Types
- Design a flexible workspace using Single Merged Feature Types and Fanout

4.3 Using the WorkspaceRunner

Demonstrates	Fundamentals of using the WorkspaceRunner
Overall Goal	Chain two workspaces using the WorkspaceRunner
Data	Contours (ESRI Shapefile)
Start Workspace	C:\FMEModularData\Workspaces\4.03-Workflow-WorkspaceRunnerChild-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\4.03-Workflow-WorkspaceRunnerParent-Complete.fmw

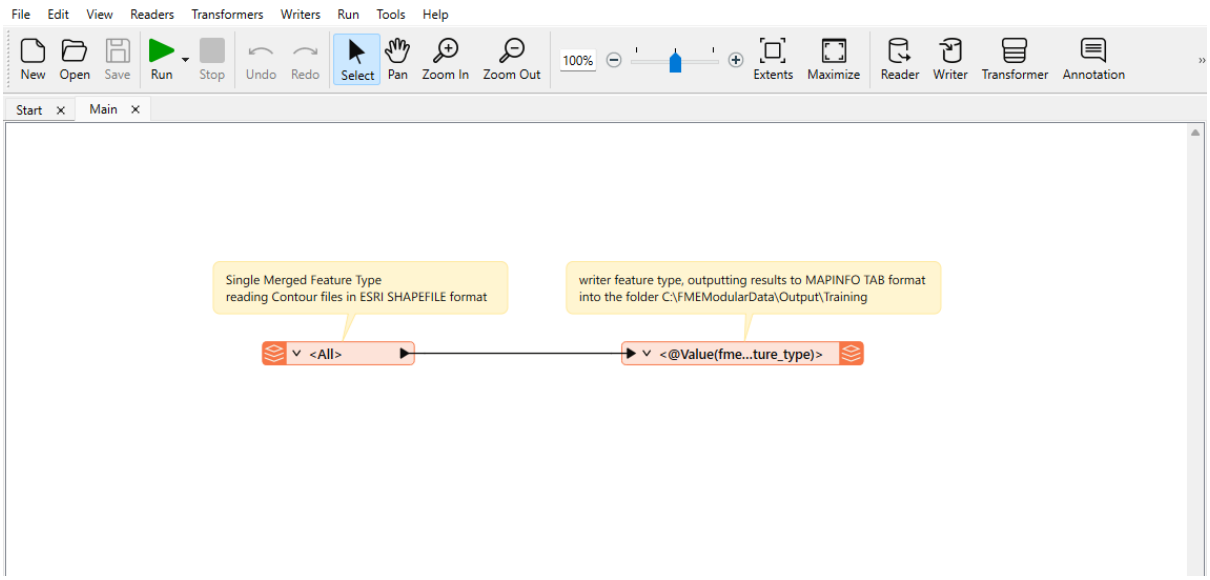
We have a future need to enhance our production environment deployment of FME Desktop; batch processing of bulk data, generating email success/failure notifications for scheduled workspaces, chain multiple workspaces to run in series or parallel, and more…But before we can employ these techniques, we first need to understand the fundamentals of using the WorkspaceRunner transformer.

To demonstrate how the WorkspaceRunner can be used to trigger other workspaces to run, we'll setup a simple parent-child workspace scenario. We'll use the parent workspace to specify one or multiple contour tiles to process, then it will initiate the child workspace – which in turn will actually perform the translation to convert the chosen contour tile(s) into MapInfo TAB format.
The important part is how to pass information between the parent and child workspaces!

It's usually easiest to create and configure the child workspace first, then create the parent workspace. In this exercise the child workspace has already been created for us.

4.3.1 Launch Workbench and open existing Child Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open. Navigate to and open:
C:\FMEModularData\Workspaces\4.03-Workflow-WorkspaceRunnerChild-Begin.fmw



This workspace is already configured for us. It consists of an ESRI Shapefile Reader with a Single Merged Feature Type. – therefore this reader will accept and read in any of the SHP contour tiles of our choice.
The workspace also has a Writer, of MapInfo TAB format. It has Fanout applied on its Feature Type, using the *fme_feature_type* attribute. – therefore will create separate MapInfo TAB datasets for each contour tile it receives.

We have finished examining this workspace, we can now close it and create the Parent workspace.

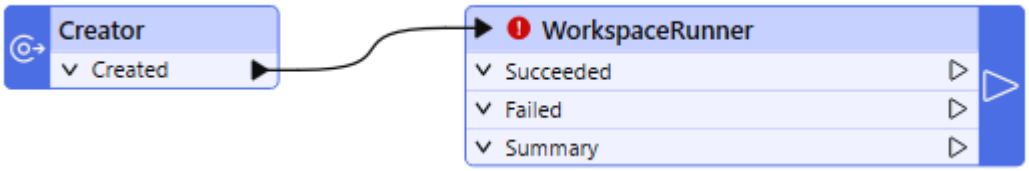
4.3.2 Create the Parent Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Blank Workspace.

Add a Creator transformer to the canvas.

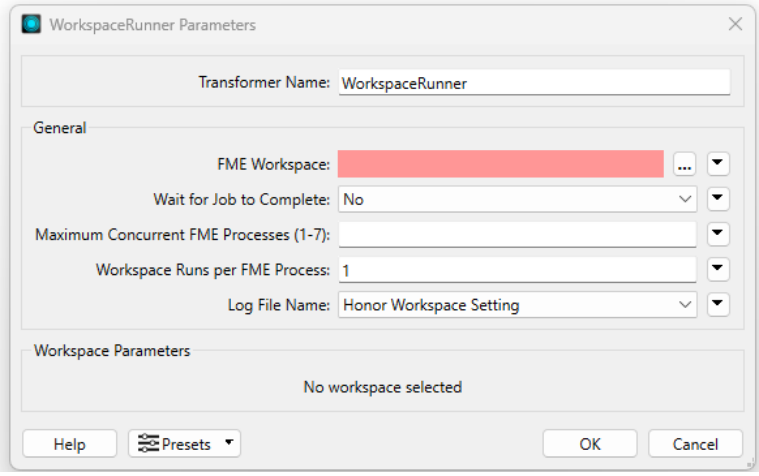
The Creator transformer creates features and sends them into the workspace for processing. You can select the desired number of features and their geometry type. In this instance we are using the transformer to generate a single feature which will then trigger the rest of our workflow.
Leave the Creator transformer's parameters as default.

Add a WorkspaceRunner transformer to the canvas and connect it to the Creator.



4.3.3 WorkspaceRunner general configuration

Click on the transformer's parameters cog (or double-click on the transformer titlebar).



In the General parameters section we need to specify the child workspace that is to be initiated and whether or not to *Wait for Job to Complete*.
If the Wait for Job to Complete parameter is set to Yes, then the transformer will wait until the workspace has finished running. In this case, the initiating feature is output via the Succeeded port if the job successfully ran to completion.
The initiating feature will be output via the Failed port if the workspace did not run to completion, and will have a *_failure_message* attribute added to it that contains the error message returned from the FME that ran the workspace.

If this parameter is set to No, the transformer will output the initiating feature as soon as an FME has been spawned off to do the translation. In this case, the initiating feature is output via the Succeeded port if the request was successfully submitted.

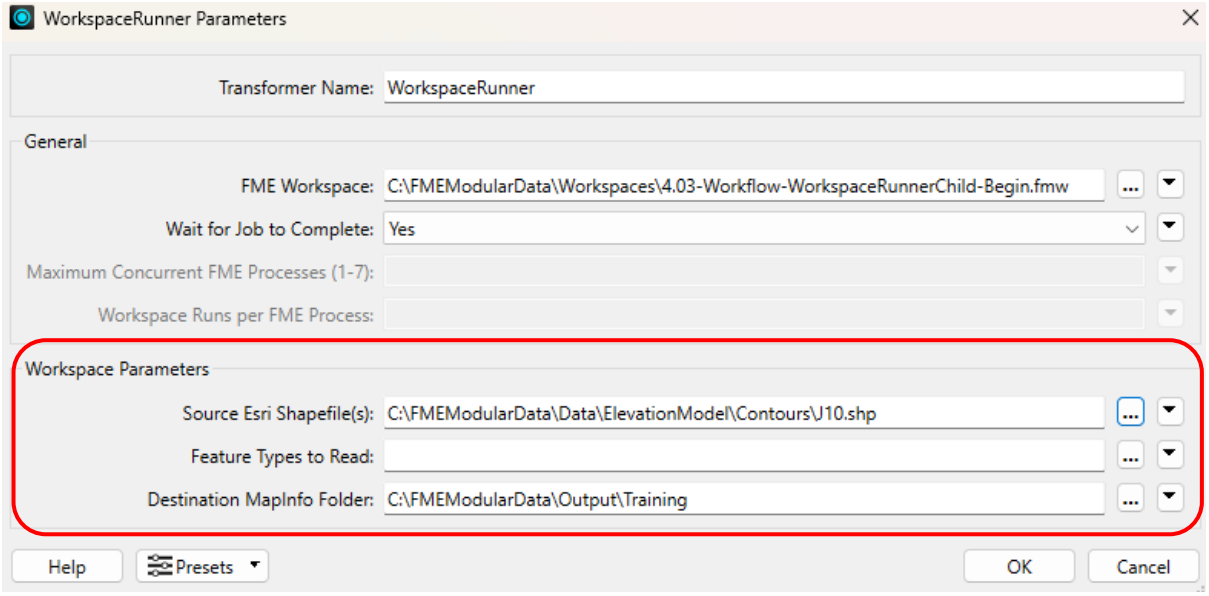
Why wait for workspace completion? Continuation of this workspace to perform further tasks; run another child workspace, send email notifications, capture details of the child workspace run results (success/failure).

Set the General parameters as follows:

FME Workspace:	*C:\FMEModularData\Workspaces\4.03-Workflow-WorkspaceRunnerChild-Begin.fmw*
Wait for Job to Complete:	*Yes*

4.3.4 Configure Workspace Parameters required by Child workspace

Next we need to configure the Workspace Parameters. This is the really important bit, and the focus of this exercise!



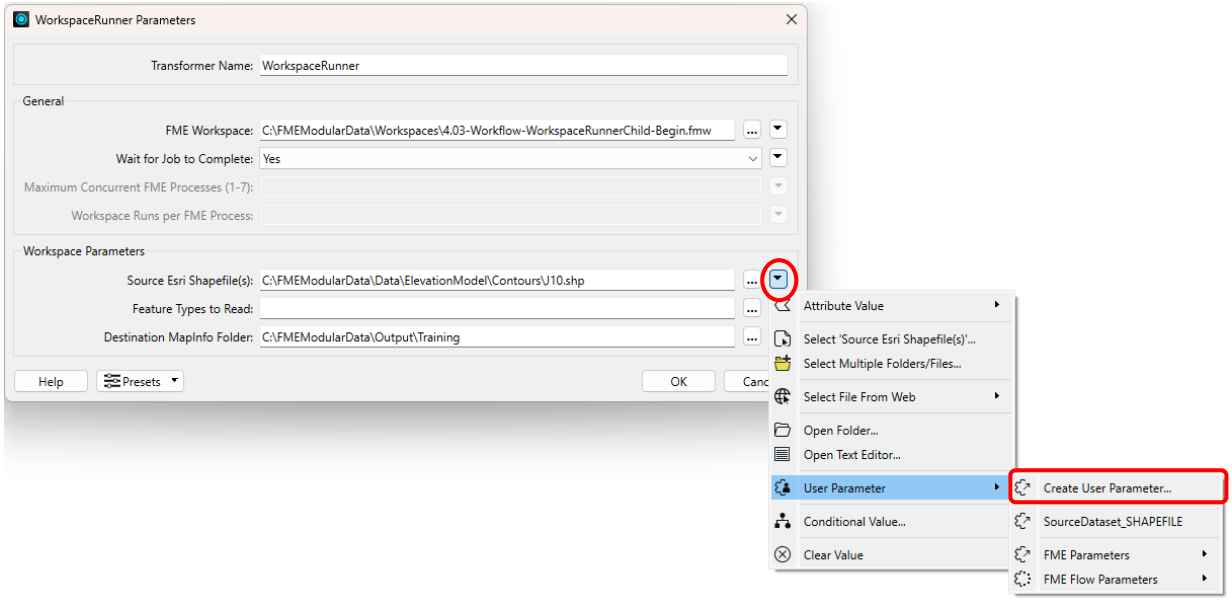
The scenario in this exercise is very simple, and actually we could do it with just a single workspace. However, the key to using the WorkspaceRunner is understanding how to pass parameters (details) from the parent workspace to the child workspace – where they are needed in order to perform the translation.

Parameters is a topic that we cover on our advanced training module, but we need to introduce the concept here in order to use the WorkspaceRunner. User Parameters are a way of enabling end-users, at the time of running the workspace, to define/pass values into the workflow that are needed by either readers/writers or transformers. Typically this could be defining the source dataset to read, choosing a destination/output folder, or setting a value used by a transformer (such as Buffer Distance within the Buffer Transformer).

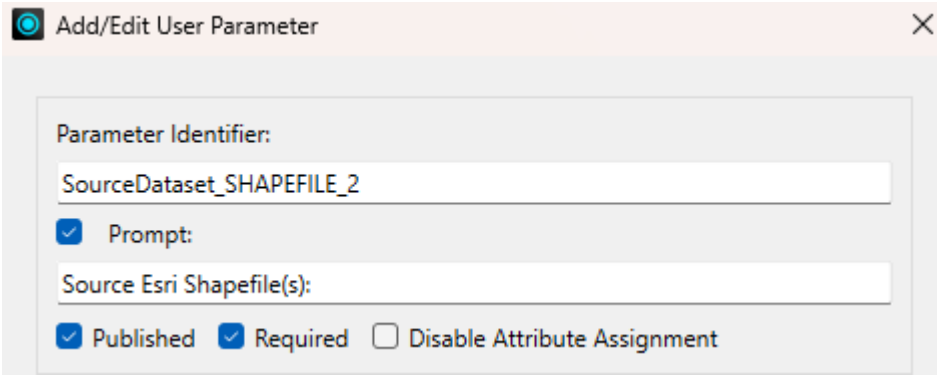
We're keeping it simple in our scenario; we just want the end-user to be able to select the Source ESRI Shapefiles that are to be processed, at the time of running the workspace. Then pass this information from the parent workspace to the child workspace.

Currently the Source ESRI Shapefiles input value is static and set to only read in J10.shp. However we want the end-user to be able to select which files to process at the time of running the parent workspace (as they will want to process different tiles each time). So, we need to convert the *Source ESRI Shapefile* value to become a User Parameter.

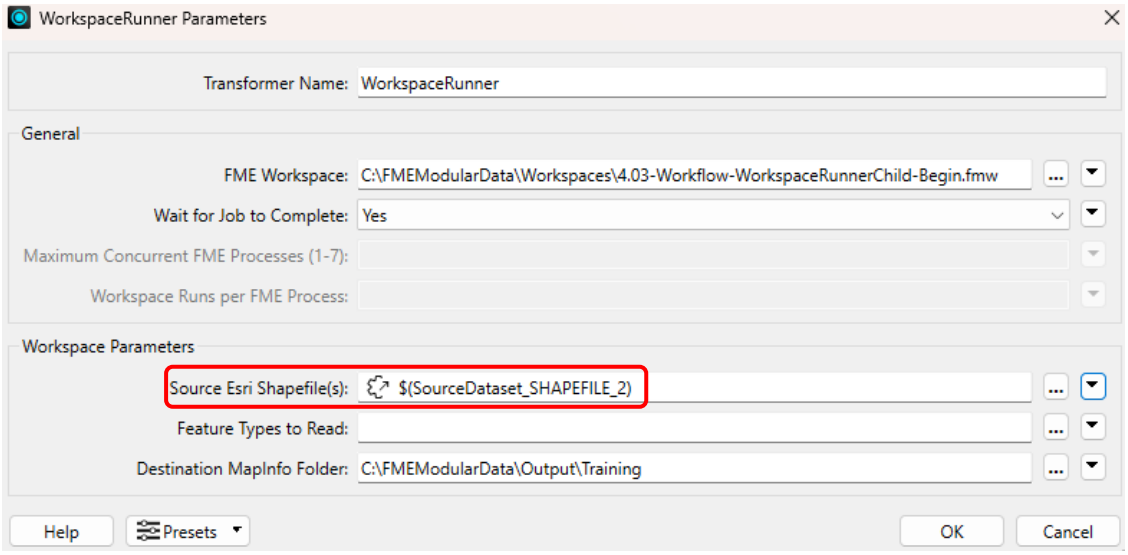
Use the Source ESRI Shapefiles menu arrow to select *User Parameter* > *Create User Parameter…*



The Add/Edit User Parameter dialog will display. Here you can set define the parameter Identifier, Prompt and Default Value. For this example we'll accept the defaults and click OK



Back in the Workspace Parameters dialog the *Source ESRI Shapefile* value has become a User Parameter (instead of a static value of J10.shp):



This means that when choosing to run this parent workspace, the end-user will be prompted to specify the required Shapefiles for processing.
The choose tile values will then be passed from the parent workspace to the child workspace.

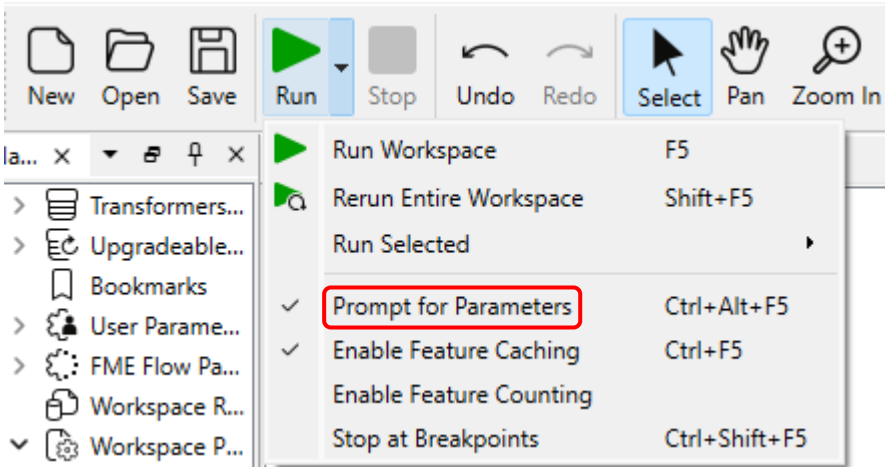
4.3.5 Save workspace

Click on the Save button, or use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces

Give your workspace a name, and Click Save

4.3.6 Make sure Prompt for User Parameters is activated

Before running the workspace it's important that end-users are prompted to enter any required parameters. In the menu-bar menu *Run* ensure that *Prompt for User Parameters* is ticked.



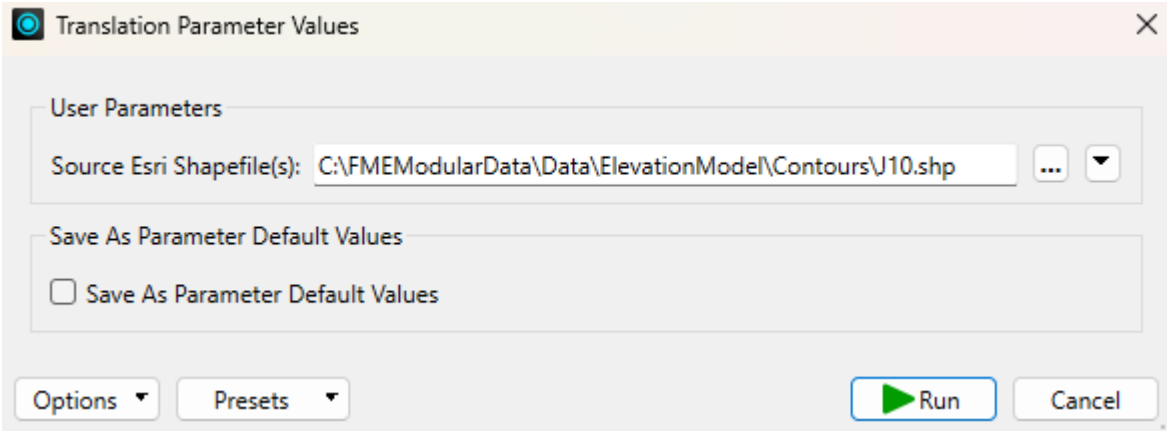
Upon running the parent workspace the user will now be prompted to modify any required parameters – such as source data or output destination.

4.3.7 Run Workspace

We are now ready to run the Parent workspace. The child workspace doesn't need to be open in Workbench.
Run the workspace by clicking the run button on the toolbar, or by using Run > Run Workspace on the menu bar.

The Translation Parameters dialog will display. Select one or multiple contour Shapefiles for processing.

Also, clear the *Save As User Parameter Default Values* tickbox – otherwise it will remember your choices and not offer the Prompt dialog next time you rerun the workspace. You can overcome this by using *Run > Rerun Entire Workspace* to run the workspace.



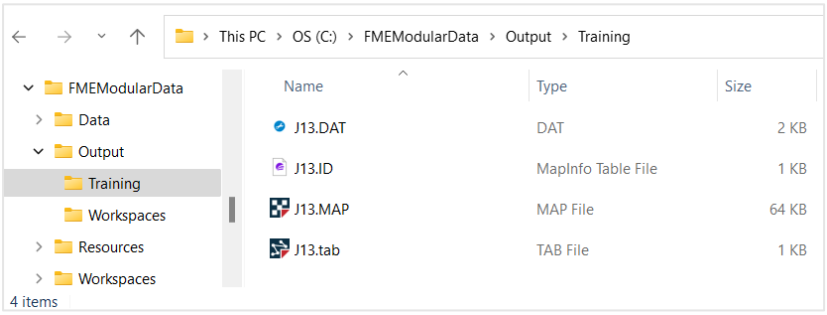
Click Run to proceed with the translation.

Upon completion of the child workspace, a feature will output from either the WorkspaceRunner Succeeded or Failed port. There will also be a single feature output from the Summary port.

The parent workspace translation will then also be complete (because we chose Yes to *Wait for Job to Complete* within the WorkspaceRunner parameters).

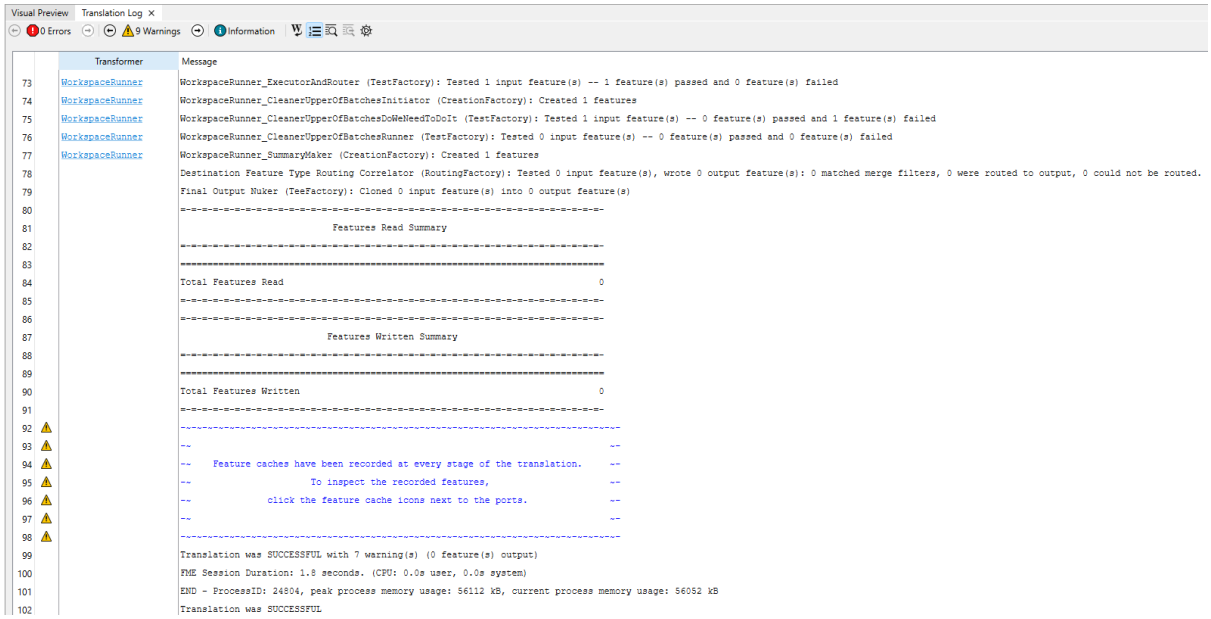
4.3.8 Examine Output Folder

Within File Explorer navigate to your output destination folder. Where you will find MapInfo TAB datasets have been created for each of your chosen input ESRI SHP contour tiles:



4.3.9 Translation Logs

Notice that the Translation Log window only details the parent workspace content/actions and that the Total Features Written value is 0.



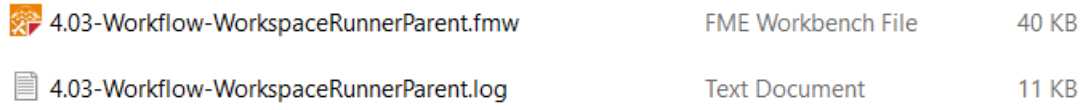
When using workspace chains, it's important to understand that multiple log files are generated and the content of each log file differs:

Parent workspace log – the content and actions of the parent workspace only. In this scenario: the Creator and WorkspaceRunner transformers. No data was actually written (output) by this workspace.

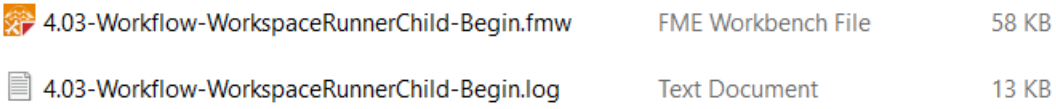
Child workspace log – this details the content and actions of the child workspace only; the input data read, processed and written. So, in our exercise its this log file to check to validate total features written and check for any warnings relating to the data processing and output.

Where are these log files?
By default the log file for each workspace will be found in the same directory as the workspace file (.fmw). These can be opened with any text editor applications:

C:\FMEModularData\Output\Workspaces *(or wherever you saved yours)*



C:\FMEModularData\Workspaces



Advanced Exercise

If you have time, also make the Destination folder for the MapInfo output configurable to the end-user at the time of running the parent workspace. Think about the prompt message, make it user-friendly.

Then re-run the workspace selecting different input contour tiles and specifying a new output folder.

FME Lizard

The WorkspaceRunner transformer can be useful in a number of scenarios, including:

- *chaining multiple workspaces to run in series or parallel*
- *workflows where different workspaces need to be triggered to run based on criteria (useful for self-serve or production environments)*
- *in conjunction with Emailers for translation status notifications*
- *Batch processing bulk data – for performance*

Many of the above workflow design techniques require intermediate-advanced FME skills to deploy (most of which are covered in our 'Advanced Workflow Design' training module). However, this exercise has introduced the fundamentals of using the WorkspaceRunner transformer.

Usage Notes

FME Desktop and FME Server

Publishing to FME Server: Publishing a workspace that includes the WorkspaceRunner transformer is not recommended. The transformer will try to start an FME outside of FME Server to run the workspace. When using FME Server for workspace chaining or email notifications, use FME Server Automations.

Only use the WorkspaceRunner transformer when running workspaces with FME Desktop.

Congratulations

By Completing this exercise you have learned how to:

- Chain two workspaces using the WorkspaceRunner transformer
- Pass parameters required by the child workspace from the parent workspace

5 Best Practice and Performance

5.1 Debugging a Workspace

Demonstrates	Checking the Log window for errors and warnings Debugging through use of Feature Counts and Visual Preview
Overall Goal	Add to the workspace by creating a 'walkability' score and debug any issues
Data	Addresses (ESRI Geodatabase), Crime Data (CSV), Parks (MapInfo TAB), Swimming Pools (OSM)
Start Workspace	C:\FMEModularData\Workspaces\5.01-BestPractice-Debugging-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\5.01-BestPractice-Debugging-Complete.fmw

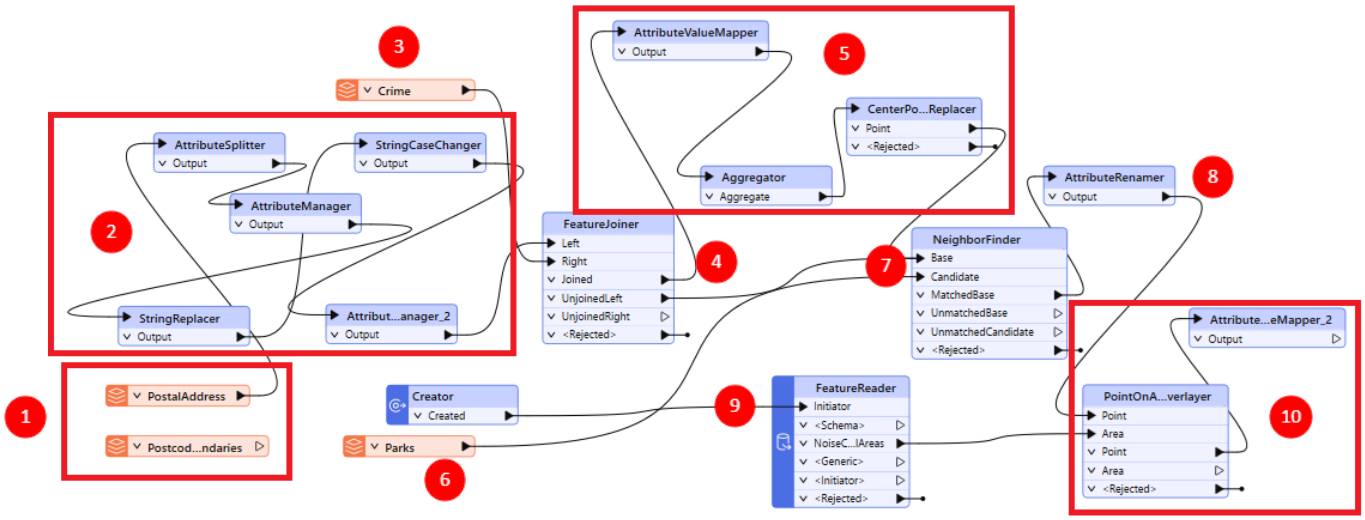
You have just taken over a project from your colleague and they've passed their workspace on to you. This project is to calculate the "walkability" of each address in the city of Vancouver. Walkability is a measure of how easy it is to access local facilities on foot.
Currently the workspace calculates a number of metrics but not walkability – lets add that in.

5.1.1 Launch FME Workbench and Open Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open Workspace. Navigate to and open C:\FMEModularData\Workspaces\5.01-BestPractice-Debugging-Begin.fmw

Then run the workspace to cache the data.

This workspace is a bit messy, but we will fix that in a later exercise. First, let's figure out what this workspace does:

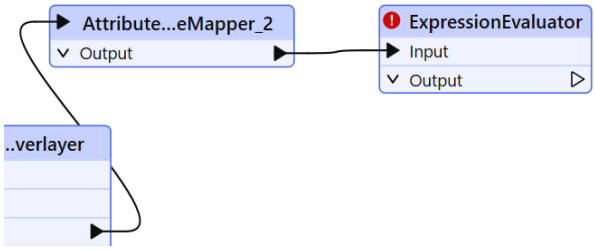


1. Reading Addresses.gdb creates PostalAddress and PostcodeBoundaries feature types.
2. Transformers clean attributes from the PostalAddress feature type and create a separate Number and Street attribute. Then the last two digits of the Number are being replaced by XX to create an attribute that will be the Join Key for joining the crime data.
3. Reading crime.csv creates the Crime feature type. Substituting XX for the last two digits anonymizes the street number for each crime incident.
4. The FeatureJoiner joins PostalAddress and Crime based on the Join Key attribute created in 2 and the Block attribute from Crime.
5. Transformers set the crime Type attribute to a number based on severity and then calculate the total CrimeValue for each address block. The CenterPointReplacer ensures only one point exists if there are multiple crime incidents in the same location.
6. Reading Parks.tab creates the Parks feature type. This data will let us measure the walking distance from addresses to parks.
7. Using the NeighborFinder, the park closest to each address is determined.
8. The NeighborFinder creates the _distance attribute. The AttributeRenamer renames it to ParkDistance.
9. The Creator and FeatureReader are used to read the Planning Restrictions OGC Geopackage. Then from that dataset, the NoiseControlAreas feature type is used to obtain the noise restriction areas.
10. The PointOnAreaOverlayer joins the point data containing the crime, distance to park, and addresses with the NoiseControlAreas polygons. The merged data assigns the noise restrictions to any overlapping points. The AttributeValueMapper creates the attribute NoiseZoneScore, giving a score to each point based on its zone. This new attribute reflects that addresses in noise-restricted areas are more walkable.

This workspace is very messy as our colleague didn't follow any best practices. We will be cleaning it up in a later exercise!

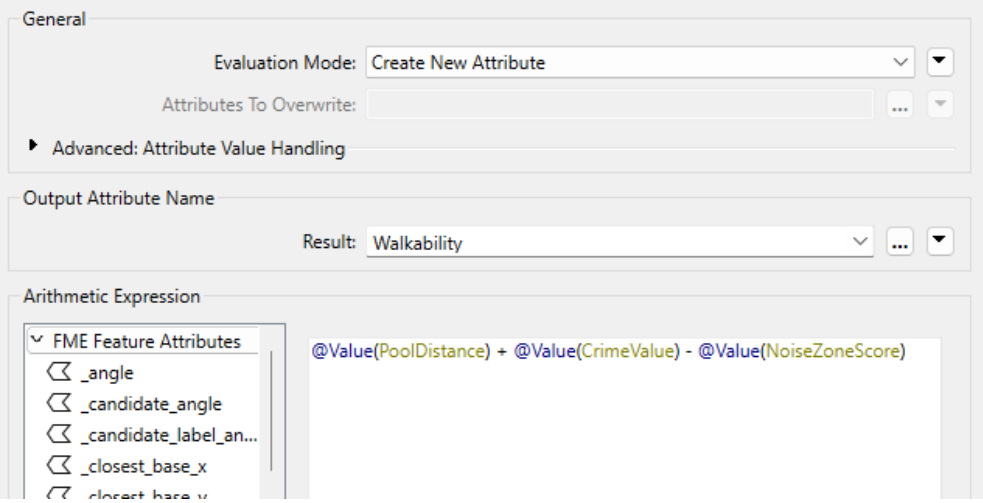
5.1.2 Add an ExpressionEvaluator transformer

We will create a measure of walkability that combines all of our current values using the ExpressionEvaluator transformer. So add an ExpressionEvaluator transformer to the end of the workspace and connect it to the AttributeValueMapper_2.



Inspect its parameters. Set it up to create a new attribute called Walkability that is:

```
@Value(ParkDistance) + @Value(CrimeValue) - @Value(NoiseZoneScore)
```



With this expression, the smaller the result, the better.

5.1.3 Save and Run the Workspace

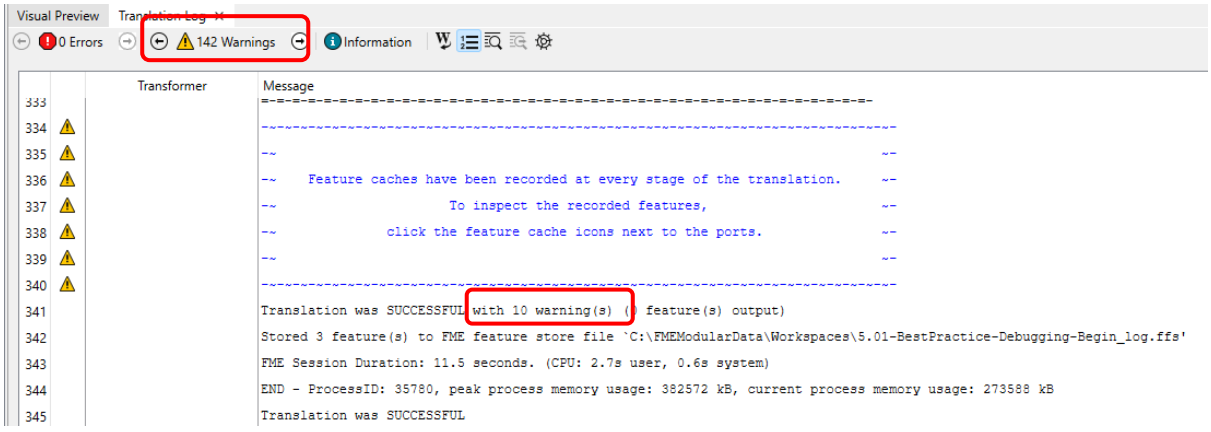
Use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.

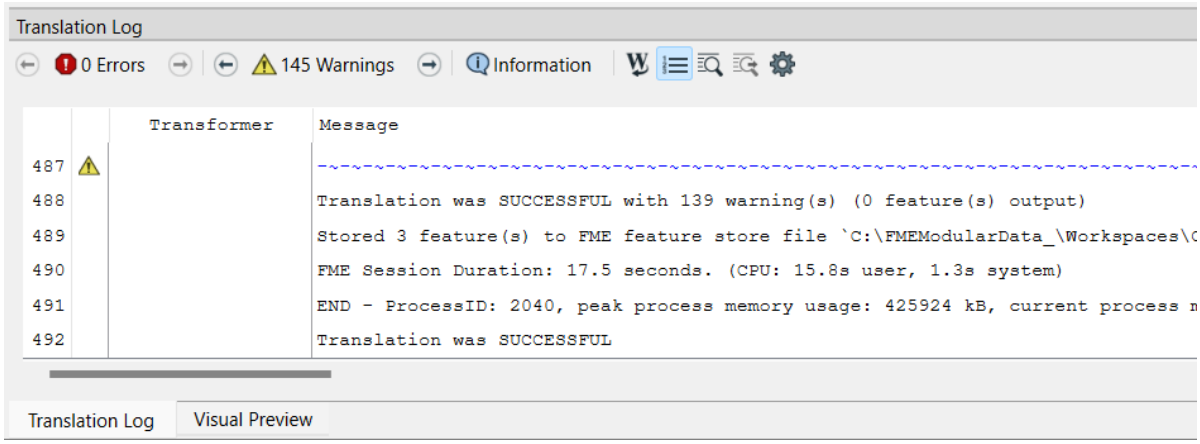
Then Run the workspace.

5.1.4 Examine the Translation Log

Let's assess whether the result of the translation is correct.
Firstly, check the log window for errors and warnings.

There are no errors, but there are several warnings, which is not a good sign:





Note: *The number of warnings showed in the Translation Log may be different, this is based on the Logging Parameters set in FME Options.*

Click on the warnings button to filter out the warnings.
There are some warning lines relating to running the workspace with Feature Cache turned on – *we can ignore those*.

We can then see warning messages relating to the output of the ExpressionEvaluator transformer:

234	⚠	ExpressionEvaluator	Null, missing, or empty string operand was found in expression '@real64(319.9256214903732 + <null> - 0)'. Result is set to null
235	⚠	ExpressionEvaluator	++
236	⚠	ExpressionEvaluator	Feature Type: `ExpressionEvaluator_OUTPUT'

5.1.5 Inspect the ExpressionEvaluator Feature Cache

Click on the feature cache of the ExpressionEvaluator to examine the features at this point in the workflow.

Some addresses do indeed have a Walkability value of <null>:

NoiseZoneScore	Walkability
0	<null>
200	<null>
0	<null>
0	<null>
0	<null>
0	<null>
0	<null>

So, we know there is a problem, let's try and figure out where the problem is and why it occurs.

5.1.6 Locate the Problem

We can tell the warning comes from the ExpressionEvaluator, but that doesn't necessarily mean that is where the problem lies.

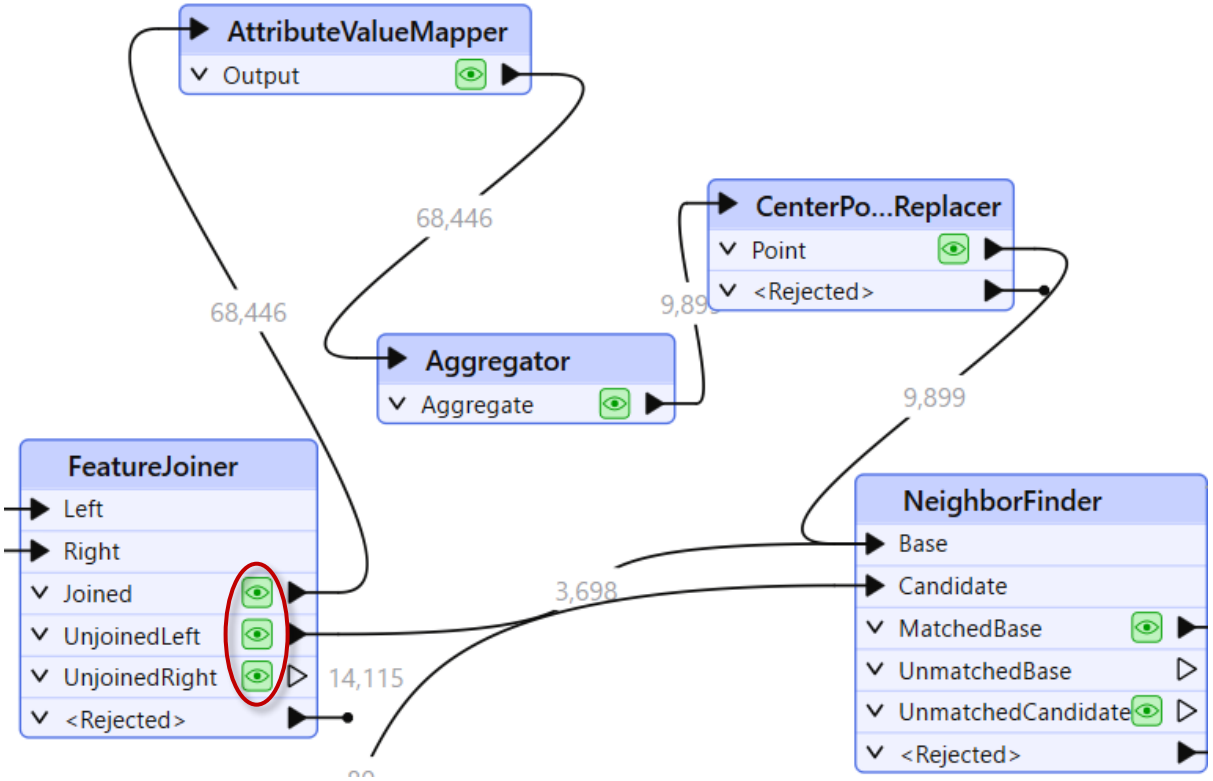
The ExpressionEvaluator calculation requires three values to perform the calculation:

```
ParkDistance + CrimeValue - NoiseZoneScore
```

Continue examining the feature cache of the ExpressionEvaluator within the Visual Preview Table view. Right-click on *CrimeValue* attribute and sort by ascending numeric order. That will put any null values to the top of the table.

The calculation is failing because some features are missing the *CrimeValue*

Let's find out where it becomes an issue. First, organize the workspace a bit, and then inspect the caches on the FeatureJoiner transformer.
We are inspecting the FeatureJoiner because that's where we first get our Crime data:



There are no <null> values coming from the FeatureJoiner, so let's move along the translation. Check the cache for the AttributeValueMapper. That's where values are set, so perhaps nulls are coming out of there?

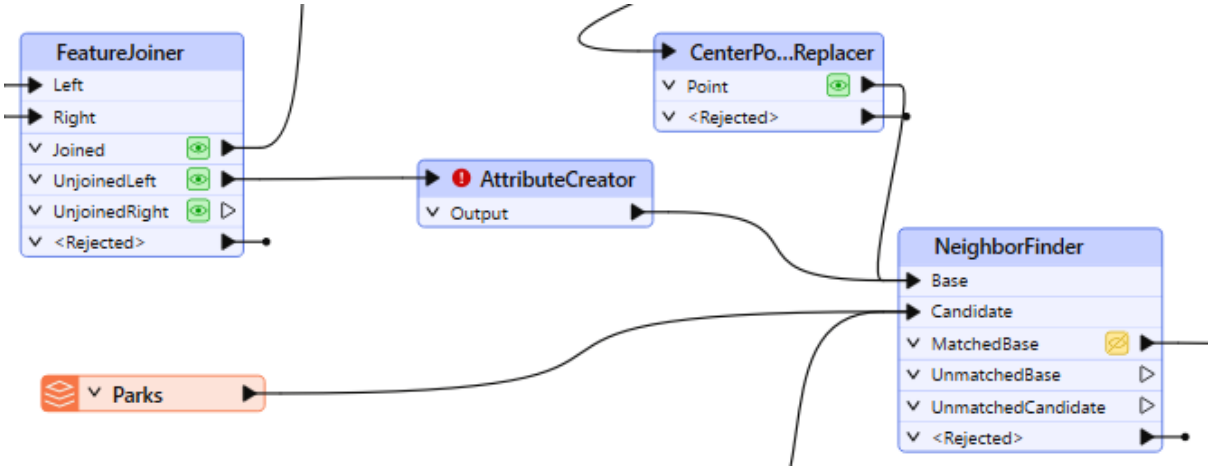
On inspection, there are no <null> values for the CrimeValue or the crime type attribute in there. There are also no nulls for the Aggregator and CenterPointReplacer caches.

Checking each feature cache is a bit time consuming, let's try a different method. Check the feature counts on each connection. There are 68,446 features tagged with a crime (FeatureJoiner:Joined), but then that is reduced to 9,899 after the Aggregator and then there are 3,698 features that are not tagged with a crime (FeatureJoiner:UnjoinedLeft). That gives a total of 13,597, coming out of the NeighborFinder, which is correct.

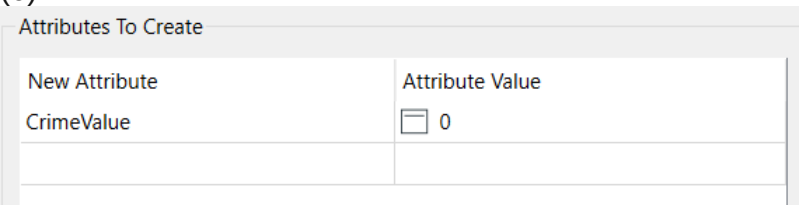
Oh. Do you see it yet? The 3,698 features that are not tagged with a crime: what CrimeValue do they get? Inspect the UnjoinedLeft output from the FeatureJoiner, and you will see that they do not have the CrimeValue attribute. That's why the ExpressionEvaluator says that there are nulls. The reason these features do not have a value for CrimeValue, is because they are not being routed through the AttributeValueMapper which assigns the CrimeValue a value.

5.1.7 Fix the Problem

If those features do not have a CrimeValue attribute, then we should give them one. To do so, add an AttributeCreator transformer to the workspace between the FeatureJoiner:UnjoinedLeft output port and the NeighborFinder:Base input port:



Open up its parameters and create an attribute called CrimeValue with a value of zero (0).



5.1.8 Resave and Rerun the Workspace

Resave and rerun the workspace.
You should find no warnings relating to the ExpressionEvaluator and the *Walkability* attribute contains no *<null>* values.

We have successfully debugged the workspace and resolved the problem.

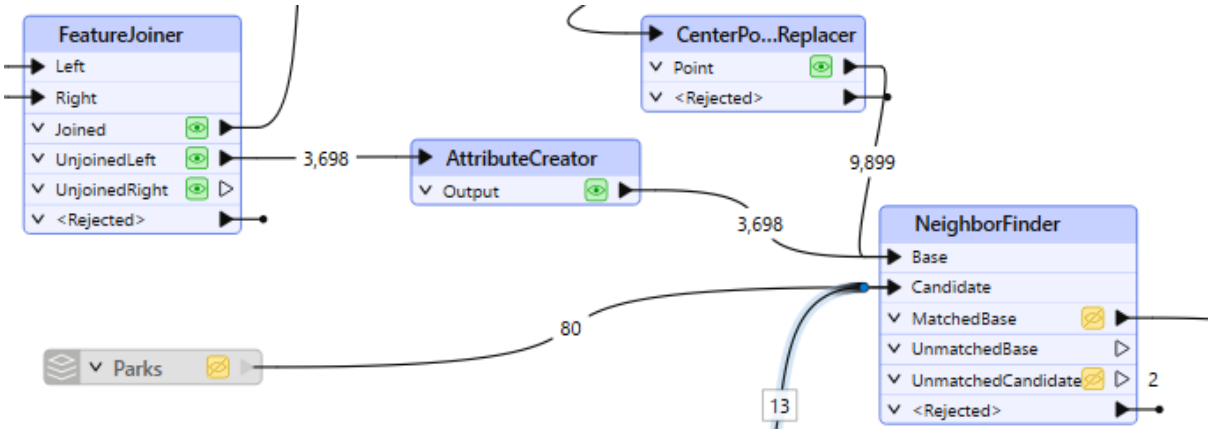
Now we have been asked to revise the approach; the city has decided that parks are not a great candidate for walkability scores because there is usually a park nearby. They have decided to evaluate how easy it is to walk to a swimming pool.

We can reuse the same workflow for swimming pools that we used for parks, with just a few minor updates.

5.1.9 Disable the Parks Reader feature type

Click on the Parks reader feature type and select the disable button.

This will cause the data links to break until re-run or the reader is enabled again.

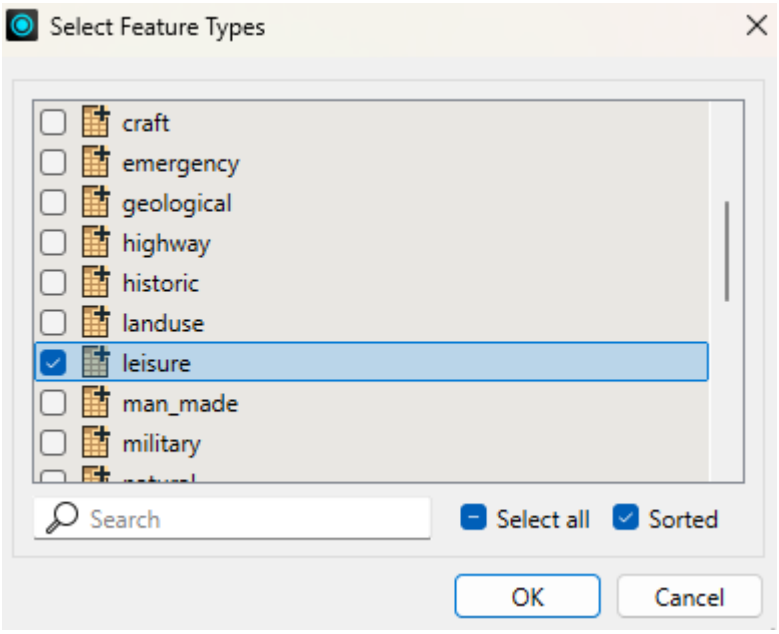


5.1.10 Add Reader

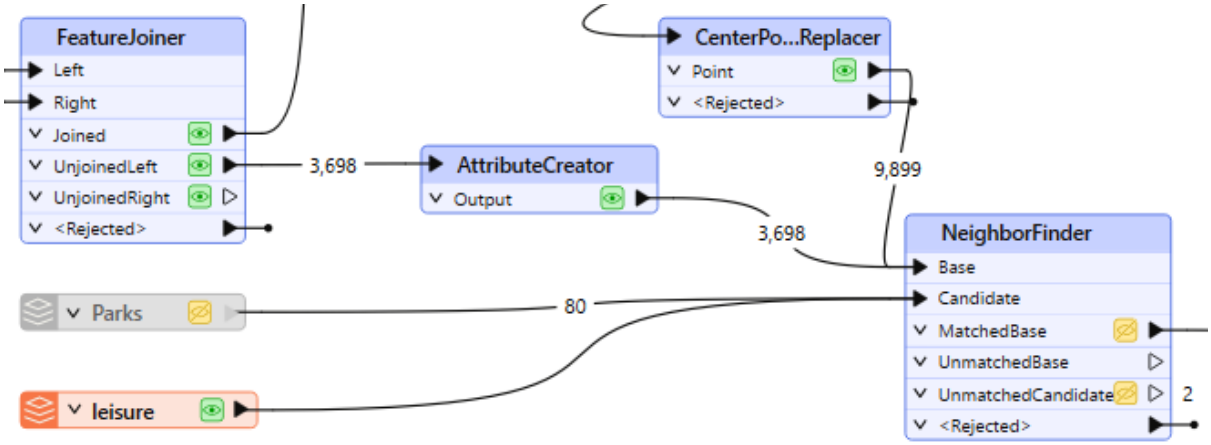
Use either the Reader button, or use Readers > Add Reader… from the menu bar. The Add Reader dialog will open, in which define the Format and Dataset settings as follows:

Reader Format	OpenStreetMap (OSM) XML
Reader Dataset	C:\FMEModularData\Data\OpenStreetMap\leisure.osm

When prompted, select only the *leisure* feature type:



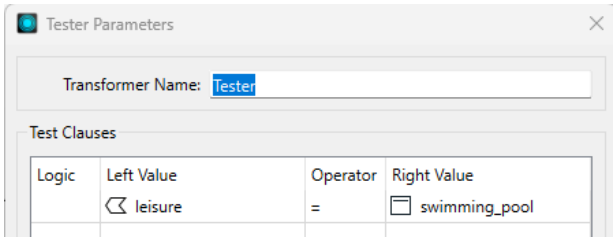
Then connect the new *leisure* feature type to the NeighborFinder:Candidate input port.

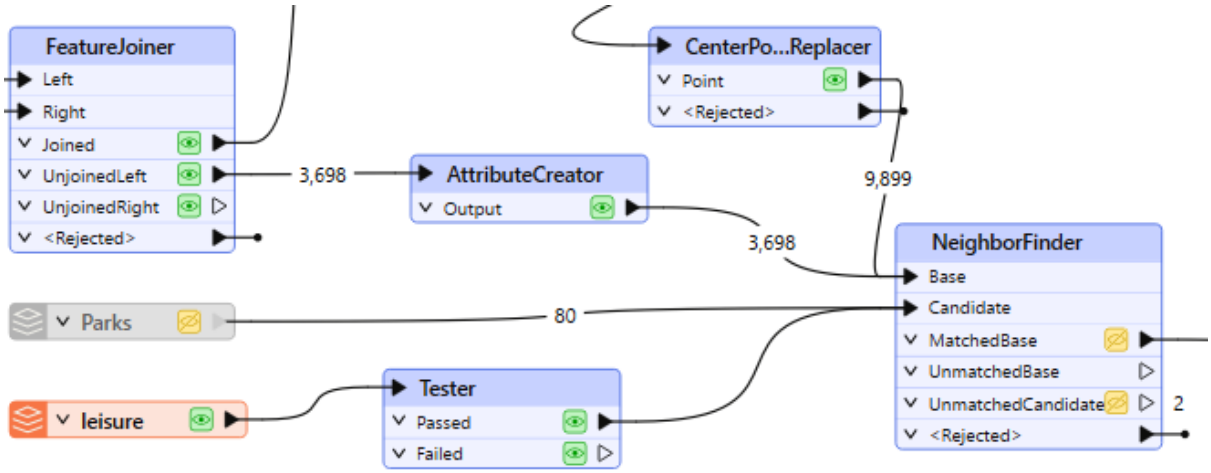


5.1.11 Filter Leisure Data

If you inspect the leisure data, you'll notice that there are various types of leisure facility, the type being recorded in the leisure attribute.

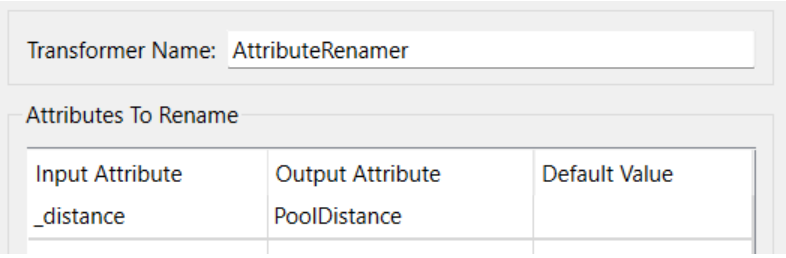
So, add a Tester transformer between the leisure feature type and the NeighborFinder. Set up the parameters to test for *leisure* = *swimming_pool*





5.1.12 Modify the AttributeRenamer and Walkability Calculation

Now update AttributeRenamer to be *PoolDistance* instead of ParkDistance.



The renaming of this attribute will cause the ExpressionEvaluator to turn red.

To fix the ExpressionEvaluator, open the parameters and change *@Value(ParkDistance)* to *@Value(PoolDistance)* to take account of the new PoolDistance attribute:

```
@Value(PoolDistance) + @Value(CrimeValue) - @Value(NoiseZoneScore)
```

5.1.13 Resave and Rerun the Workspace

Resave and rerun the workspace.

5.1.14 Examine the Log and Features

Check the log for warnings and errors, and then inspect the ExpressionEvaluator cache.
Notice that the Walkability scores are exceedingly large all of a sudden, due to the PoolDistance. Something is wrong, but what?

5.1.15 Locate Problem

The PoolDistance is the source of the problem. There is no related log message to give a clue, and the Feature Count numbers look correct.

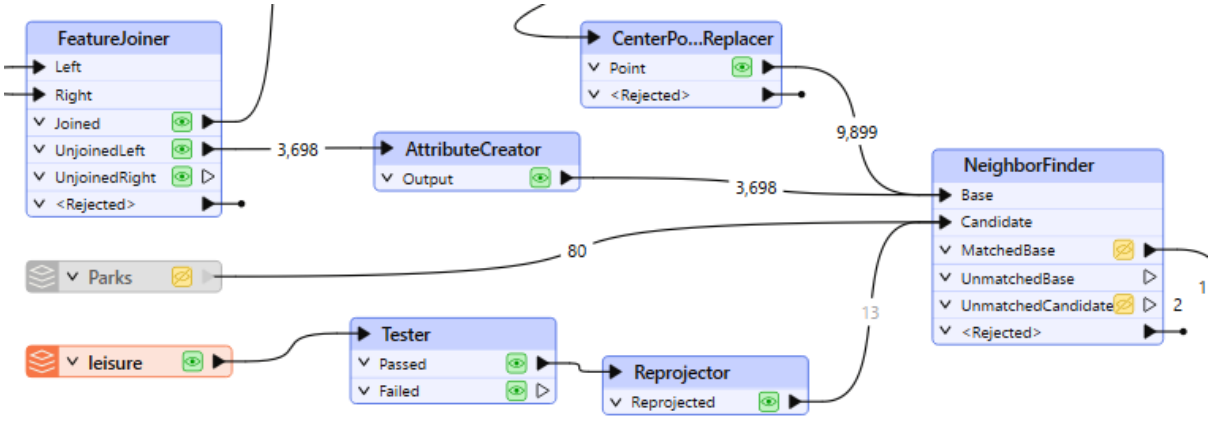
Let's inspect the data. This result is typical of a mismatch of coordinate systems.

Click on some features and select the Feature Information button. In this window you will see that the main data has a coordinate system of UTM83-10, while the leisure data from OSM has a coordinate system of LL-WGS84.

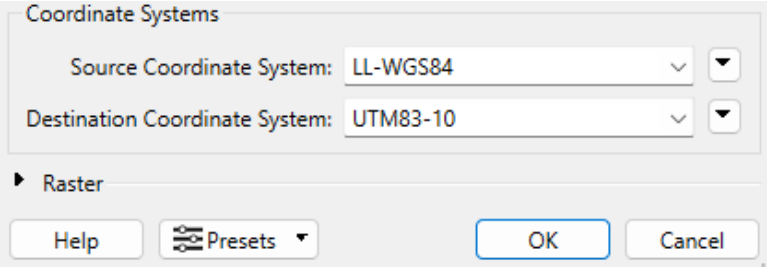
This disparity is why the "nearest" pool to each address is such a high distance.

5.1.16 Fix Coordinate System Problem

The obvious solution is to reproject the pools to the correct coordinate system. So, add a Reprojector transformer to reproject the leisure data before it gets to the NeighborFinder:



Inspect its parameters and set it up to reproject from LL-WGS84 to UTM83-10.



Re-run the appropriate parts of the workspace. Check the log window and inspect the ExpressionEvaluator cache.

Walkability
1693.61768632...
591.081471273...
780.180449185...
579.229664974...
806.148857854...
3446.52999143...
852.414545016...
867.089680156...

Each address now has a Walkability score account for pools instead of parks, with a lower number being better and a higher number worse.

Congratulations

By Completing this exercise you have learned how to:

- check the Log window for errors and warnings
- Debug a workspace through use of Feature Counts and Visual Preview

5.2 Methodology - Performance

Demonstrates	Improve workspace performance by removing unnecessary attributes and Lists. Use Collapsing Bookmarks to prevent excess caching
Overall Goal	Improve the performance of a workspace
Data	Addresses (ESRI Geodatabase), Crime Data (CSV), Parks (MapInfo TAB), Swimming Pools (OSM)
Start Workspace	C:\FMEModularData\Workspaces\5.02-BestPractice-Performance-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\5.02-BestPractice-Performance-Complete.fmw

You have taken over a project from your colleague and they've passed their workspace on to you. You have already modified the workspace to calculate a "walkability" score of each address in the city of Vancouver. Now you want to review the workspace design methodology to improve the workspace performance.

5.2.1 Launch FME Workbench and Open Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open Workspace. Navigate to and open C:\FMEModularData\Workspaces\5.02-BestPractice-Performance-Begin.fmw
Alternatively, use your workspace from the previous exercise.

Then run the workspace to cache the data.

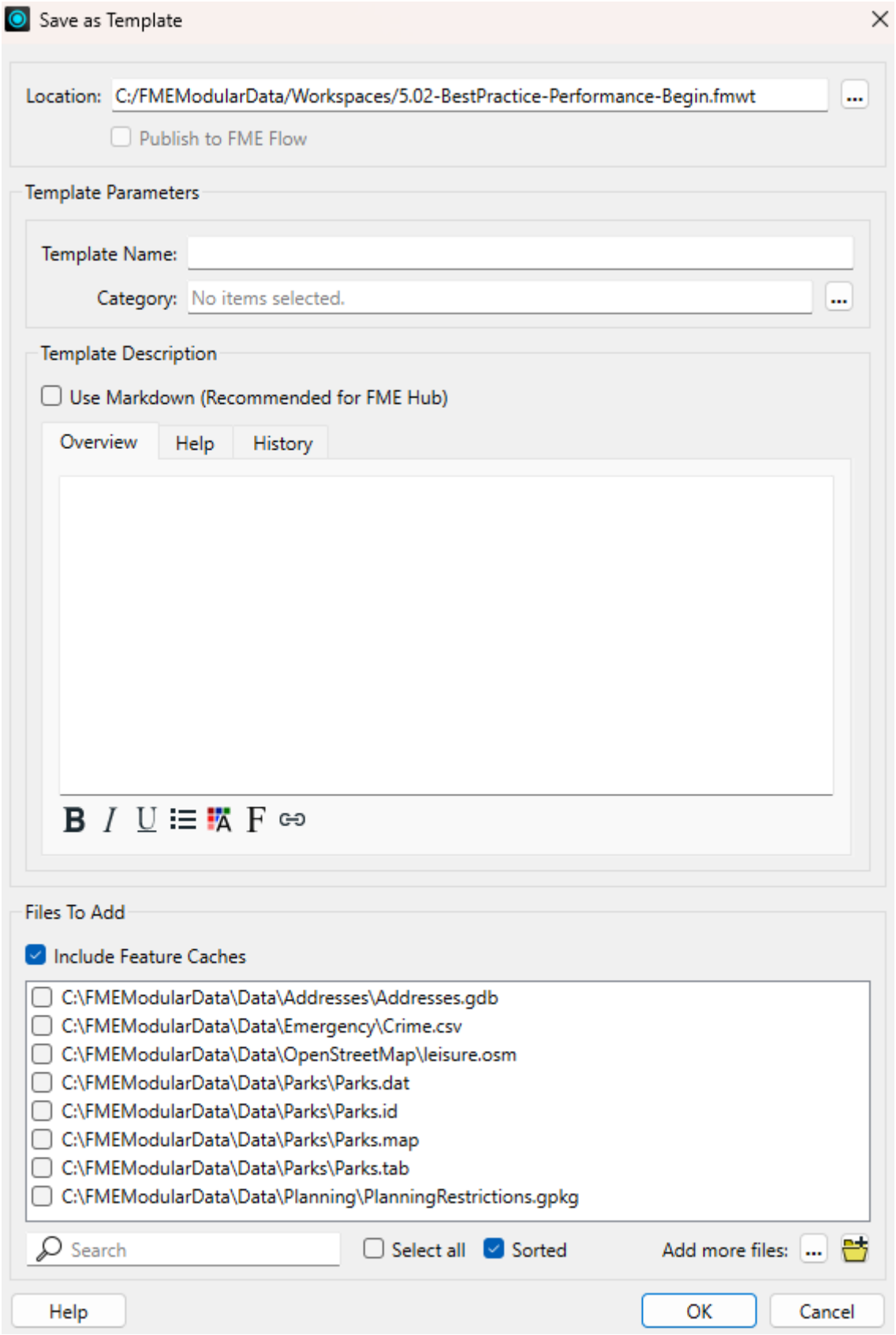
Note: If using your own workspace the layout may look different to the screenshots in this exercise. So, pay close attention to transformer and port names.

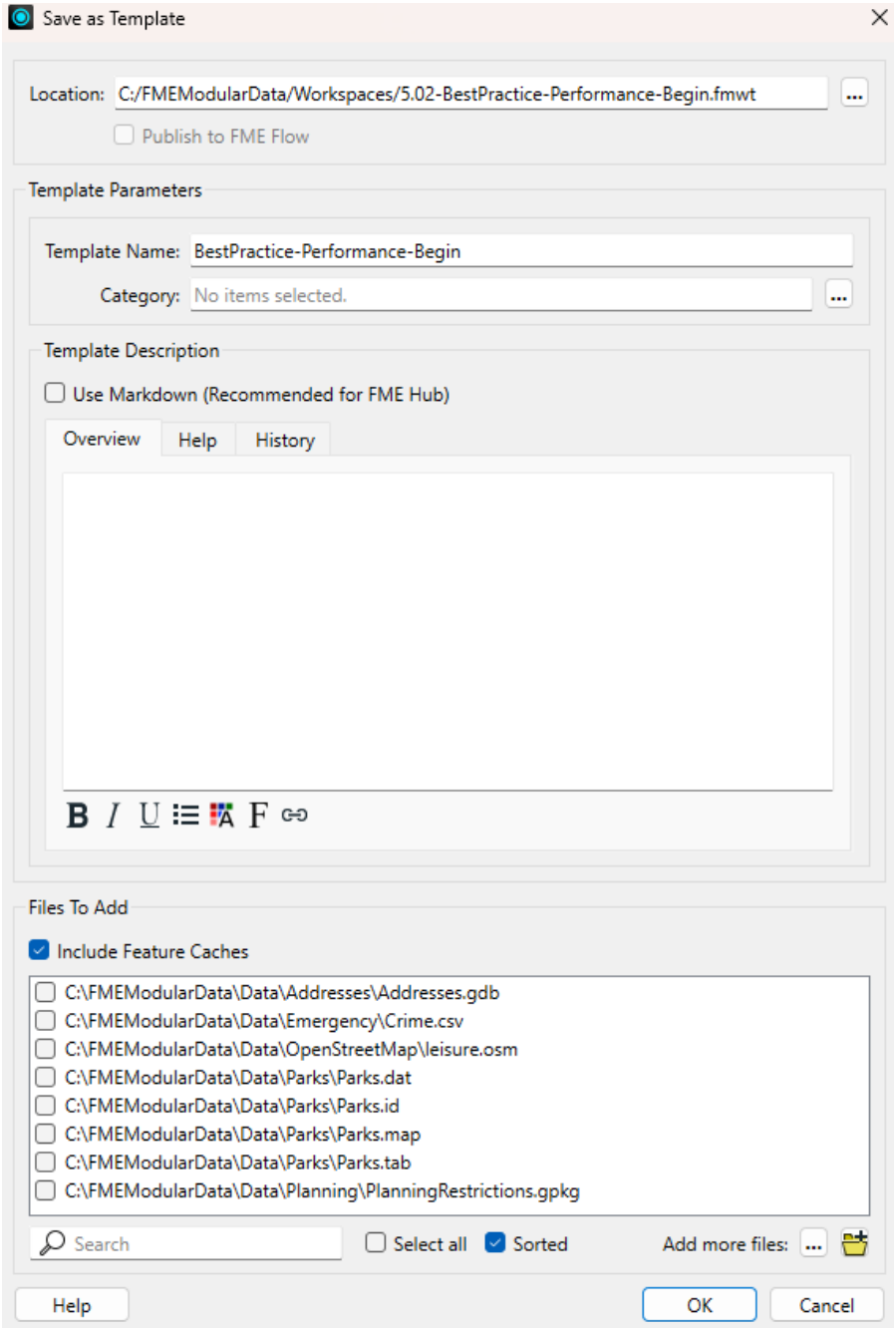
5.2.2 Determine Performance Improvements

While editing the ExpressionEvaluator in the previous exercise, you might have noticed there was a lot of additional attributes like *CrimeList{}.City* or *CrimeList{}.Block*. These excess attributes clutter the display and inspecting the output becomes hard. These attributes can hardly be helping the performance of the workspace either - even if that's mitigated by using caches during development.

Let's save the workspace as a template file. In the top menu go to *File > Save As Template*

When prompted:
save the template file to Location: *C:\FMEModularData\Output\Workspaces*
be sure to have the *Include Feature Caches* option checked
Deselect the datasets. (they aren't needed in this exercise, but this is useful if you need to 'package' the workspace to share with others):


Save as Template
Location: C:/FMEModularData/Workspaces/5.02-BestPractice-Performance-Begin.fmwt
Publish to FME Flow
Template Parameters
Template Name:
Category: No items selected.
Template Description
Use Markdown (Recommended for FME Hub)
Overview
Help
History
Files To Add
Include Feature Caches
C:\FMEModularData\Data\Addresses\Addresses.gdb
C:\FMEModularData\Data\Emergency\Crime.csv
C:\FMEModularData\Data\OpenStreetMap\leisure.osm
C:\FMEModularData\Data\Parks\Parks.dat
C:\FMEModularData\Data\Parks\Parks.id
C:\FMEModularData\Data\Parks\Parks.map
C:\FMEModularData\Data\Parks\Parks.tab
C:\FMEModularData\Data\Planning\PlanningRestrictions.gpkg
Search
Select all
Sorted
Add more files:
Help
OK
Cancel

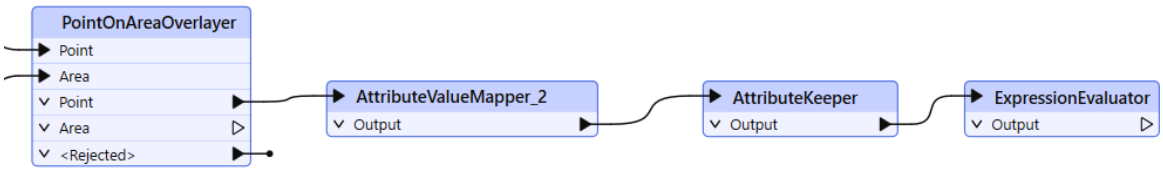


Click OK to proceed

We will come back to this template file later.

5.2.3 Remove Unnecessary Attributes

One aspect of data is the number of attributes and lists. Since there are a lot of additional attributes to remove but only a few we need to keep, we will use the AttributeKeeper transformer.
Place the AttributeKeeper between the AttributeValueMapper_2 and the ExpressionEvaluator transformers:



Inspect the AttributeKeeper parameters and set them up to keep only *CrimeValue*, *NoiseZoneScore*, and *PoolDistance*.

Take note of the names of the attributes that we are not keeping. We might be able to remove them earlier in the workspace.

5.2.4 Remove Lists

One attribute of interest is a list attribute called *CrimeList{}*, which doesn't appear necessary for any part of this translation.

Track down its source by pressing *Ctrl+F* and search for CrimeList. The search results show up in the Navigator window, and there you will find the Aggregator transformer is creating CrimeList.

Check the parameters for the Aggregator transformer and turn off the Generate List parameter, to prevent the list from being created. This step will cause many caches to become stale, but we will re-run the workspace shortly to solve this.

5.2.5 Remove Extra Feature Types

Another reason a workspace is running slowly is if you are reading in extra data that is not being used in the workspace. It looks like the original author read in the *PostcodeBoundaries* feature type from the Addresses.gdb. Additionally, we didn't remove the *Parks* feature type once we were done with it.

Delete both of those now and click Yes on any warnings that pop up.

5.2.6 Save and Run the Workspace

Use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces
Give your workspace a name. Click Save.

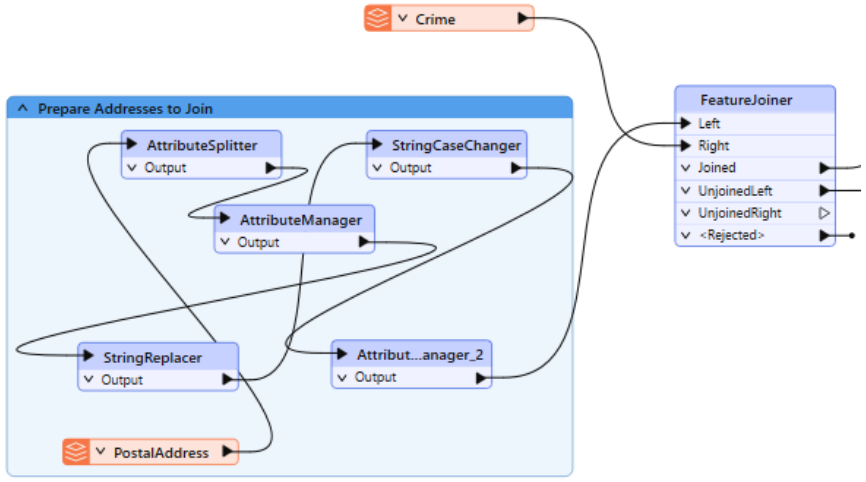
Then Run the workspace - to update the feature caches.

5.2.7 Add a Bookmark

Add a bookmark around all of the transformers between the PostalAddress reader and the FeatureJoiner, by selecting all the transformers and then either clicking the

Bookmark toolbar button, or pressing Ctrl+B on your keyboard.

Then name the bookmark *Prepare Addresses to Join*:

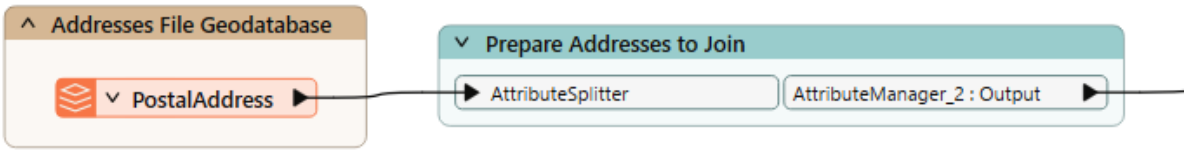


Note: *Bookmarks will be covered in greater detail later on in this module.*

5.2.8 Collapse the Bookmark

Another source of excess caching are transformers producing output that we don't need to inspect. These can be prevented by hiding these transformers within a collapsed bookmark.

Now we will collapse the bookmark and then when we re-run the translation only the last transformer will have a cache.
To collapse the bookmark, click on the arrow beside the bookmark name:



5.2.9 Save and re-run the Workspace

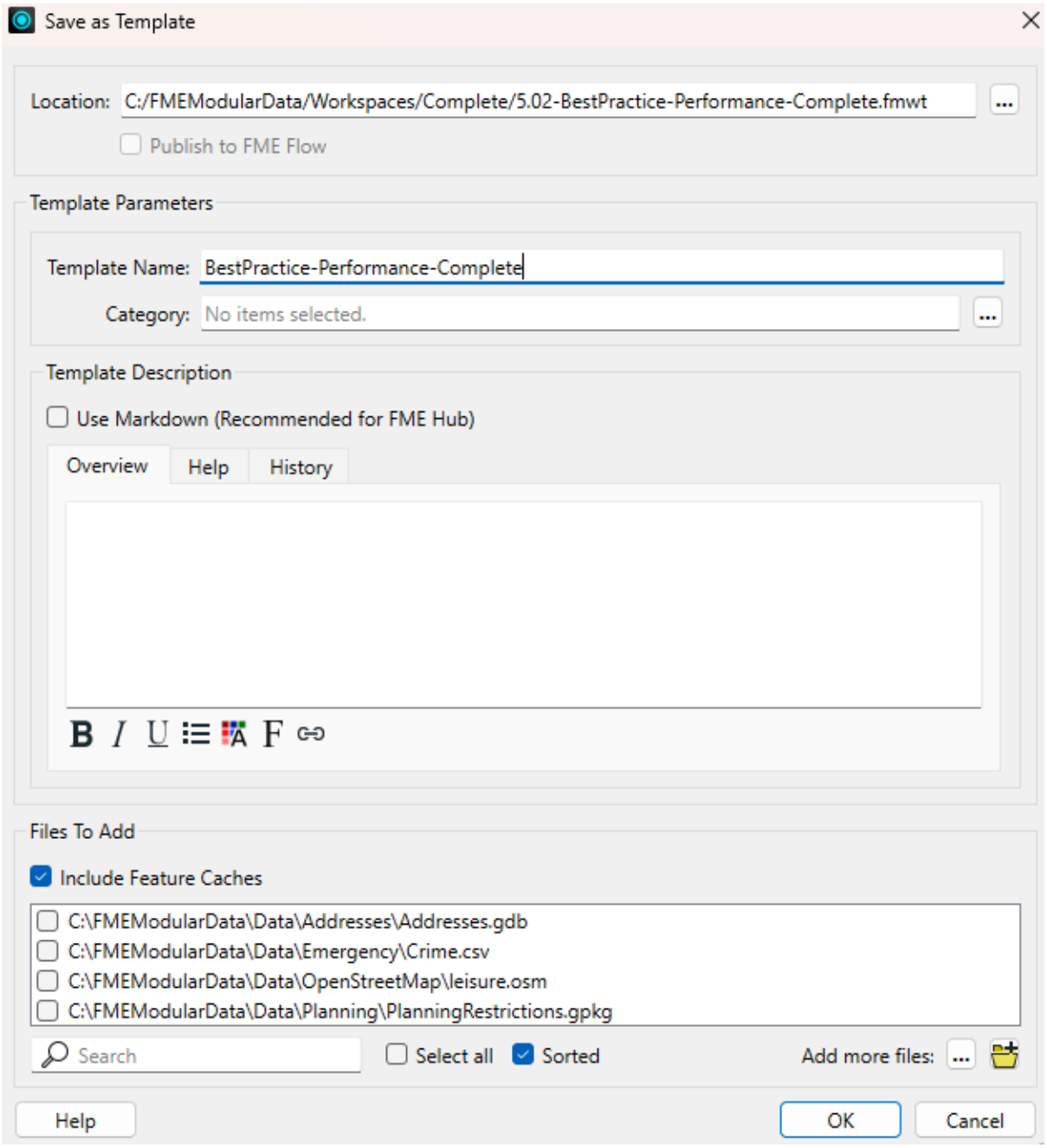
Use Save on the menu bar, then rerun the workspace.

The workspace will run and data will be cached, but for the collapsed bookmark, only one cache will be created for its five transformers. Attributes unnecessary to the output will also be removed by the AttributeKeeper.

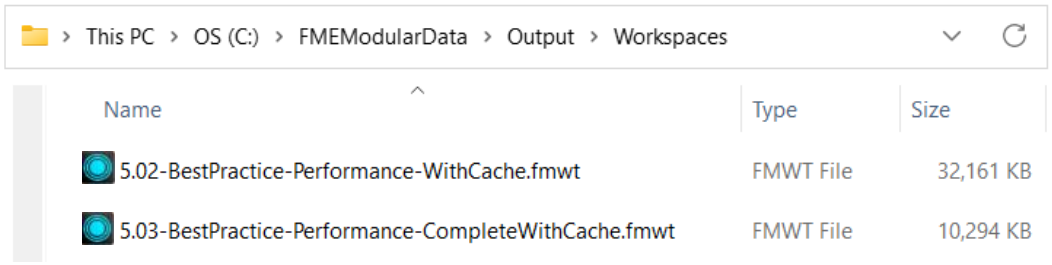
5.2.10 Create a New Template

Use *File > Save As Template* to create a new template of our revised workspace and feature cache.

When prompted:
save the template file to Location: *C:\FMEModularData\Output\Workspaces*
be sure to have the *Include Feature Caches* option checked
Deselect the datasets



Within File Explorer navigate to where both Template files were saved:



Compare the file size of the new template to the original, it should be considerably smaller.

Congratulations

By Completing this exercise you have learned how to:

- Remove unnecessary attributes to improve performance
- Track down unnecessary Lists and remove them
- Delete unused Feature Types to avoid reading in unnecessary data
- Improve performance by collapsing Bookmarks to prevent excess caching

5.3 Workspace Layout and Styling

Demonstrates	Use of best practice workspace layout, Annotations, Bookmarks and styling
Overall Goal	Implement best practice workspace layout and styling
Data	Addresses (ESRI Geodatabase), Crime Data (CSV), Swimming Pools (OSM)
Start Workspace	C:\FMEModularData\Workspaces\5.03-BestPractice-WorkspaceStyling-Begin.fmw
End Workspace	C:\FMEModularData\Workspaces\Complete\5.03-BestPractice-WorkspaceStyling-Complete.fmw

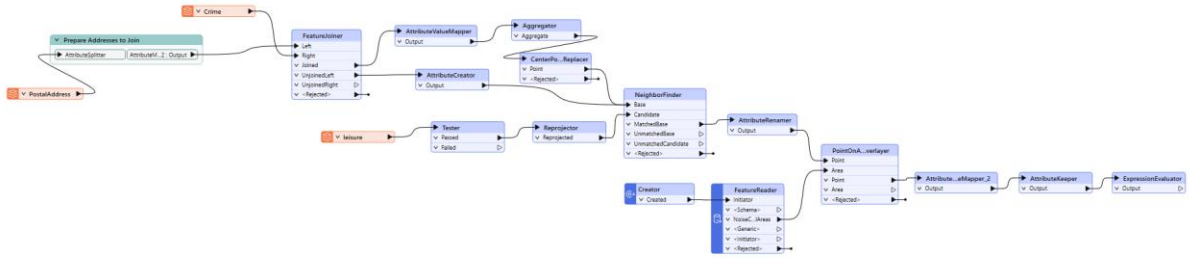
You have been assigned to a project to calculate the "walkability" of each address in the city of Vancouver.

Your colleague wasn't aware of FME style best practices when they gave us the workspace, which made working with it a bit challenging. We need to present our workspace, so we want it to look neat, organized, and well-documented.

5.3.1 Launch FME Workbench and Open Workspace

Launch the FME Workbench, if it isn't open already. Within the Get Started section of the Workbench, click on Open Workspace. Navigate to and open C:\FMEModularData\Workspaces\5.03-BestPractice-WorkspaceStyling-Begin.fmw Alternatively, use your workspace from the previous exercise.

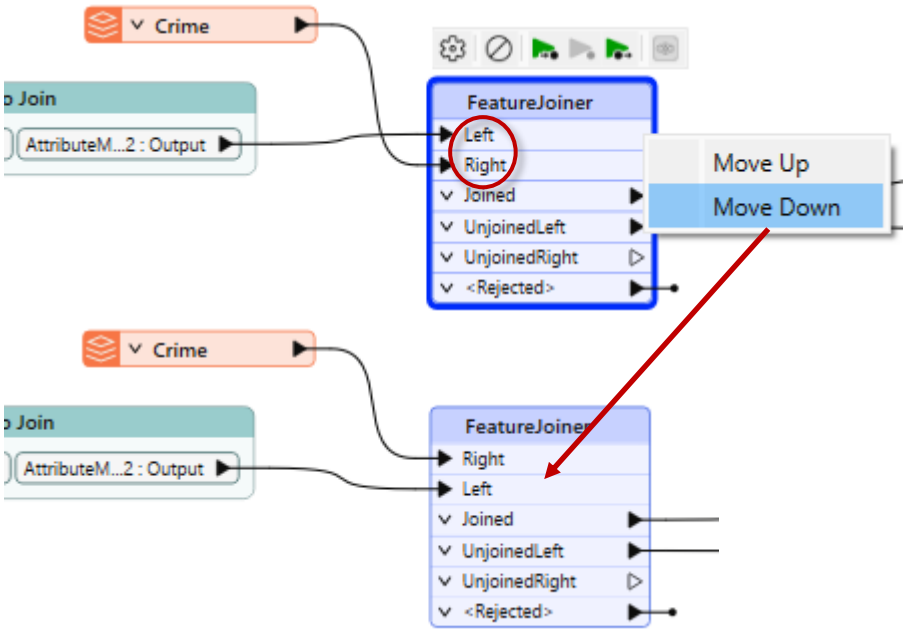
Note: If using your own workspace the layout may look different to the screenshots in this exercise. So, pay close attention to transformer and port names.



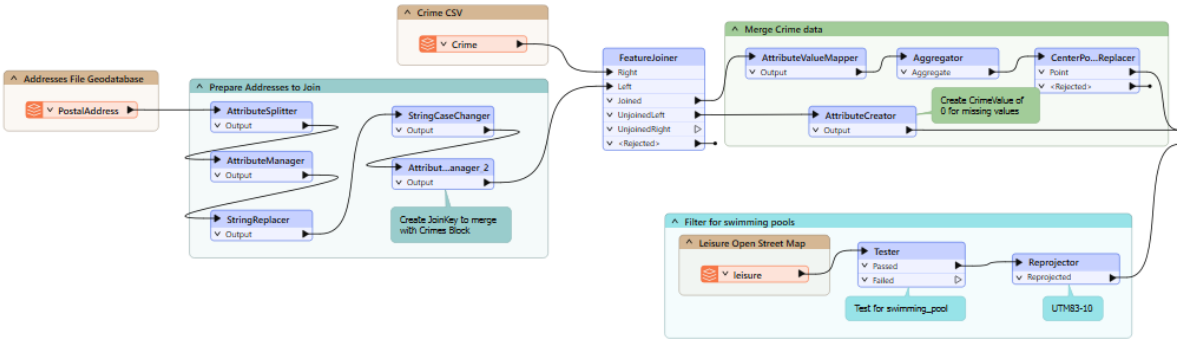
5.3.2 Rearrange Transformers

Firstly, let's clean up the transformers. Move the transformers around so that there are no overlapping connections.

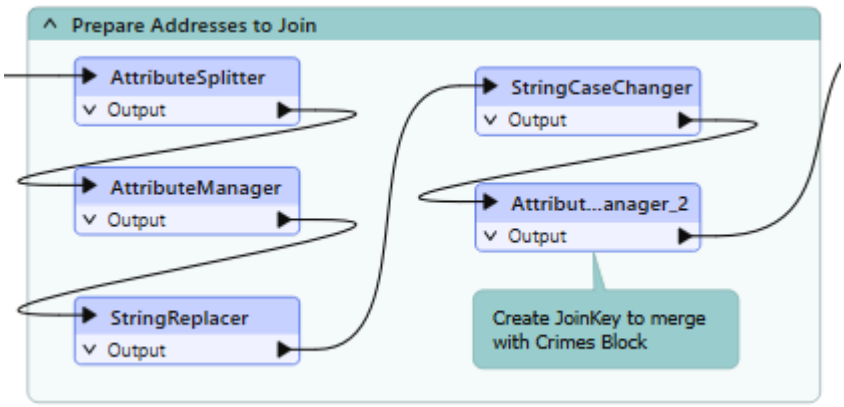
For the FeatureJoiner, you could move the Crimes reader below the Prepare Addresses to Join bookmark, or you can reorder the FeatureJoiner ports. Right click on the Left input port, and select Move Down. Now the two connection lines are not crossing:



Move the transformers into a logical order and add a bookmark around any logical groupings.
Drag a box and select all the required transformers and then either click on the Bookmark toolbar button, or press Ctrl+B on your keyboard :

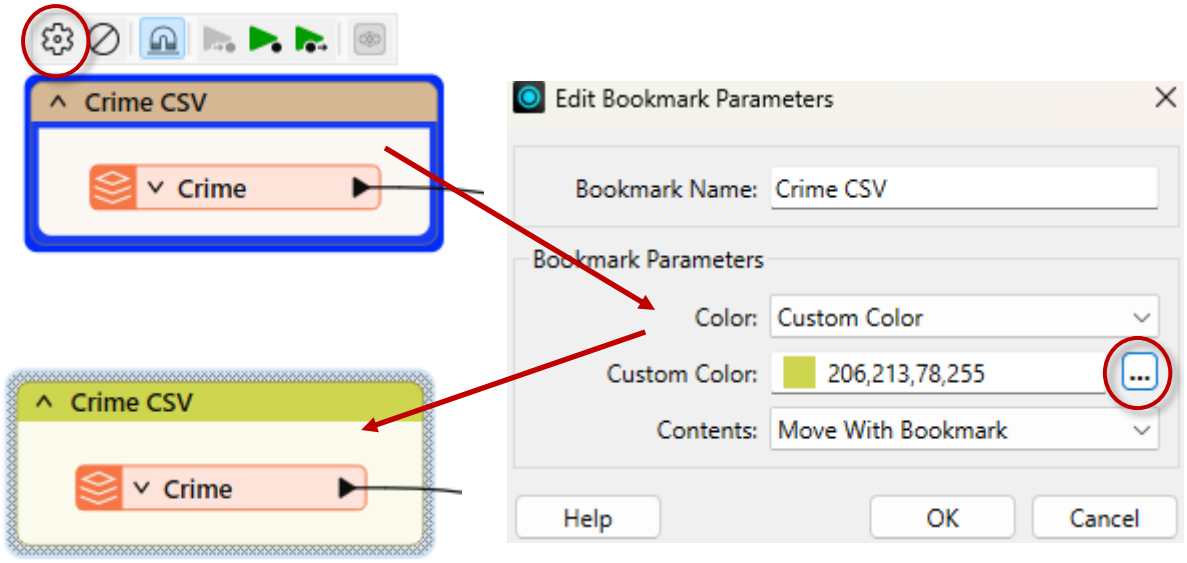


Don't forget to expand the Prepare Addresses to Join bookmark from the previous exercise and organize those transformers:

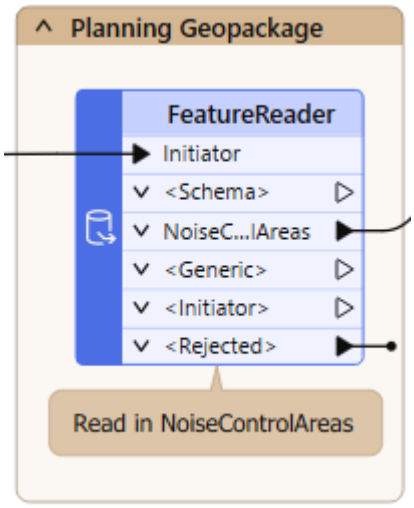


5.3.3 Add Style

Use colour to highlight what is going on.
By adding a bookmark around a reader or writer and then setting the colour to the preset Readers/Writers colour, it is quick to see at a glance where your readers or writers are:

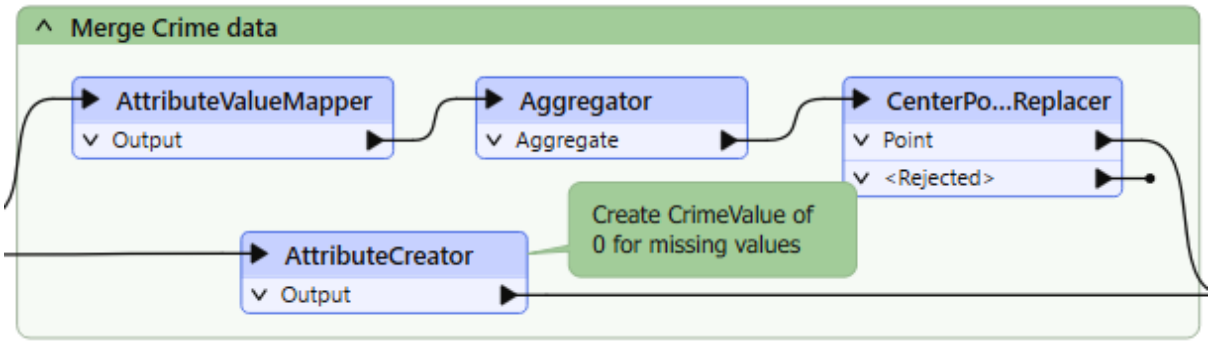


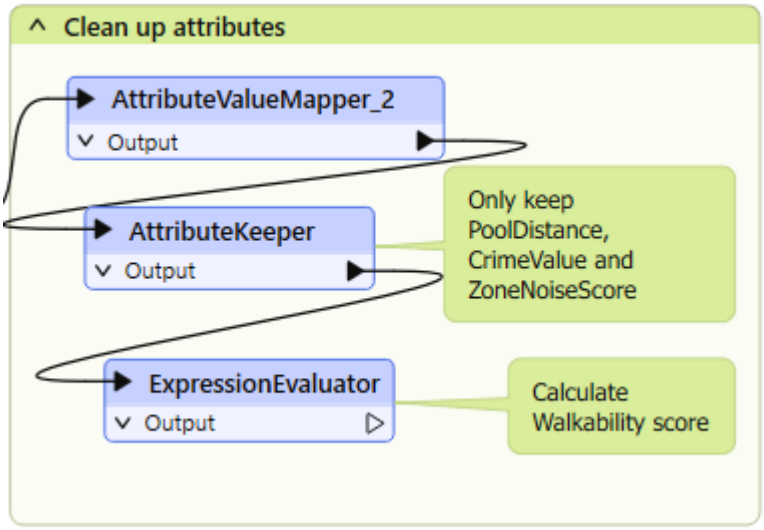
You can also do this with FeatureReaders and FeatureWriters:



5.3.4 Use Annotations

Now add annotations. Adding good annotation where necessary will help determine what is going on in the workspace.

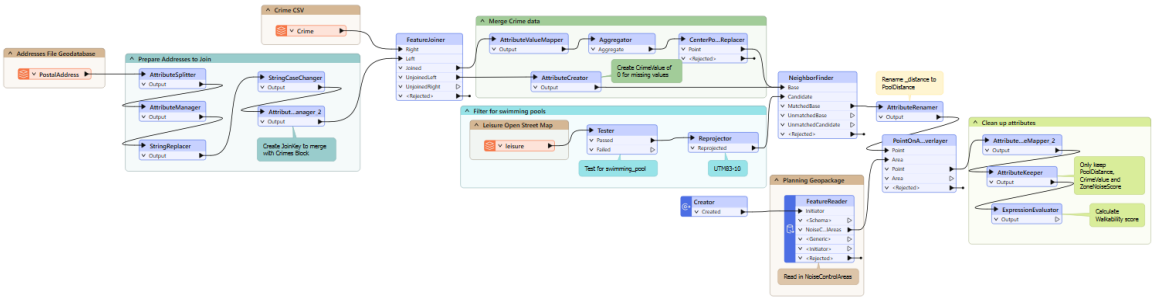




FME Lizard Tip

Don't make your annotations too technical, write them like a human!

Your final workspace should look something like this:



5.3.5 Save the Workspace

Click on the Save button, or use File > Save As on the menu bar. Then navigate to where you'd like to save your workspace file: C:\FMEModularData\Output\Workspaces Give your workspace a name. Click Save.

Congratulations

By Completing this exercise you have learned how to:

- Implement best practice layout and styling
- Rearrange transformers into logical layout that groups those carrying out a single or related task
- Use annotations to clarify the processes taking place in a workspace
- Use bookmarks to turn a single workspace into defined sections
- Avoid poor design choices like overlapping connections

Got any questions?

We’re happy to help!

0121 232 8000

info@misoportal.com