Article

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

How to Merge and Join Tabular Data | Creating Enhanced Datasets by Integrating Disparate Data Sources

Aug 22, 2022 • Knowledge

Product Type
FME Desktop

FME Version
2022.0

Introduction

This tutorial will work through several exercises to introduce the core concepts used in typical tabular workflows and the associated FME transformers that can be utilized to perform merging and joining operations. Each exercise builds upon the lessons from the previous exercise.

We will be generating a report that enhances Water Meter data for an inspection program by supplementing it with additional attributes (fields) using Municipal Lot/Property data.

Content Overview

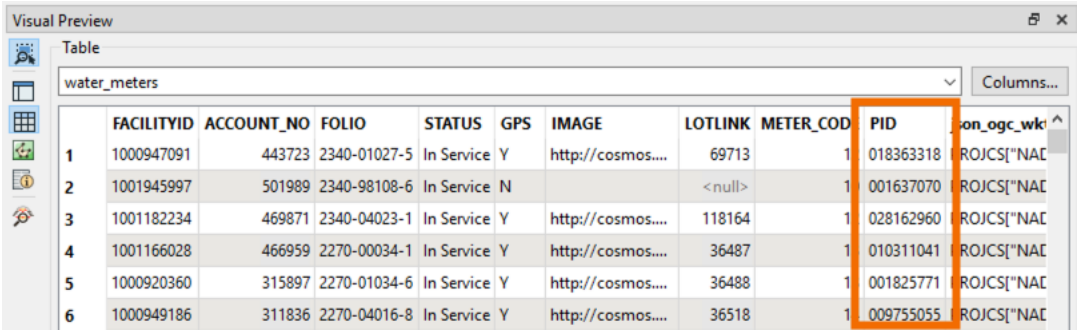
Step-by-step Instructions

Exercise 1: FeatureJoiner

1. Inspect water_meters

Open the starting workspace Merging and Joining Tabular\Workspaces\Exercise1-4Begin.fmw in FME Workbench. This workspace already has the readers and writers set up for us, but we will be adding in the transformers.

First, let's view the starting water_meter dataset, this is the dataset we will base our entire project around. Enable Feature Caching, then run the workspace. Inspect the output of the water_meters reader feature type. There isn't much in this dataset that would be useful for an inspector in the field. Take note of the PID attribute which will be used as a key to join additional tables to this water meter dataset.



	FACILITYID	ACCOUNT_NO	FOLIO	STATUS	GPS	IMAGE	LOTLINK	METER_COD	PID	on_ogc_wk
1	1000947091	443723	2340-01027-5	In Service	Y	http://cosmos....	69713	1	018363318	ROJCS["NAD
2	1001945997	501989	2340-98108-6	In Service	N		<null>	1	001637070	ROJCS["NAD
3	1001182234	469871	2340-04023-1	In Service	Y	http://cosmos....	118164	1	028162960	ROJCS["NAD
4	1001166028	466959	2270-00034-1	In Service	Y	http://cosmos....	36487	1	010311041	ROJCS["NAD
5	1000920360	315897	2270-01034-6	In Service	Y	http://cosmos....	36488	1	001825771	ROJCS["NAD
6	1000949186	311836	2270-04016-8	In Service	Y	http://cosmos....	36518	1	009755055	ROJCS["NAD

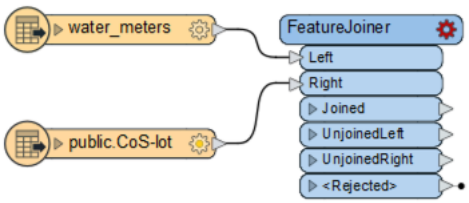
2. Add Lot Zone Codes for Each Water Meter

The public.CoS-lot table comes from the Training PostgreSQL database. It contains the neighborhood zones for each property that we want to add to the water meters dataset.

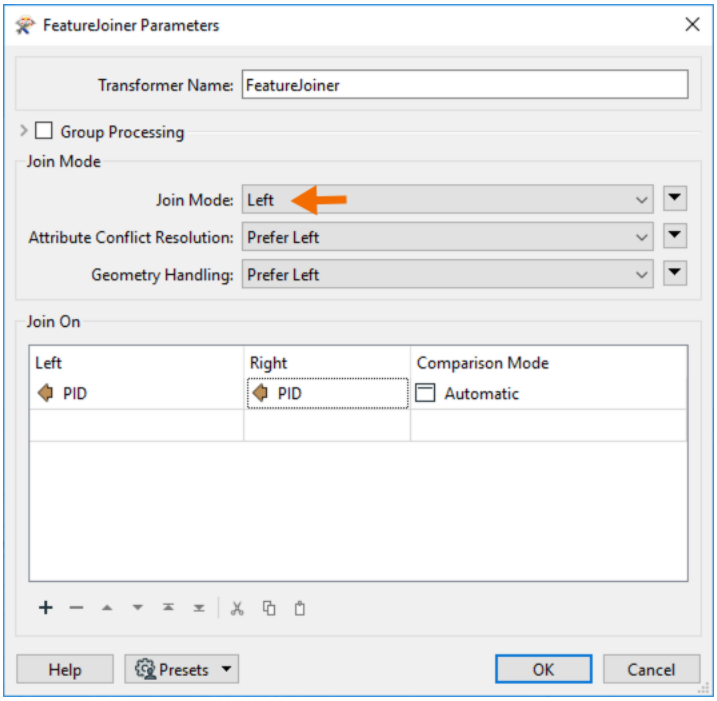
To connect to the Safe Software provided training database, use the following credentials when setting up your connection. Please be aware that this is a public database, and the data available can change at any time. This database is for training purposes only and should only be used during training courses and articles:

- **Name:** Training
- **Host:** postgis.train.safe.com (http://postgis.train.safe.com/)
- **Port:** 5432
- **Database:** fmedata
- **Username:** fmedata
- **Password:** fmedata

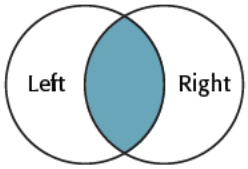
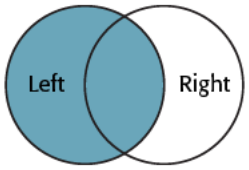
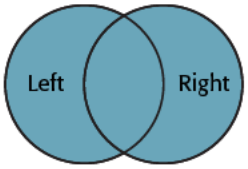
To join the datasets together, we will use the FeatureJoiner (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/featurejoiner.htm) transformer. Add a FeatureJoiner to the canvas and connect the water_meters reader feature type to the Left input port and the public.CoS-lot reader feature type to the Right input port.



Open the FeatureJoiner parameters and set the Join Mode to Left to retain all of the water meter features. Then use the PID attribute for both the Left and Right join attributes.



A Left join will retain all the water meter features whether there is a matching CoS-lot feature or not:

Join Mode	Joined Output	
Inner	• Only features that have matching Left and Right pairs.	
Left	• Features that have matching Left and Right pairs, AND • All other unmatched Left features.	
Full	• Features that have matching Left and Right pairs, AND • All unmatched Left and Right features.	

3. Inspect Joined Datasets

Run the workspace and inspect the output from the Joined port of the FeatureJoiner. Notice how the attributes from the Lot table are now joined to the water meter data.

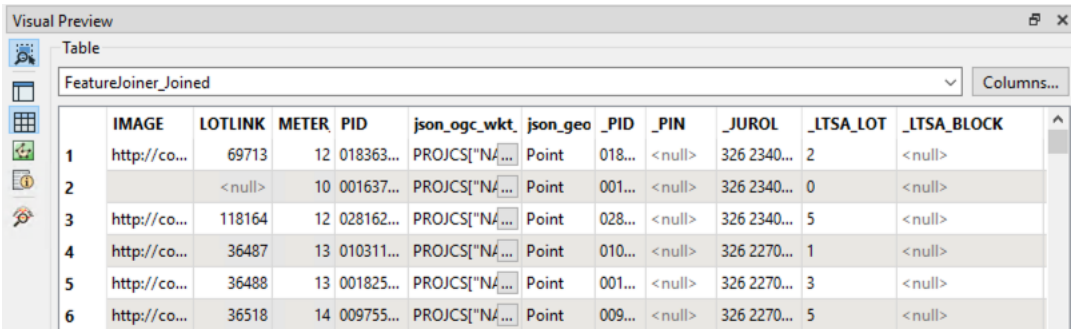


	IMAGE	LOTLINK	METER	PID	json_ogc_wkt_	json_geo	_PID	_PIN	_JUROL	_LTSA_LOT	_LTSA_BLOCK
1	http://co...	69713	12	018363...	PROJCS["NA...	Point	018...	<null>	326 2340...	2	<null>
2		<null>	10	001637...	PROJCS["NA...	Point	001...	<null>	326 2340...	0	<null>
3	http://co...	118164	12	028162...	PROJCS["NA...	Point	028...	<null>	326 2340...	5	<null>
4	http://co...	36487	13	010311...	PROJCS["NA...	Point	010...	<null>	326 2270...	1	<null>
5	http://co...	36488	13	001825...	PROJCS["NA...	Point	001...	<null>	326 2270...	3	<null>
6	http://co...	36518	14	009755...	PROJCS["NA...	Point	009...	<null>	326 2270...	5	<null>

The screenshot above has the attributes from the Lot table prefixed with a _ to highlight which table the attributes came from. Your data will not have this.

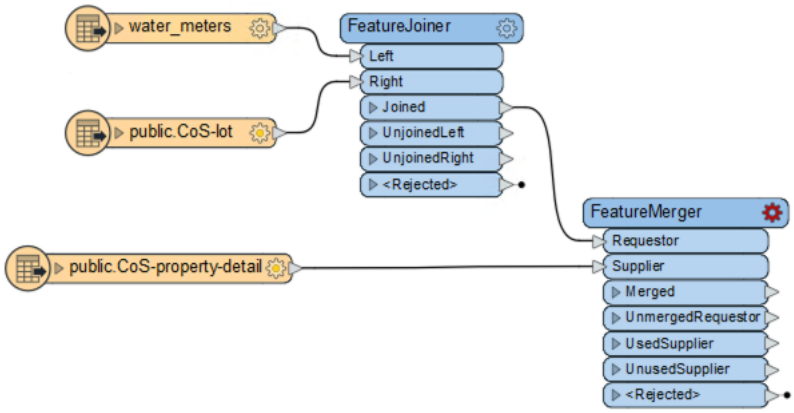
Note: We could have used the DatabaseJoiner instead, but it is slower at 2 minutes vs. 6.5 seconds with the FeatureJoiner in this scenario. You may consider using the DatabaseJoiner for joining smaller tables (i.e. look-ups) or when joining an indexed column. In this case, since the column is not indexed, the FeatureJoiner will have better performance. For more information, see the DatabaseJoiner (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/databasejoiner.htm) documentation.

Exercise 2: FeatureMerger

The source water meter data contains a FOLIO attribute with only one value. However, the field inspectors are requesting that we replace the existing FOLIO value with all the FOLIO numbers for each water meter. We'll need to perform a one-to-many (1:M) join. Let's do this with the Property Details table and a FeatureMerger.

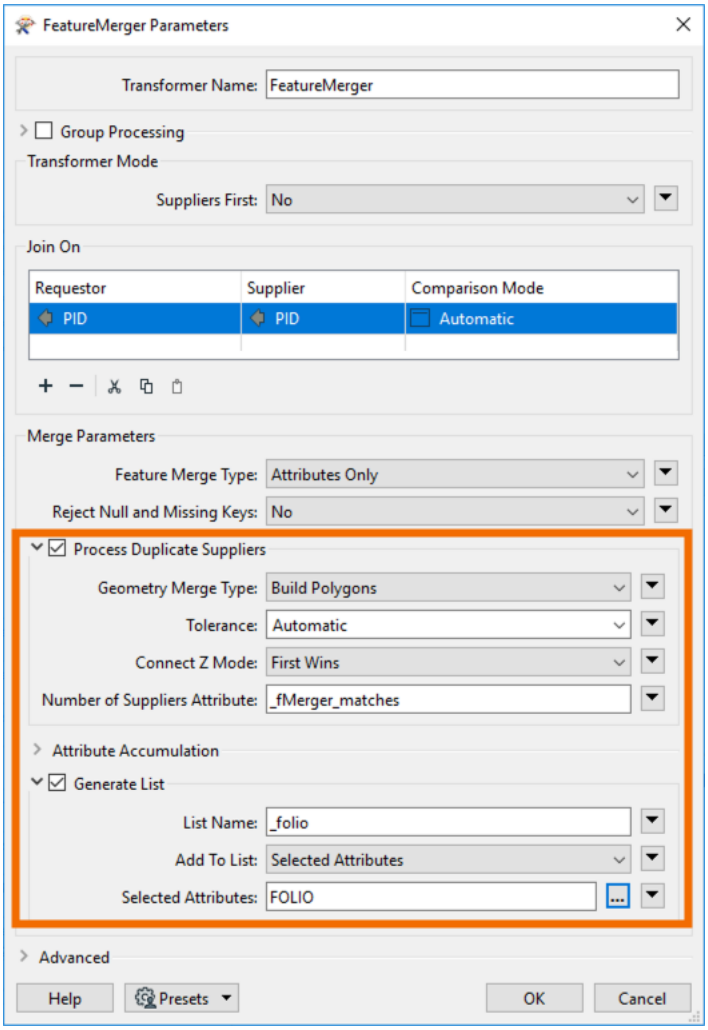
4. Get All FOLIO Values for Each Water Meter

To get all of the FOLIO values for each water meter, we will use the FeatureMerger (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/featuremerger.htm) transformer. Add a FeatureMerger to the canvas and connect the Requestor input port to the Joined output port on the FeatureJoiner. Then connect the Supplier input port to the public.CoS-property-detail reader feature type.



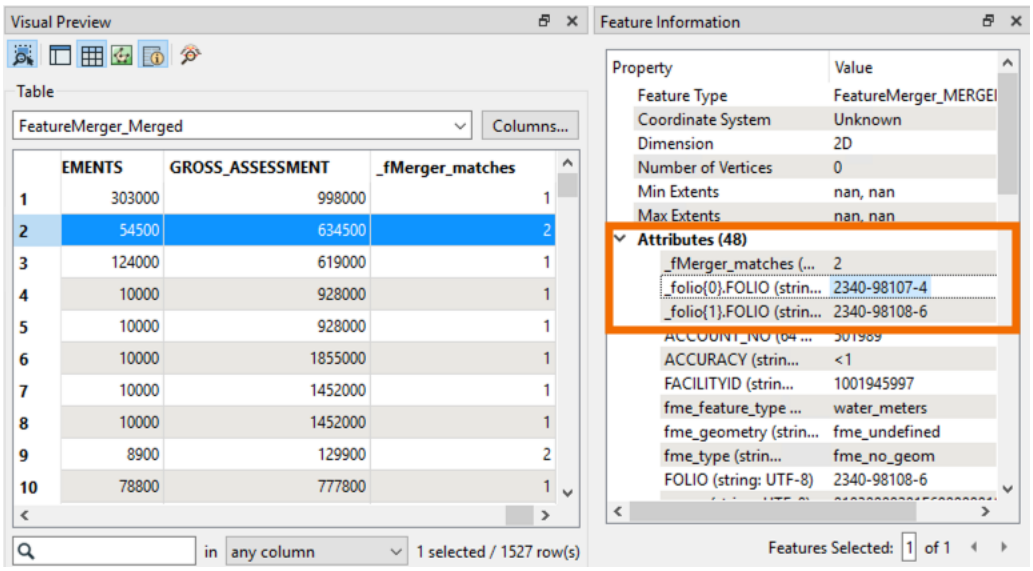
In the FeatureMerger parameters, use the PID as the join key for both the Requestor and Supplier. Then enable Process Duplicate Suppliers, to ensure all FOLIO values are matched with their corresponding PID. Set the Number of Suppliers Attribute to _fMerger_matches. We'll use this attribute to help validate the features with multiple FOLIO numbers later on.

Next, enable Generate List and set the List Name to _folio. Click on the ellipsis next to Selected Attributes and select FOLIO. Click OK to close the parameters.



5. Inspect _fMerger_matches

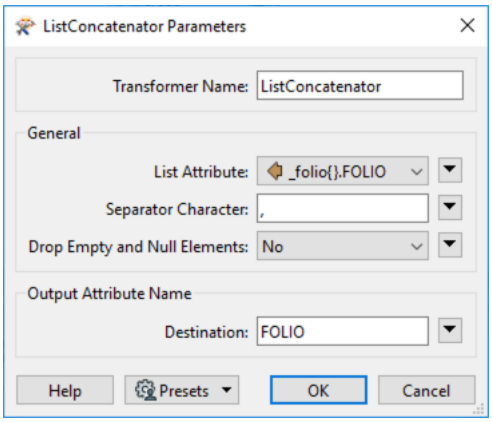
Run the workspace and inspect the Merged output port on the FeatureMerger. Find a feature with a value greater than 1 for the _fMerger_matches attribute. Select the feature in the Table view, and then open the Feature Information Window. You should see a list attribute called _folio{}.FOLIO which will contain one FOLIO number per list element.



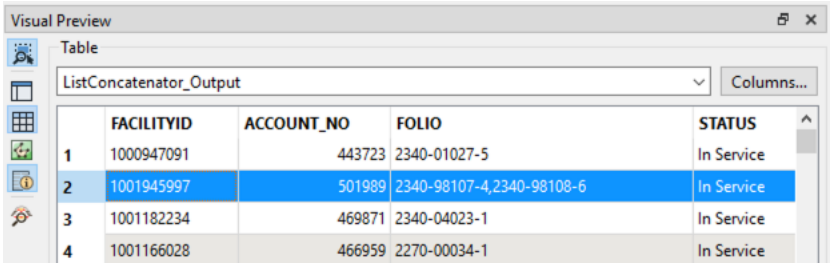
6. Concatenate FOLIO Numbers

We need all of these FOLIO numbers to be contained within a single attribute. Add a ListConcatenator to the canvas and connect it to the Merged output port of the FeatureMerger. In the parameters set the List Attribute to _folio{}.FOLIO and set the Separator Character to a comma (,). Finally, set the Destination to

FOLIO, this will overwrite the FOLIO attribute with the newly concatenated list of FOLIO numbers.



If you were to rerun the workspace, you'll see the FOLIO attribute now has the attributes concatenated separated by a comma.

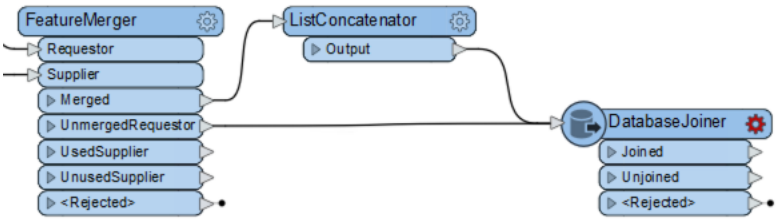


Exercise 3: DatabaseJoiner

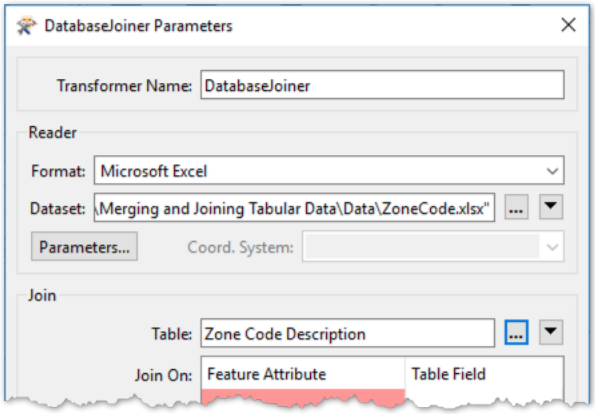
7. Lookup Zone Descriptions

From the previous Exercise, notice that there are some water meters that don't have matching property details (inspect the Unmerged Requestor port on the FeatureMerger). We want to continue working with all our water meter features in the next step of the workflow.

Add a DatabaseJoiner (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/databasejoiner.htm) to the canvas, and connect it to both the Output port from the ListConcatenator and the Unmerged Requestor output port on the FeatureJoiner to ensure we're performing a Left Join from the previous step.



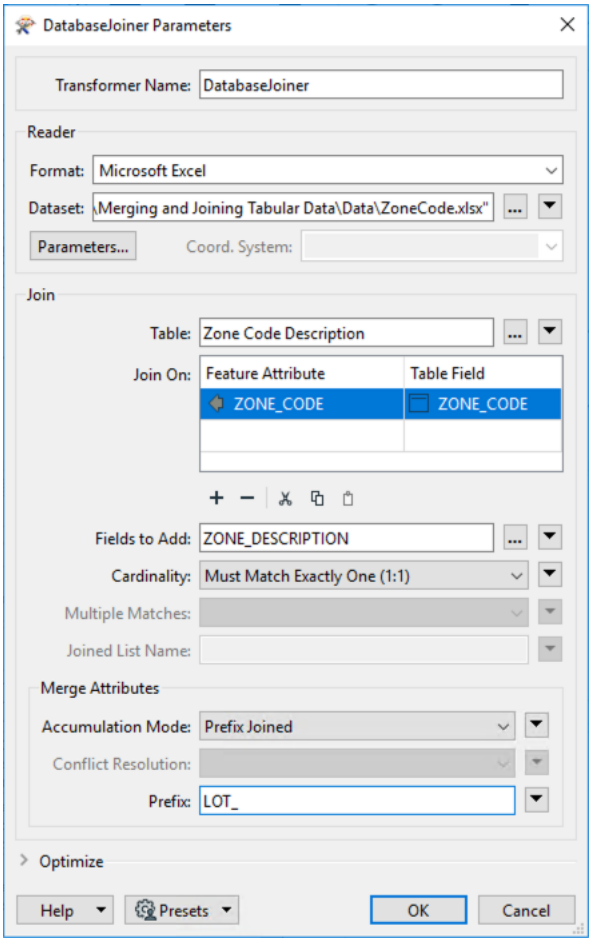
In the DatabaseJoiner parameters, set the Format to Microsoft Excel then browse to the ZoneCode.xlsx dataset. Click on the ellipsis next to Table and select Zone Code Description.



For the Join On parameter, the Feature Attribute is from the data coming into the DatabaseJoiner from the workspace. The Table Field is from the table we are reading in using the DatabaseJoiner. Use the ZONE_CODE attribute for both parameters to perform the join.

Unlike the FeatureJoiner, you can filter out any attributes you do not want to merge to your dataset. For the Fields to Add parameter, add the ZONE_DESCRIPTION. The DatabaseJoiner also allows for greater control on cardinality. For this example, since we know that there should only be one ZONE_CODE in the lookup table for each ZONE_CODE, we'll select Must Match Exactly One (1:1) as the Cardinality.

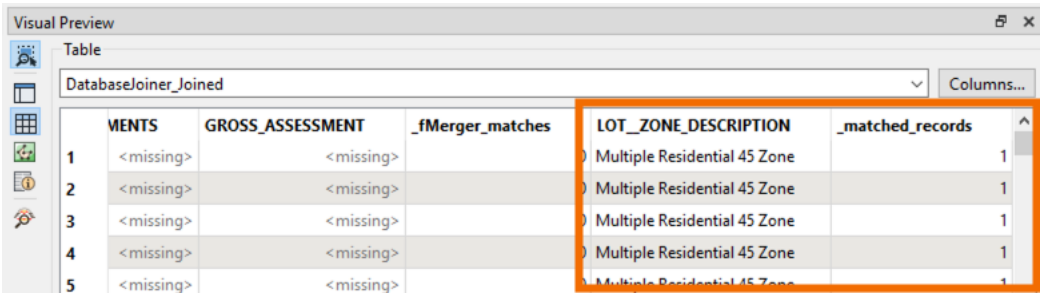
Finally, so we know which table the data is coming from, set the Accumulation Mode to Prefix Joined and set the Prefix parameter to LOT_.



8. Inspect Joined Zones

Run the workspace and inspect the Joined output port on the DatabaseJoiner. You should see LOT_ZONE_DESCRIPTION merged with the data, along with an

attribute called _matched_records which indicates the number of matched records as a result of the join.

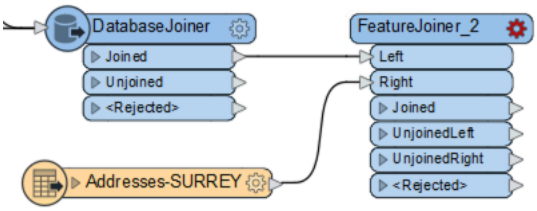


Exercise 4: Obtain Addresses & Output Validation Report

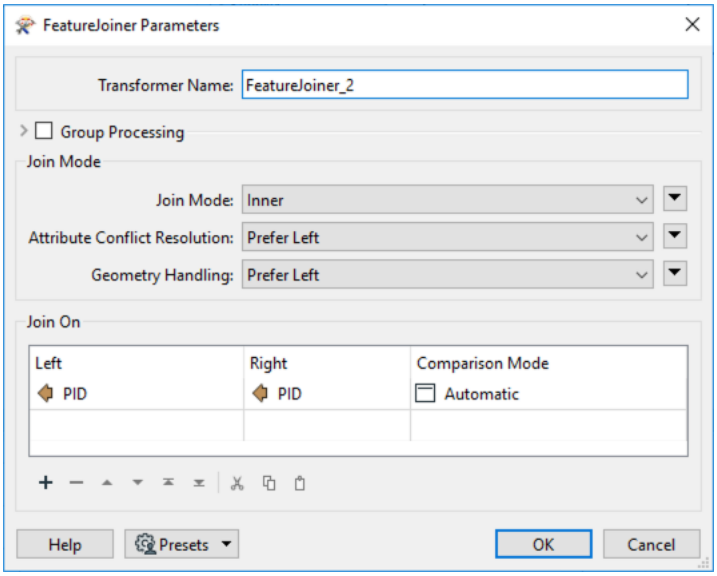
The field inspectors require the lot addresses for each water meter. We'll use the FeatureJoiner again to do a quick join with Addresses-SURREY.csv.

9. Join Addresses

Add a FeatureJoiner to the canvas and connect the Left input port to the Joined output port on the DatabaseJoiner. Then connect the Right input port to the Addresses-SURREY reader feature type.



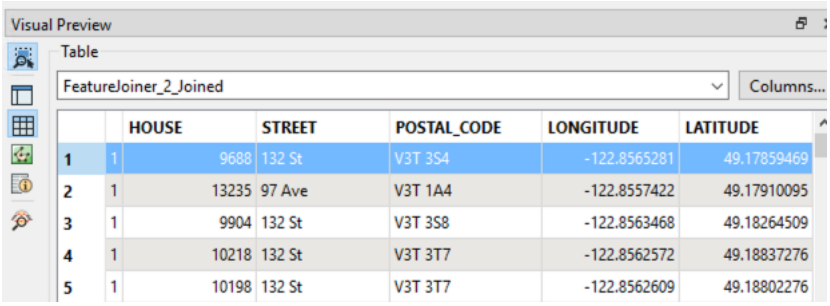
In the FeatureJoiner_2 parameters, set the Join Mode to Inner. This will allow us to filter out any water meters/PIDs that don't have an associated address. Then set PID as the join key for both Left and Right tables.



10. Inspect Joined Addresses

Run the workspace and inspect the Joined output port on the FeatureJoiner_2, which contains the water meter information for our field inspectors. The information from the Unjoined Left output port contains the water meters that are missing addresses.

Note that the workspace may fail due to features missing the right key. You can either set Workspace: Rejected Feature Handling (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Desktop/Workbench/options_workspace_defaults.htm) to Continue or by connecting a Logger (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/logger.htm) to the <Rejected> output port on the FeatureJoiner_2.

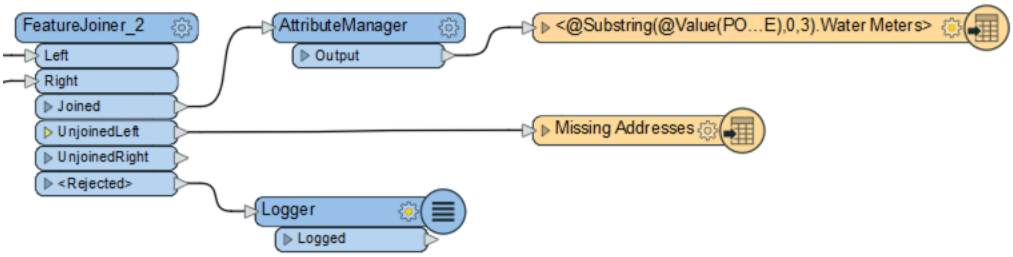


11. Output Results

Connect the Joined output port on the FeatureJoiner_2 to the AttributeManager. The AttributeManager was set up ahead of time to remove the attributes we don't need as well as to clean up other attributes. Take a moment to review the changes in the AttributeManager.

The AttributeManager is connected to the @Substring(@Value(POSTAL_CODE),0,3).Water Meters writer feature type. This writer feature type is creating a worksheet for the Water Meters - Field Inspection Microsoft Excel Workbook. Each postal code FSA, which is the beginning three characters of a postal code, will have its own worksheet with the water meter data listed.

Now connect the UnjoinedLeft output port on the FeatureJoiner_2 to the Missing Addresses writer feature type. This is writing out a validation report all of the water meters that are missing addresses to the same Water Meters - Field Inspection workbook for the data custodians to review at a later date.



Run the workspace, and inspect the output Water Meters - Field Inspection INLINEQUERIER.xlsx file.

If you have time, see Optional: Build an HTML Report, Publish to FME Server and Create a Workspace App at the bottom of this article. You will create an HTML report that will come in handy for the field inspectors, and you will use FME Server to create an App to generate the report.

Save and close the workspace. We will start with a different workspace for Exercise 5.

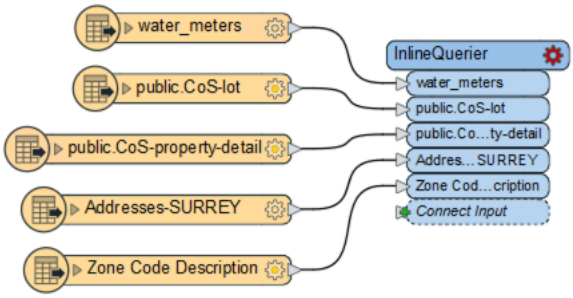
Exercise 5: InlineQuerier

We can also join datasets as we did in Exercises 1-4 using an InlineQuerier transformer. The InlineQuerier can perform SQL queries on any dataset already read into your workspace. As you'll see in this exercise, for those who are comfortable working with SQL, the InlineQuerier can help simplify your workspace.

12. Add an InlineQuerier Transformer

Open Exercise5Begin.fmw in FME Workbench. This workspace uses the same data we were using in the previous exercises, but it only has the readers and writers.

Add an InlineQuerier (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/inlinequerier.htm) to the canvas and connect all of the reader feature types to the Connect Input port (there should be five reader feature types connected in total).



13. Create Output Port SQL Queries

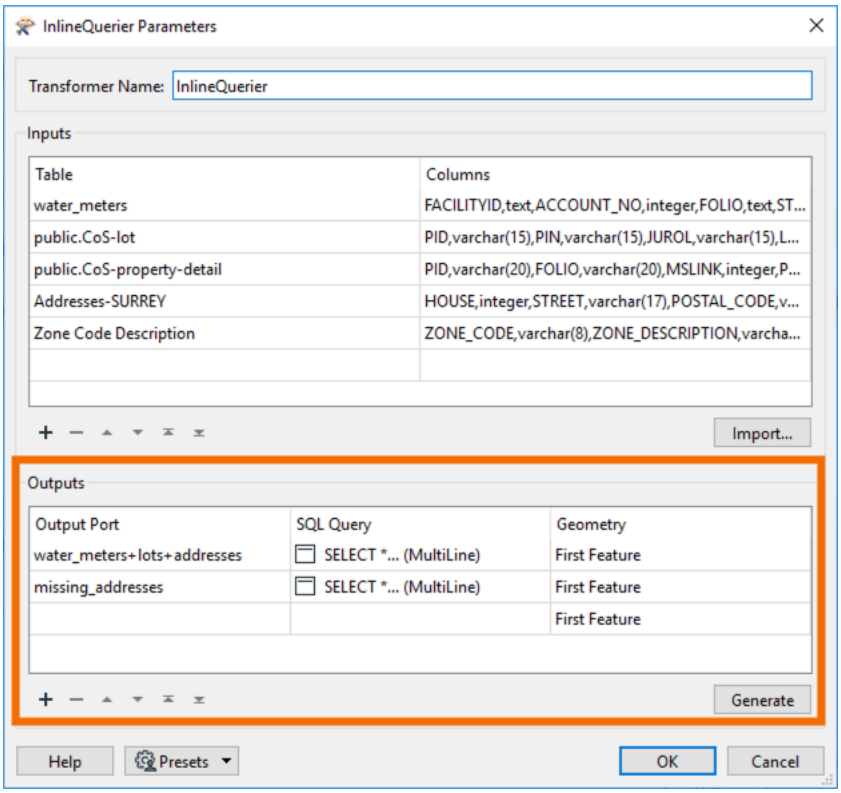
After connecting the reader feature types, a new input port was created for each dataset, but there are no output ports. We will need to modify the parameters to create these ports.

Open the InlineQuerier parameters, and notice that all of the input datasets are listed at the top. Under Outputs, enter the following parameters:

Output Port	SQL Query

water_meters+lots+addresses	SELECT *
	FROM "water_meters" m
	  LEFT JOIN "public.CoS-lot" l
	  ON m.PID=l.PID
	  LEFT JOIN "Zone Code Description" z
	  ON l.ZONE_CODE=z.ZONE_CODE
	  INNER JOIN "Addresses-SURREY" a
	  ON m.PID=a.PID;

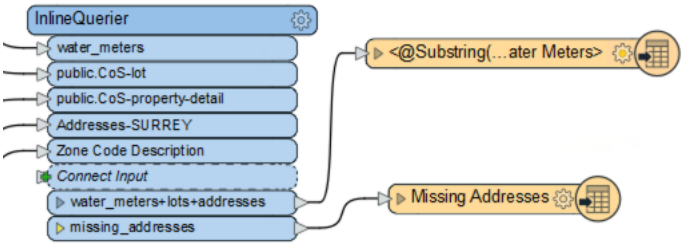
missing_addresses	SELECT *
	FROM "water_meters" m
	  LEFT JOIN "Addresses-SURREY" a
	  ON m.PID=a.PID
	WHERE a."POSTAL_CODE" IS NULL;



14. Connect Writer Feature Types

One final step is to connect the newly created output ports. Connect the water_meters+lots+addresses output port to the @Substring(@Value(POSTAL_CODE),0,3).Water Meters writer feature type. Then connect the missing_addresses to the Missing Addresses writer feature type.

The SQL queries are performing the exact same joins that we did in Exercises 1 - 4, only condensed into a single transformer.



Exercise 6: SQLExecutor & SQLCreator

We'll be modifying our original workspace from Exercises 1-4 again, this time demonstrating how the SQLExecutor can be used to leverage the database instead of reading all your features into your workspace.

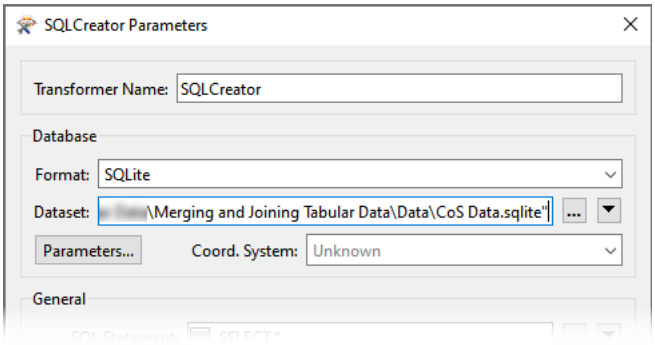
When you have multiple tables being read in from a database, consider using the SQLCreator or SQLExecutor transformers. These transformers allow you to query the database directly, often improving performance without having the need to read entire tables into your workspace. Learn more about leveraging the power of databases in Tutorial: Let the Database Do the Work (https://community.safe.com/s/article/tutorial-let-the-database-do-the-work).

15. Generate the Water Meter Report using an SQLCreator

Open Exercise6Begin.fmw in FME Workbench. The two writer feature types from the previous exercises have been added to the canvas. For this exercise, all the source tables are provided in an SQLite database in Merging and Joining Tabular\Data\CoS Data.sqlite

Add an SQLCreator (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/sqlcreator.htm) to the canvas. The SQLCreator allows you to initiate a workspace with a database query.

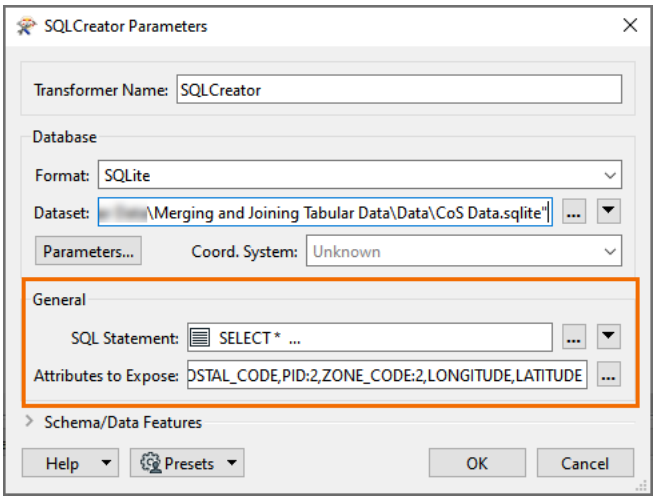
Open the SQLCreator parameters, select SQLite as the Format, and navigate to Merging and Joining Tabular\Data\CoS Data.sqlite.



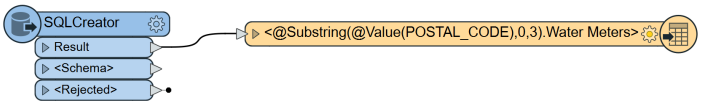
Click the ellipsis next to the SQL Statement parameter. Let's create a SQL query that replicates the joins performed earlier in Exercises 1-4, joining the water_meters table with the lot, zone_code_description and addresses tables:

```
SELECT *
    FROM "water_meters" m
        LEFT JoiN "lot" l
            ON m."PID"=l."PID"
         LEFT JOIN "zone_code_description" z
            ON l."ZONE_CODE"=z."ZONE_CODE"
         INNER JOIN "addresses" a
            ON m."PID"=a."PID";
```

Click OK to exit from the SQL Statement dialog and back to the SQLCreator Parameters. Click on the ellipsis next to the Attributes to Expose parameter. Click on the Populate from SQL Query… button in the resulting dialog. This will generate the attributes that will be output by the SQLCreator Result output port from the SQL Query above.



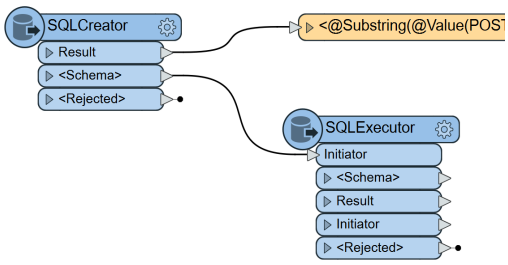
Connect the Result output port from the SQLCreator to the @Substring(@Value(POSTAL_CODE),0,3).Water Meters writer feature type for the water meter inspection report.



16. Generate the Data Validation Report Using an SQLExecutor

Let's create the Data Validation Report for missing addresses using the SQLExecutor (http://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/sqlexecutor.htm) transformer. The SQLExecutor allows for midstream database queries to be performed.

Add an SQLExecutor to the canvas and connect it to the SQLCreator <Schema> output port–we're simply using the schema feature as an initiator for the SQLExecutor.



Open the SQLExecutor parameters, and select SQLite as the Format, and navigate to Merging and Joining Tabular\Data\CoS Data.sqlite.

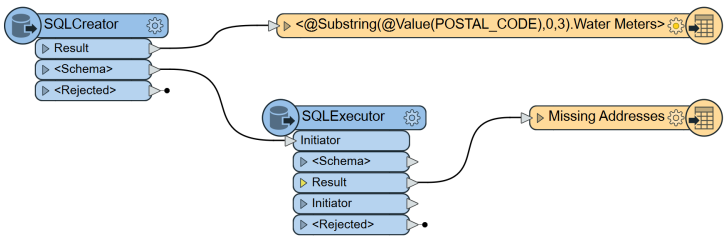
Click the ellipsis next to the SQL Statement parameter. Let's create a SQL statement that queries for water meter features don't have any associated addresses. We'll do this by joining the water meters table with the addresses table and determining which features have no postal codes:

```
SELECT *
FROM "water_meters" m
    LEFT JOIN "addresses" a
        ON m.PID=a.PID
WHERE a."POSTAL_CODE" IS NULL;
```

Click OK to exit from the SQL Statement dialog and back to the SQLCreator Parameters. Click on the ellipsis next to the Attributes to Expose parameter. Click on the Populate from SQL Query… button in the resulting dialog. This will generate the attributes that will be output by the SQLExecutor Result output port from the SQL Query above.

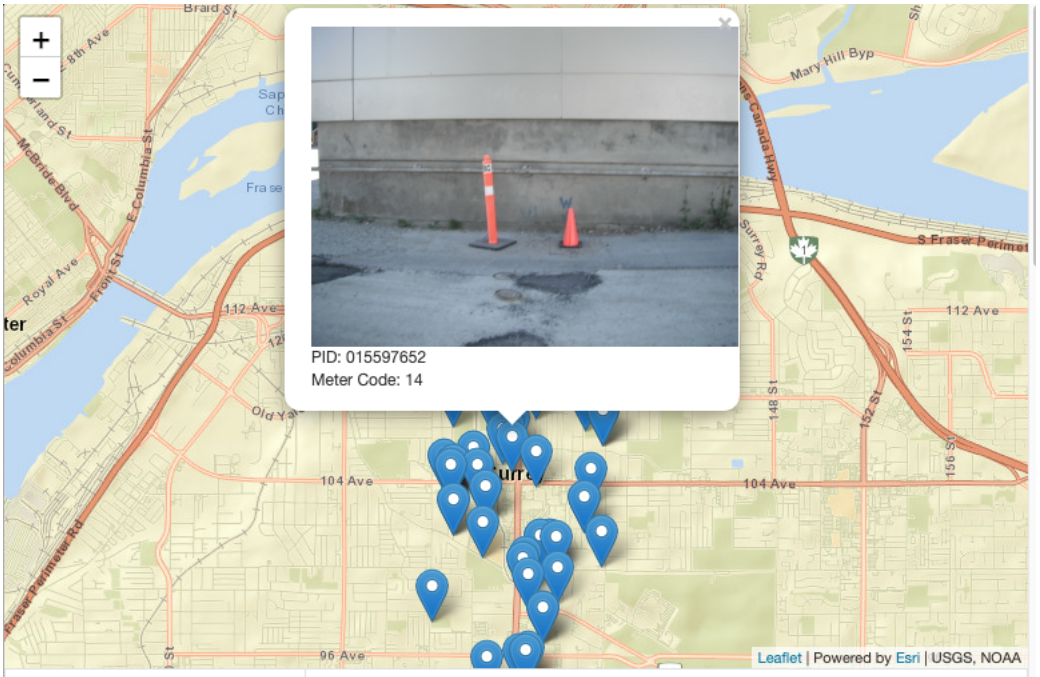
Connect the Result output port from the SQL Executor to the Missing Addresses writer feature type.

Run the workspace, and inspect the output Water Meters - Field Inspection SQL.xlsx file.



OPTIONAL EXERCISE: Build an HTML Report, Publish to FME Server and Create a Workspace App

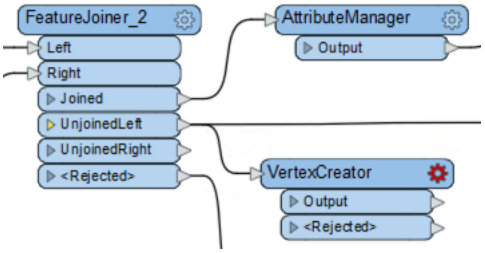
This is the optional section of Exercise 4. In this section, we will learn how FME can be used to generate an HTML report for an uploaded Water Meters dataset. We will then stream the results (HTML report) of a workspace using an FME Server Workspace App, similar to the image below. Click here (https://se-demo-fme-server-support.fmecloud.com/fmeserver/apps/FMEUCOptionalExercise4) to see it in action.



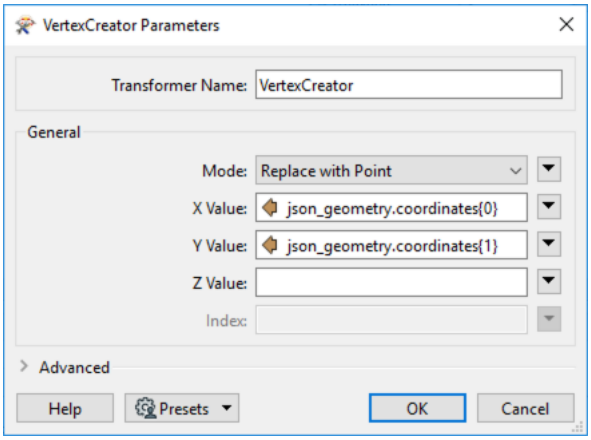
ID	Zone
1002257215	Single Family Residential Zone
1002258033	Single Family Residential Zone

1. Create Geometry

Open the Exercise1-4Complete.fmw workspace to begin or continue in the workspace you were building in Exercise 4. We're going to use the VertexCreator to create point geometry from the coordinate values that originated in the water_meters dataset (JSON), which have been carried on through the workflow. Add a VertexCreator to the canvas and connect it to the UnjoinedLeft output port on the FeatureJoiner_2.

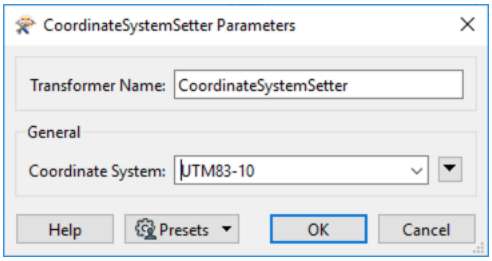


In the VertexCreator parameters, set the X Value to json_geometry.coordinate{0} and the Y Value to json_geometry.coordindate{1}.



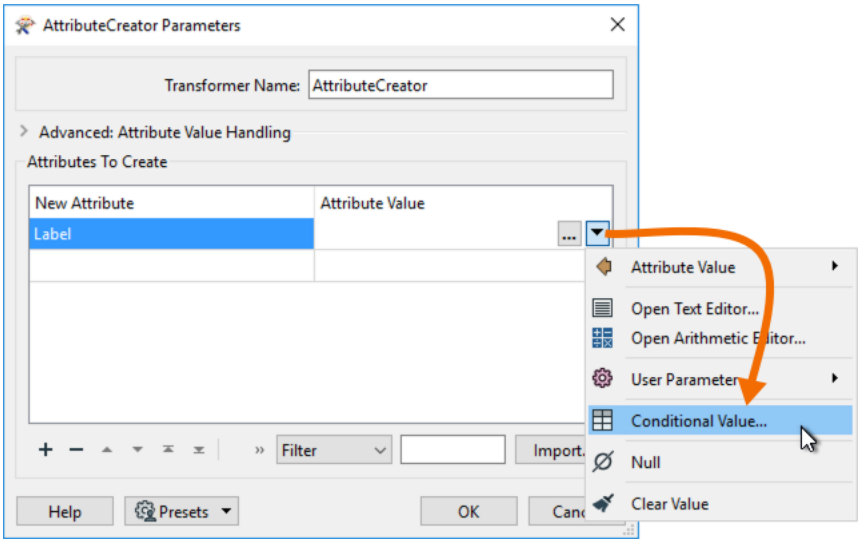
2. Define Coordinate System

Now that we have created the point geometry, we need to define the coordinate system. These points were recorded using UTM83-10. Add a CoordinateSystemSetter to the canvas and connect it to the VertexCreator. In the parameters, set the Coordinate System to UTM83-10.



3. Create Label for Pop-Up Box

Next, we'll create a label to use in the pop-up box that will appear when you click on a point in the HTML Report. Add an AttributeCreator to the canvas and connect it to the CoordinateSystemSetter. Create a new attribute called Label. Since some of the features in this dataset are missing an image URL, we will use a conditional value to set the value of the label attribute. Click on the drop-down arrow for Attribute Value and select Conditional Value.

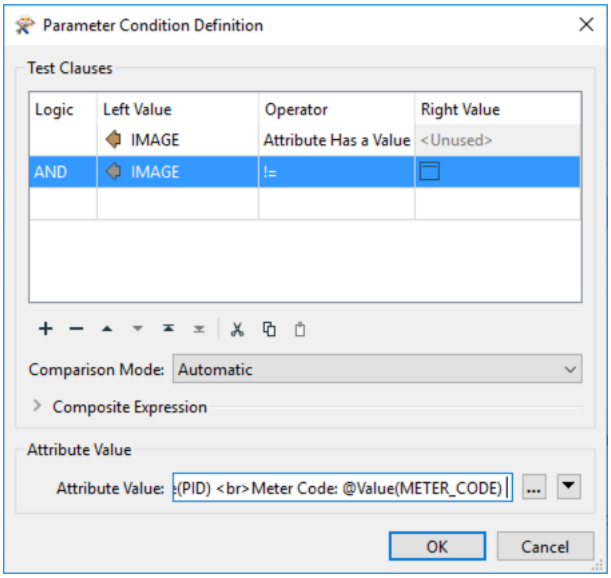


For the Condition Statement, double-click on the box next to If to open a dialog similar to the Tester transformer. Set the following condition:

```
IMAGE Attribute Has a Value AND IMAGE NOT <enter in a single space>
```

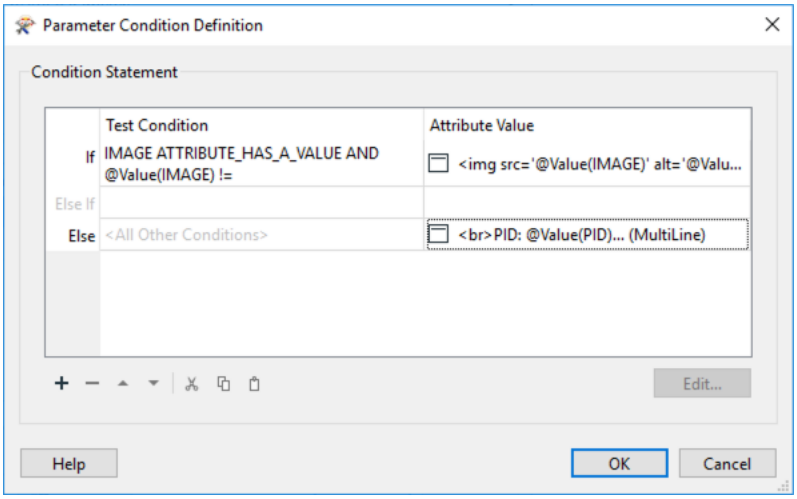
Set the Attribute Value to:

```
<img src='@Value(IMAGE)' alt='@Value(PID)' width ='320' height= '240'>
<br>PID: @Value(PID)
<br>Meter Code: @Value(METER_CODE)
```



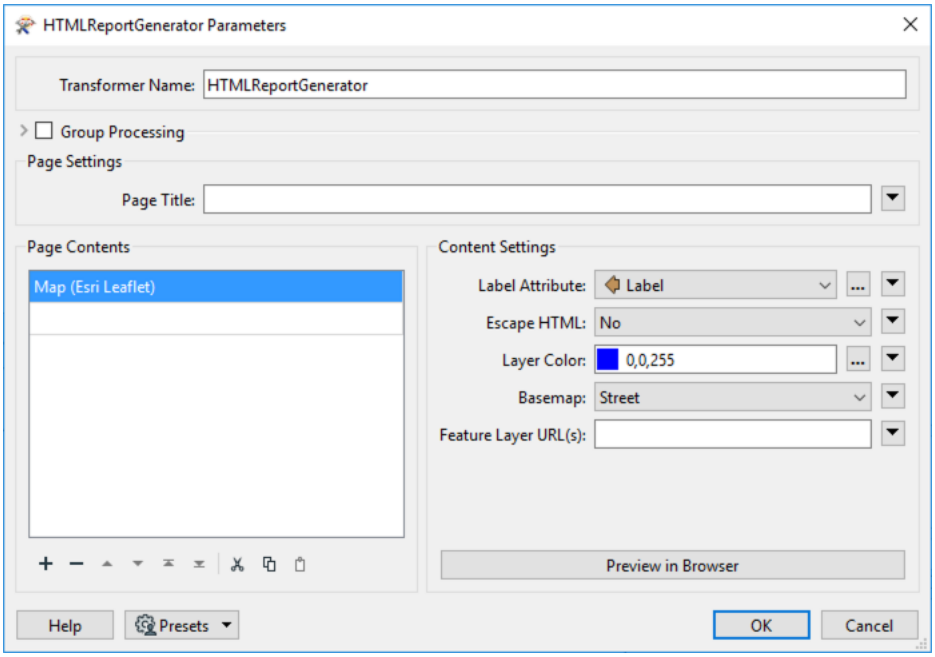
Then click OK. Set the Else Attribute Value to:

```
<br>PID: @Value(PID)
<br>Meter Code: @Value(METER_CODE)
```



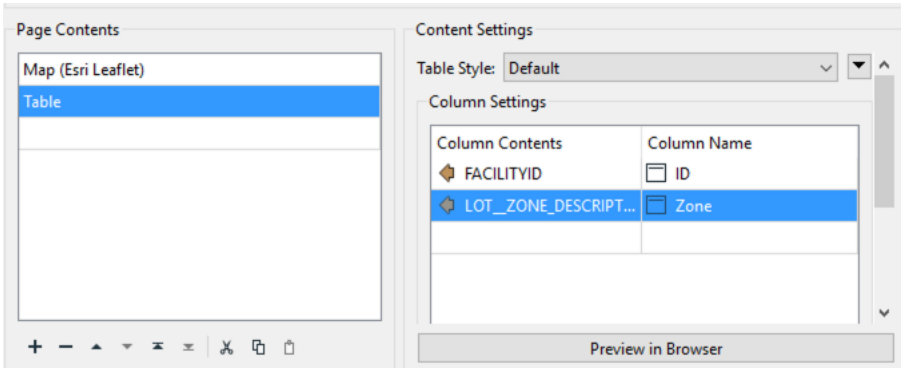
4. Create HTML Report

Now that we have our label created, we can create the HTML report. Add an HTMLReportGenerator to the canvas and connect it to the AttributeCreator. Set the first Page element to Map (Esri Leaflet), then click anywhere to update the parameters on the right. Set the Label Attribute to the Label attribute we just created and then set the Escape HTML to No.



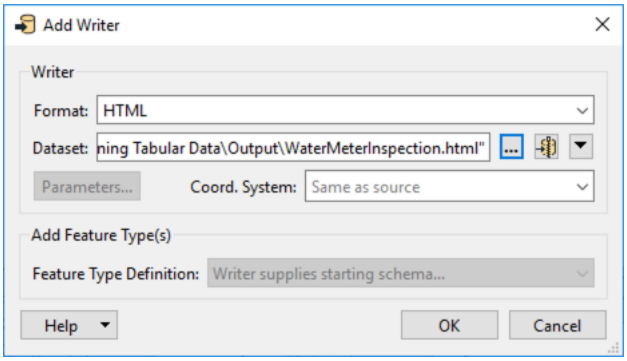
Next, add a Table Page element by clicking on the plus sign (+), click anywhere to update the parameters on the right. Configure the table as follows:

Column Contents	Column Name
FACILITYID	ID
LOT_ZONE_DESCRIPTION	Zone

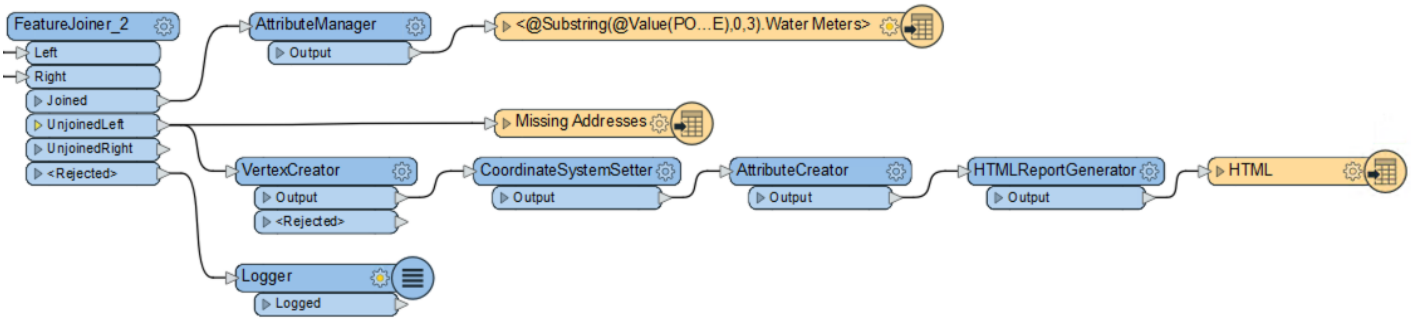


5. Add HTML Writer

Add an HTML writer to the canvas and browse to the Merging and Joining Tabular Data\Output folder. Name the file WaterMeterInspection.html.

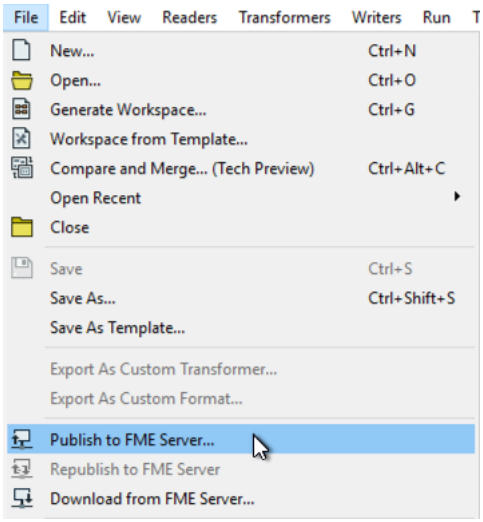


Connect the HTMLReportGenerator to the HTML writer feature type.



6. Publish to FME Server

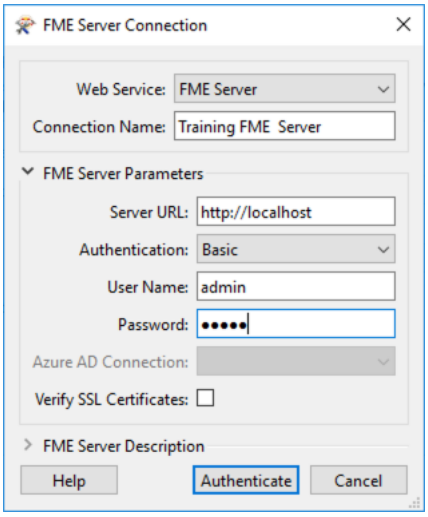
Save the workspace, then publish the workspace to FME Server by going to File > Publish to FME Server on the top menu bar.



In the Publish to FME Server wizard, click the drop-down next to Connection and select Add Web Connection. Enter in the following parameters:

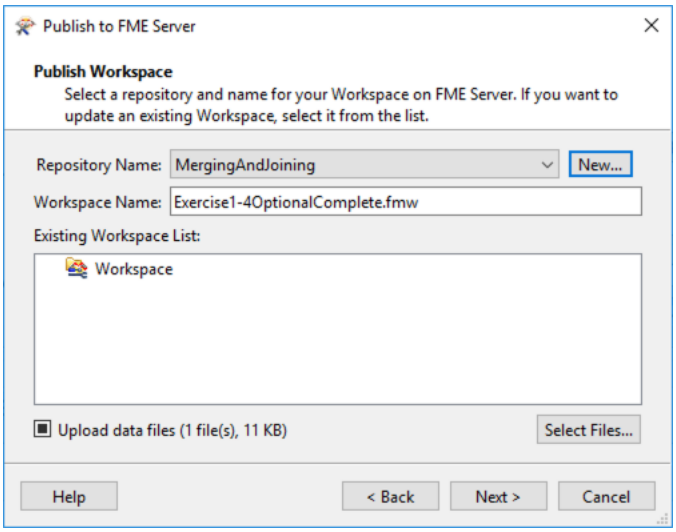
- **Connection Name:** Training FME Server
- **Server URL:** http://localhost (http://localhost)
- **Authentication:** Basic
- **Username:** admin
- **Password:** FMELearnings

Then click Authenticate. Once authenticated, click Next.

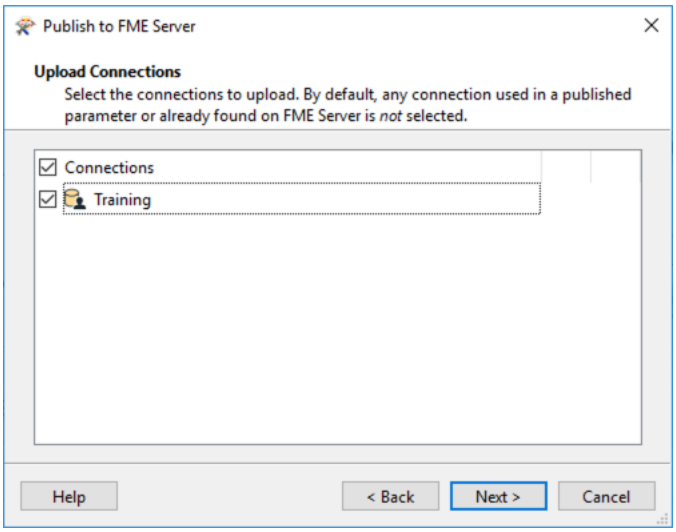


Note: If you are not using a Safe Software training machine, these credentials may be different. Please contact your FME Server Administrator for credentials.

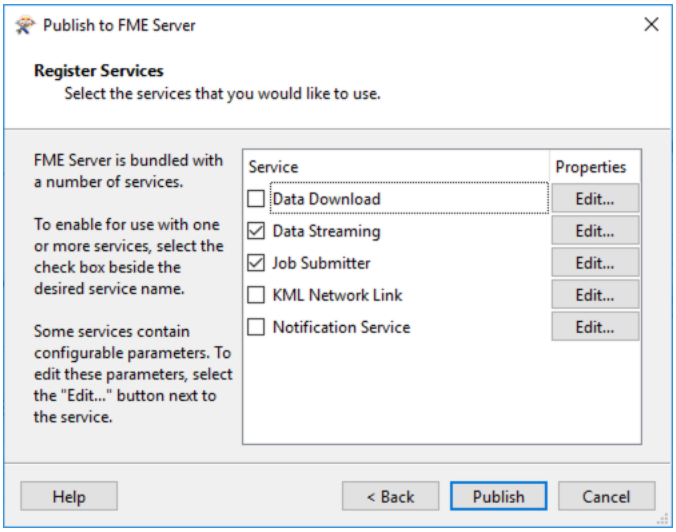
Create a New repository called MergingAndJoining, then confirm that Upload Files is enabled. Click Next to continue.



Upload the Training PostgreSQL database, by enabling the check box, then click Next.



Finally, select the Data Streaming Service, then click Publish. You can leave the Data Submitter Service enabled as well.



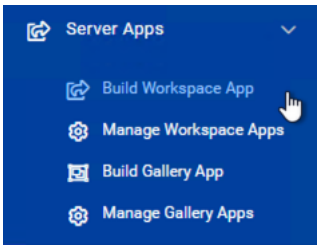
The workspace is now published to FME Server.

7. Build Workspace App

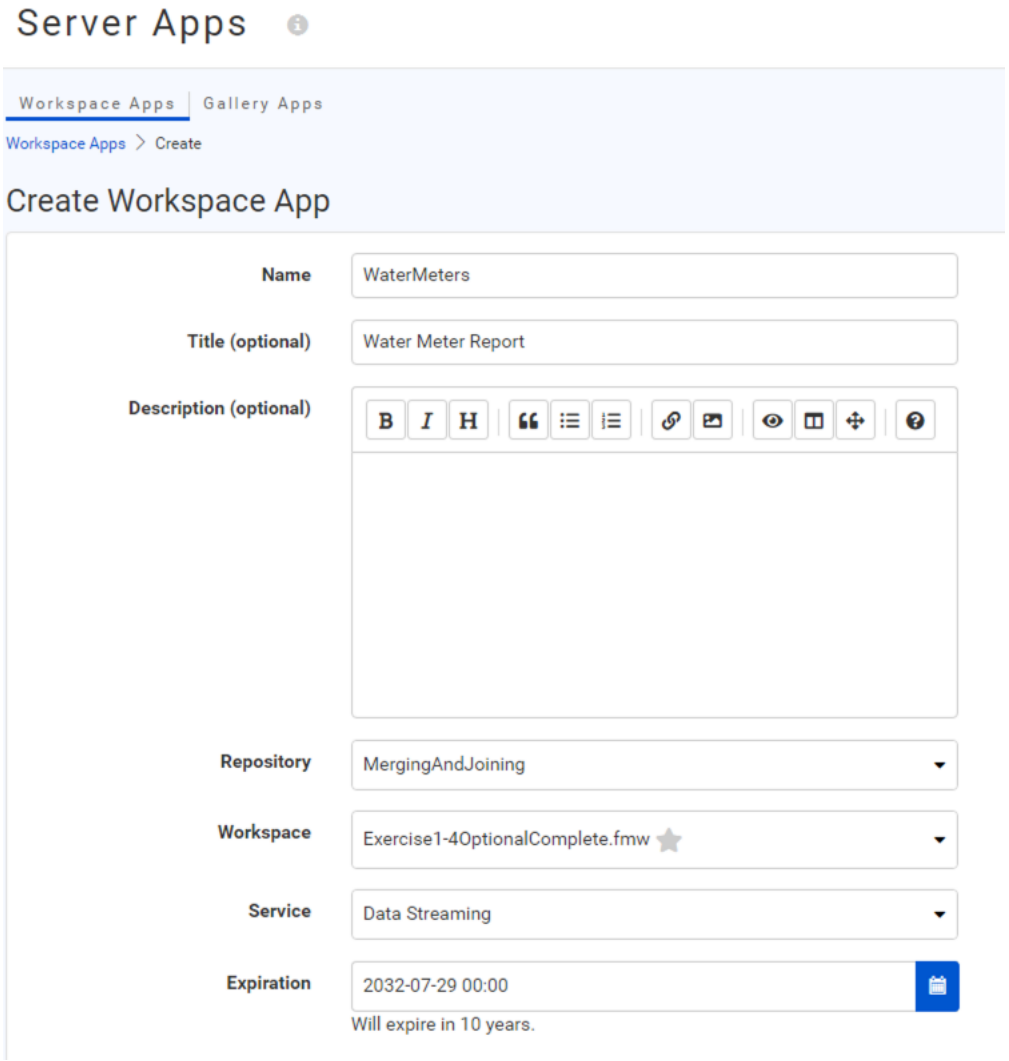
Open FME Server by either opening a web browser and typing in http://localhost (http://localhost) (or your Named Server) or by going to Start > FME Server > FME Server Web Interface (in Windows).

Login using the admin/FMELearnings credentials (or your FME Sever Administrator supplied credentials).

Expand Server Apps on the left menu bar, then go to Build Workspace App.



On the Create Workspace App page, set the Name to WaterMeters then set the Title to Water Meter Report. Change the Repository to MergingAndJoining. Confirm that the Service is set to Data Streaming, then click OK to create the app. If you have extra time, customize the app in the Customize section.



After clicking OK, a URL will appear. Click on the URL to open the app.

Additional Resources

Webinar: Data Integration Basics | Merging & Joining Data (https://www.safe.com/webinars/data-integration-basics-merging-joining-data/): the exercises in this tutorial were inspired by a demo from this webinar

Tutorial: Merging or Joining Spreadsheet or Database Data (https://community.safe.com/s/article/working-with-database-transformers-1)

The FeatureJoiner Transformer tutorial (https://knowledge.safe.com/articles/67526/the-featurejoiner-transformer.html)

The FeatureMerger Transformer tutorial (https://knowledge.safe.com/articles/59539/the-featuremerger-transformer.html)

The InlineQuerier Transformer tutorial (https://knowledge.safe.com/articles/23532/inlinequerier-1.html)

The SQLExecutor and SQLCreator Transformers tutorial (https://knowledge.safe.com/articles/23528/sqlcreator-sqlexecutor-1.html)

Using the SQLExecutor to do a SQL Join (https://knowledge.safe.com/articles/19634/using-the-sqlexecutor-to-do-a-join.html)

Data Attribution

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

First Published Date
8/10/2022, 6:00 PM

Last Published Date
8/22/2022, 6:23 PM

Integration (/s/topic/0TO4Q000000...

Authoring (/s/topic/0TO4Q000000...

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

Sort by:

Latest Posts

nampreetatsafe (/s/profile/0050c00000DlADBAA3) (Employee) published a new version of this Knowledge.
2m ago (/s/feed/0D54Q00009jkDcfSAE)

Like Comment

Log In to Comment

nampreetatsafe (/s/profile/0050c00000DlADBAA3) (Employee) published a new version of this Knowledge.
August 10, 2022 at 6:14 PM (/s/feed/0D54Q00009iO0SzSAK)

Like Comment

Log In to Comment

nampreetatsafe (/s/profile/0050c00000DlADBAA3) (Employee) published this new Knowledge.
August 10, 2022 at 6:00 PM (/s/feed/0D54Q00009iNzTKSA0)

Like Comment

Log In to Comment

Follow

Files (1) (/s/relatedlist/ka14Q000001DX7jQAG/AttachedContentDocuments)

Merging and Joining Tabular Data
Aug 10, 2022 • 2.4MB • zip

View All (/s/relatedlist/ka14Q000001DX7jQAG/AttachedContentDocuments)

Related Articles

Merging or Joining Spatial Data (/s/article/performing-spatial-joins-and-merges)

Manipulation of Tabular Data (/s/article/tutorial-manipulation-of-tabular-data)

Combining Multiple Streams of Data (Append vs. Merge or Join) (/s/article/combining-multiple-streams-of-data)

Sorting Tabular or Graphical Data (/s/article/sorting-tabular-or-graphical-data)

Converting Tabular (Non-Spatial) and Spatial Data to CSV (/s/article/converting-non-spatial-and-spatial-data-to-csv)

SAFE SOFTWARE®
(https://safe.com)

Getting Started (../s/topic/0TO4Q000000QKioWAG/welcome)
Ideas (../s/bridea/acideasULT__brIdea__c/00B4Q000000fWlFEVJA1)
Feedback (https://forms.gle/LkoFWpziDYaWQkL78)
Forums (../s/forums/)
Groups (../s/group/CollaborationGroup/00Ba000000A0BxJEAV)
Knowledge Base (../s/knowledge-base/)
Support (../s/support/)

Register / Log In (/s/login/)

(https://twitter.com/SafeSoftware) (https://instagram.com/safesoftware/) (https://linkedin.com/company/safe-software) (https://youtube.com/user/SafeSoftwareFME) (https://safe.com/blog/)

Land Acknowledgement —

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