Maximizing the Value of Data

Learning Objectives

After completing this unit, you'll be able to:

- Understand how FME solves business problems.
- Define data integration.
- Define a data silo.
- Explain the value of spatial data integration.
- Define basic spatial data concepts: vector, raster, point, line, and polygon.

Video

Videos are provided in this module if you prefer to watch instead of reading the text below. Note that some Quiz answers might require you to read the text.

Powering the Flow of Data

Safe Software began in 1993, helping forestry companies exchange maps with the provincial government. It was technically possible to share the maps back then, but only after hours of manual work. Often, an incredible amount of information was lost in the process.

Nobody was happy. Safe Software created FME to address this problem and has been solving data challenges ever since.

What is FME?

FME (aka. Feature Manipulation Engine) is a data integration platform with the best support for spatial data worldwide. FME allows you to easily address the question of "where" and convert data precisely for your needs. You can author your own custom workflows that improve access to data and solve compatibility issues, without needing to code anything.

Data integration: the process of bringing together data from disparate sources in a unified view to create a dataset with both valuable and usable information.

Learn More

Throughout this trail, you will learn the FME essentials by completing hands-on problem-solving exercises. You will learn how FME helps you integrate data through three phases:

- **Connect** data and applications
- **Transform** data to fit your needs
- **Automate** workflows to run when you need them

By the end of the trail, you will be ready to create your own FME data integration workflows.

Breaking Down Data Silos

Data integration helps address the following challenges faced by many organizations:

- Data silos and separate software systems make accessing data difficult.
- Existing integration processes are manual and labor-intensive.

- Poor data quality means wasting time on manual data cleaning.

Data silo: data or databases that are maintained and used outside organization-wide data administration. Often they are associated with a single individual or department.

The City of Coquitlam uses FME to break down data silos across a number of departments and systems, including:

- Garbage & organics cart collection (IBM Maximo, Tempest, 3rd party data)
- LED streetlights conversion project (Excel, Oracle eBusiness Suite)
- Parking violations & ticketing (AMANDA)
- As-built drawings (OpenText document management system)
- Spatial data catalog (Amazon S3)

Learn more about the City of Coquitlam's use of FME in this blog post.

Removing data silos through data integration can:

- Improve data sharing and accessibility between internal teams and external vendors.
- Enable faster and more accurate decision-making.

The Value of Spatial Data Integration

To many, the words 'spatial data' translates directly to 'map'. Maps are certainly a great way to display spatial data, but there is much more spatial data is good for. After all, everything we see and do has some kind of spatial component. Where we live, how we travel - the list doesn't end.

By analyzing spatial data and how certain variables impact our lives, we can learn more about why certain spatial relationships exist. Why are certain locations popular travel destinations? Why does a brand do successfully in one country and not another? It's time to start adopting spatial and location data practices to better understand human behavior and our influence on our planet.

More and more organizations are producing and using spatial data. However, getting value out of that data through automating business processes and analyzing spatial patterns requires spatial data integration. Many data integration platforms exist, but FME provides the best support for spatial data integration.

Spatial data: data that is representative of a specific geographic location on the surface of the Earth.

Spatial data is often used with a **geographic information system (GIS)**, a system designed to capture, store, manipulate, analyze, manage, and present all types of spatial data.

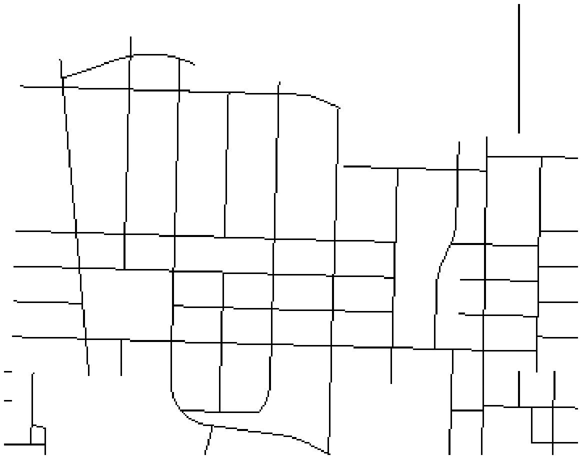
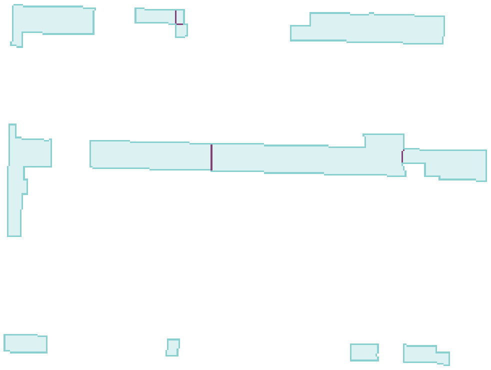
If you are already familiar with spatial data, you can skip ahead to the Quiz.

Learn More

Spatial data can be stored in vector or raster formats. Vector spatial data is made up of points, lines, and polygons. At its core, it consists of lists of coordinates on a plane and information regarding how they are connected.

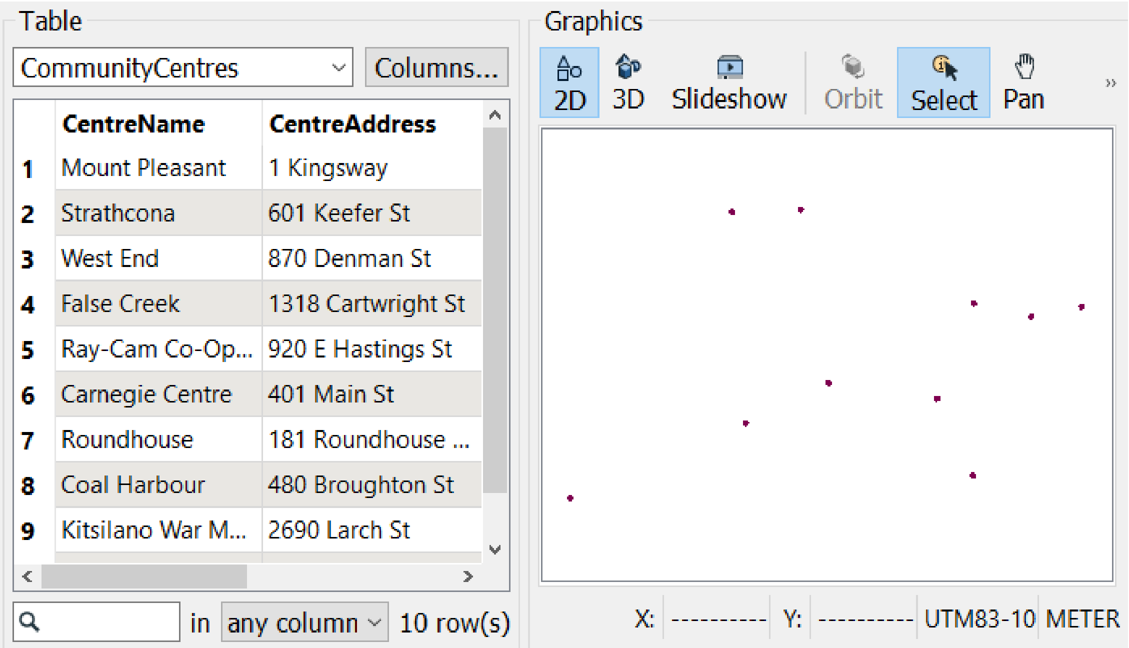
Raster data is made up of pixels, where the value (or color) of a pixel represents the value of a phenomenon, e.g. the average annual precipitation. Standard imagery uses the visible spectrum like a regular photograph. However, other kinds of imagery are available. For example, imagery in the near-infrared band can be used to measure the prevalence of vegetation beyond what the human eye can detect.

Raster data is better for representing continuous data that varies over an entire area, such as elevation. Vector data is better for representing discrete objects, such as the outlines of buildings (polygons).

Geometry type	Illustration	Examples
Point		Cell towers Community centers Fire hydrants Oil wells
Line		Electricity distribution network Pipelines Roads Trails Water distribution network
Polygon		Administrative borders Building footprints Service areas Water bodies Zoning districts
Raster		Classified land use Elevation Orthophoto (a satellite or aerial photograph adjusted so the scale is uniform) Scanned documents

Learn more about FME’s geometry model.

Spatial data contains geometry data that describes the actual location of the data. It also usually includes attribute data that describes the features. For example, here is a dataset of point locations of community centers that includes attributes such as “CentreName” and “CentreAddress”.



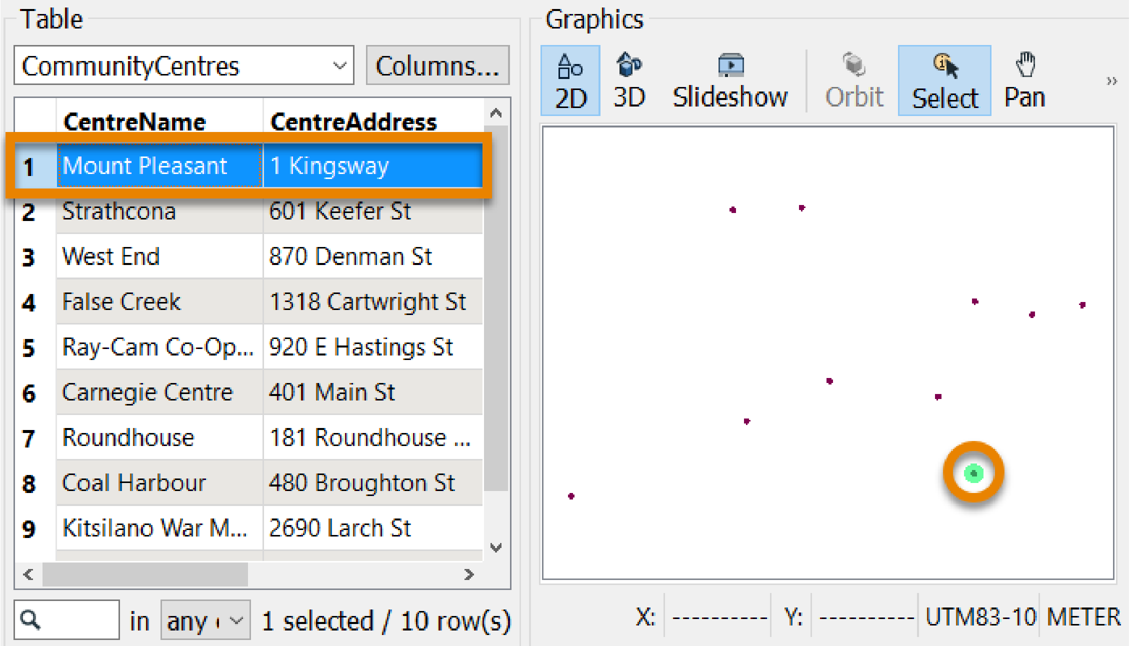
Table

CommunityCentres Columns...

	CentreName	CentreAddress
1	Mount Pleasant	1 Kingsway
2	Strathcona	601 Keefer St
3	West End	870 Denman St
4	False Creek	1318 Cartwright St
5	Ray-Cam Co-Op...	920 E Hastings St
6	Carnegie Centre	401 Main St
7	Roundhouse	181 Roundhouse ...
8	Coal Harbour	480 Broughton St
9	Kitsilano War M...	2690 Larch St

in any column 10 row(s)

Having both geometry and attributes in the same dataset lets you do things like query or filter the data. For example, you could filter the dataset of community centers to select the point that has a “CentreName” that equals “Mount Pleasant”.



CommunityCentres
Columns...

	CentreName	CentreAddress
1	Mount Pleasant	1 Kingsway
2	Strathcona	601 Keefer St
3	West End	870 Denman St
4	False Creek	1318 Cartwright St
5	Ray-Cam Co-Op...	920 E Hastings St
6	Carnegie Centre	401 Main St
7	Roundhouse	181 Roundhouse ...
8	Coal Harbour	480 Broughton St
9	Kitsilano War M...	2690 Larch St

in any 1 selected / 10 row(s)

X: ---------- Y: ---------- UTM83-10 METER

By itself, spatial data can be used to create maps or analyzed to identify patterns such as clusters. However, to gain maximum value from spatial data, it must be integrated with other data sources. For example, retail businesses integrate existing store locations, road networks, and neighborhood demographic data to identify the best place to build new stores.

It is also possible to integrate spatial with nonspatial data, such as spreadsheets or database tables. This integration is possible as long as the nonspatial data has an attribute that holds some spatial information. Many nonspatial datasets contain addresses, coordinates, or other identifiers. Combining these with spatial data allows you to unlock new insights. For example, retail businesses combine customer transaction data (containing their zip or postal code) with neighborhood demographic data to understand their customers and market their product or service more effectively.

For examples of how FME users are getting value from spatial data integration, check out these customer stories:

- Natural gas and electricity utility FortisBC uses FME to create an integrated view of wildfires and assets to inform decision making and send spatially aware notifications..
- The Iowa Department of Transportation uses Internet of Things data with FME to integrate snow plow locations, plow cams, and road conditions.
- The California Earthquake Authority nonprofit insurance company uses FME to monitor for earthquakes and automatically notify stakeholders when a seismic event occurs nearby.

1Which kind of platform is FME?

○ A.Data governance
○ B.Blockchain
○ C.Customer relationship management (CRM)
○ D.Data integration

2Which of the following is not an example of a data silo?

○ A.AutoCAD drawing datasets stored on a single CAD technician’s computer.
○ B.Excel files of budgeted financial statements stored on a network drive shared by a few members of an Accounting department.
○ C.An asset management system with data only accessible by members of the facilities management team.
○ D.All the spatial data used by an organization stored in a single cloud database, integrated with other systems where necessary.

3Which of the following is an example of spatial data?

○ A.An inventory management system database.

○ B.Employee work schedules in a spreadsheet.
○ C.Salesperson service areas as polygons.
○ D.PDF versions of quarterly financial reports.

4Which of the following scenarios doesn’t require spatial data integration?

○ A.A fast-food chain needs to find an optimal location for a new restaurant.
○ B.A construction company needs to automatically email invoices to its customers.
○ C.A local government needs to track the locations of graffiti reports to efficiently dispatch removal crews.
○ D.A weather network needs to provide customers with real-time notifications of lightning strikes, including coordinates.

5Which of the following spatial datasets is most likely to be stored using a raster format?

○ A.Aerial photographs of a dam site.
○ B.A road network with attributes such as the name of the street.
○ C.The location of fire halls.
○ D.The boundaries between counties within a state.

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Who Uses FME?

Learning Objectives

After completing this unit, you'll be able to:

- Understand where you fit among different kinds of FME users.
- Describe FME users' industries.
- Describe FME users' data types.

Video

Who Are FME Users?

People use FME for a variety of reasons. Most of our users fall into one or more of the following categories:

- **GIS Technicians**: professionals with a background in geography who work entry-level positions in GIS. Their day-to-day tasks include conducting spatial analysis, loading data in and out of GIS databases, and creating maps or GIS applications.
- **GIS Experts**: mid- or late-career professionals with a background in geography, computer science, or engineering who are highly advanced GIS users working on big cross-departmental projects. They often design the systems GIS Technicians implement and end up working with a wider variety of data types and applications.
- **Business Data Analysts**: professionals with a background in business analytics who normally work in programs like Excel and business intelligence tools like Qlik, Tableau, or PowerBI. They find value from FME because their data has gotten too big and complex to manage in spreadsheets, and they want to automate the data preparation process.
- **Citizen Integrators**: professionals with various backgrounds working with data who are trying to reduce their data integration costs and automate everyday tasks. Their solutions are tactical, and they value FME for being able to integrate data and applications quickly.
- **Integration Specialists**: professionals creating data and application integration solutions at an enterprise level. They often work in an IT department or are a consultant. Their solutions are strategic, and they value FME for creating durable and comprehensive integrations.

Which Industries Use FME?

FME is used by a variety of industries due to its flexibility and ease of use. Not only are 450+ formats supported, but custom content is always being uploaded by community members to FME Hub, which is free to access.

Airports & Aviation	Architecture, Engineering, and Construction	Commercial
Defense and Aerospace	Education and Non-Profit	Emergency Services
Energy	Federal Government	Health Care and Health Technology

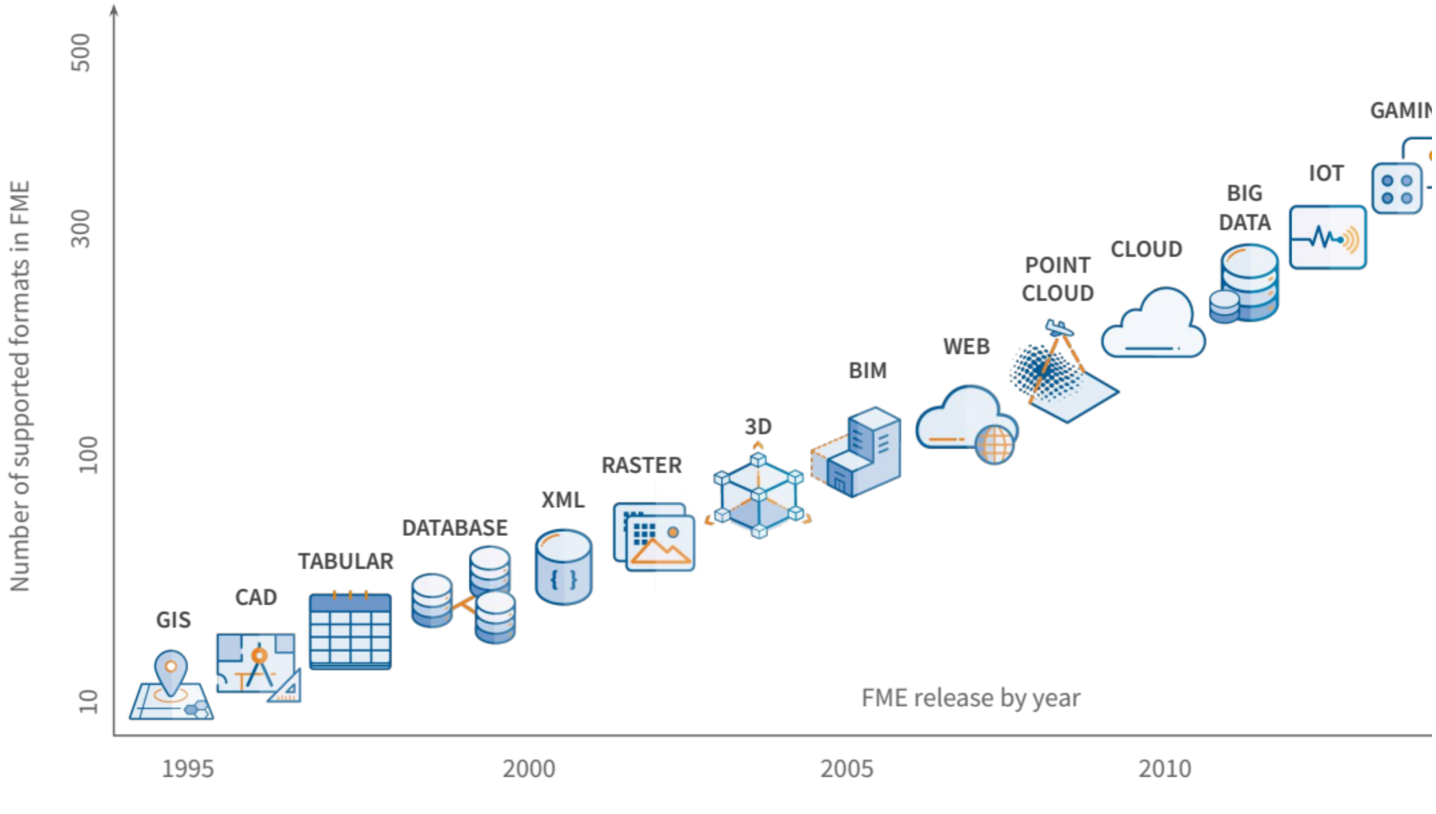
Local Government	Natural Resources	State Government
Telecommunications	Transportation	Utilities

We are in the process of creating scenario-based training content for these industries. In the meantime, check out the industry solutions pages or our Knowledge Base for more information. Interested in taking training for one topic in particular? Let us know with this quick survey.

You can get in touch with your industry peers in the FME Community.

What Data Types Does FME Support?

Similarly, FME supports a wide variety of data types. You can rely on FME to support core data types for spatial data integration, including CAD and GIS. You can also trust that FME will adapt to your rapidly-changing data integration needs, adding widely-used systems and cutting-edge technology.



Check out the data type solutions pages or our Knowledge Base for more information on working with these data types. Interested in training for one data type in particular? Let us know with this quick survey.

1Which person could benefit the most from FME?

○ A.Someone who wants to create an engaging digital slideshow.
○ B.Someone who wants to create a bar chart from spreadsheet data.
○ C.Someone who wants to combine data from a spreadsheet and a database, including deleting any duplicate entries.
○ D.Someone who wants to make a detailed custom wall map.

2Which of the following is a data type that FME has recently added support for?

○ A.Augmented/virtual reality
○ B.GIS
○ C.CAD
○ D.Databases

3How many data formats does FME support?

○ A.Less than 100
○ B.101-200
○ C.201-300
○ D.301-400
○ E.401-500

Check the Quiz to Earn 50 Points

Second attempt earns 25 points. Three or more earns 12 points.

Integrate Data Across Your Organization

Learning Objectives

After completing this unit, you'll be able to:

- Identify the different use cases for FME Desktop, FME Server, and FME Cloud.
- Describe how to maximize the value of your data using the entire FME platform.

Video

Data Integration Challenge

Jennifer has a problem. She works as a GIS Specialist for a city government. Her organization is facing a significant challenge: how can they get raw crime data provided by their police department into a GIS?

Let's find out how she can solve this data integration problem with FME through the steps of **connect**, **transform**, and **automate**.

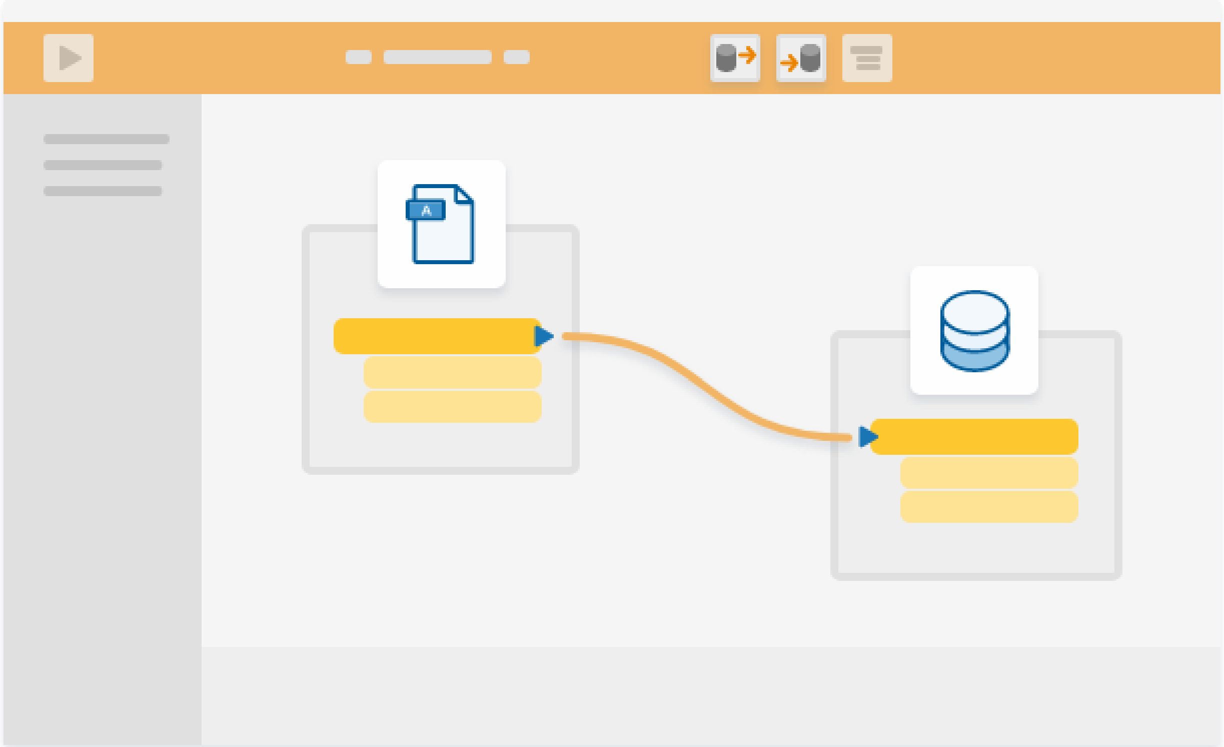
Connect

The crime data is in a MySQL database and needs to be loaded into an Esri ArcGIS geodatabase for both public and internal web mapping applications. The crime data does not have a coordinate system or any other spatial data associated with it except street addresses.

A **coordinate system** uses one or more numbers, or coordinates, to uniquely determine the position of a point or other elements on the surface of the Earth.

Learn More

Jennifer uses the desktop authoring software **FME Desktop** to connect the police crime data to a geodatabase. She does this by adding a MariaDB (MySQL) Reader and an Esri ArcGIS Geodatabase Writer, then clicking and dragging to connect them. Using FME, Jennifer doesn't have to write a single line of code.



Transform

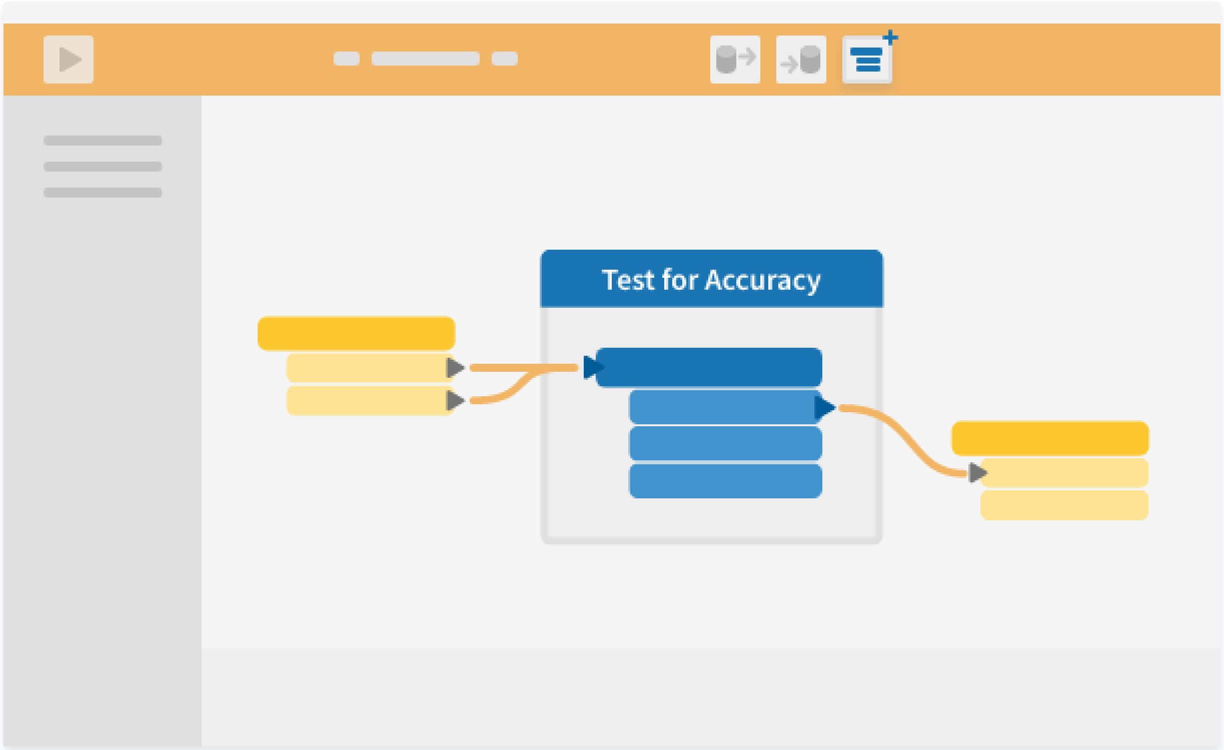
Next, Jennifer uses FME Desktop to author a workflow, which we call a **workspace**, that accomplishes multiple goals:

- **Improves data quality** using data validation transformers
- **Masks sensitive data** to ensure privacy protection
- **Edits the data's schema**, changing the attribute names and data types to match the geodatabase schema
- **Geocodes**, creating spatial data as points from the addresses that can be used in mapping applications

Using objects called transformers, Jennifer is able to restructure her data to fit the needs of her destination system.

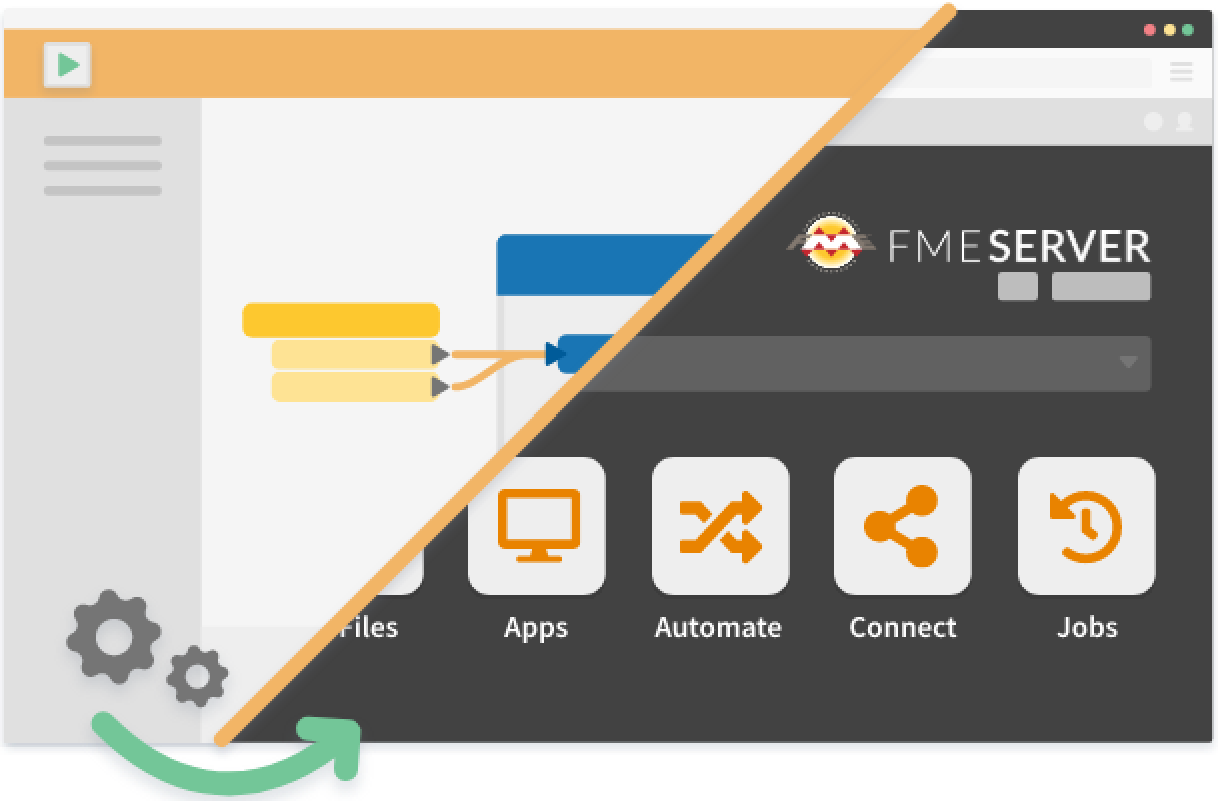
Schema: a formal definition of a dataset's structure, including table names, attribute names, and attribute data types (e.g. text, integer, float). You might also hear this called a data model.

Learn More



Automate

If Jennifer only needs to integrate this data once, she can run her workspace in FME Desktop and she’s done. However, if she needs to load data on a continuous basis, she can publish it to **FME Server**. With FME Server she can run her workspace automatically whenever new data becomes available. She also sets up email notifications to be sent to the FME Server Administrator when the workspace succeeds or fails.



If Jennifer didn’t want to install and configure FME Server on her own, she could use **FME Cloud** instead. FME Cloud is a deployment option for FME Server that is hosted by Safe Software.

1Which FME product lets you author repeatable data conversion and transformation workflows using a drag-and-drop interface?

○ A.FME Desktop
○ B.FME Server
○ C.FME Cloud

2Which FME product lets you automate workflows and run them on a schedule or in response to triggers?

○ A.FME Desktop
○ B.FME Server
○ C.FME Community
○ D.FME Hub

3Which FME service offers a hosted cloud environment for your data integration workflows?

○ A.FME Desktop
○ B.FME Server
○ C.FME Cloud

4After you author a workspace in FME Desktop, how do you move it to FME Server to automate it in an event-based workflow?

○ A.Push it

○ B.Publish it

○ C.Move it

○ D.Transfer it

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Get Help Learning and Using FME

Learning Objectives

After completing this unit, you'll be able to:

- Find help for using FME.
- Identify which resources will help you get to the next step.

Video

Getting Help with FME

Amar is a new FME user working as a GIS Technician for a city government. He just got started with the platform and has taken this FME Academy trail as part of his onboarding plan. He might come back to the Academy in the future for more advanced training. He also knows that self-serve training manuals, videos, and live training courses are available on safe.com.

The FME Community

Documentation

Now he's authoring his first workspace. As he uses FME, he refers to the documentation available in FME Desktop via the **Help** buttons. The documentation provides technical details about each transformer and data format. It is a great first resource when authoring a workspace.

FME documentation is also available online.

Knowledge Base

Amar works for a local government that uses Cityworks for asset management and permitting. He is responsible for integrating other systems with the platform, so he decides to check the Knowledge Base for a tutorial. The Knowledge Base contains tutorials and how-to guides on specific transformers and formats.

Forums

As he's working on his Cityworks integration workspace, he runs into some challenges and wants to see how other FME users have addressed them. He goes to the FME Community Forums and asks for help (view an example question). By providing details about his problem and a sample workspace, he makes it easy for other Community members to help. It's not long before Amar gets some guidance from his peers on the Community. They are able to solve his challenges together.

Ideas Exchange

As he's working more with his data, he gets some ideas for features that could improve FME Desktop. He goes to the FME Community Ideas page to suggest this as a product improvement. It turns out that other users have already suggested a related idea, so he upvotes it.

Live Chat

Amar's workspace is almost finished, but for some reason, he gets an error when he runs it. He tries to find help on the Forum, but this time everyone is stumped. So, he conducts some troubleshooting following this guide and then asks an FME expert on live chat for help to determine what might be going wrong. They are able to identify the issue. Problem solved!

File a Support Case

The expert let him know that if he runs into trouble in the future and isn't able to contact an FME Expert, he can submit a support case.

Now that he's used the Community to address his challenges, his workspace is finished! Amar's FME journey is off to a great start!

1Which resource should you use to learn the key skills and best practices for working with FME?

○ A.Ideas Exchange
○ B.Documentation
○ C.Submit a support case
○ D.FME training, including the FME Academy

2Which resource should you refer to first when trying to understand the technical requirements of working with a particular format or transformer?

○ A.Documentation
○ B.Ideas Exchange
○ C.Submit a support case

○ D.Calling Safe Software by phone

Check the Quiz to Earn 50 Points

Second attempt earns 25 points. Three or more earns 12 points.

Connect and View Data

Learning Objectives

After completing this unit, you'll be able to:

- Describe the FME platform user roles.
- Connect to a data source.
- Understand the concept of a feature type.
- View your existing data.

Learning content in the FME Academy presents the story of a user addressing their data integration challenges with FME. **You should follow along with their actions using your own installation of FME (2022.0 or later)** or request an on-demand virtual machine in the footer link below. Some units will require you to follow their steps or to take additional steps on your own to answer a quiz question.

A **Resources** section will provide you with links to interactive tutorials and starting workspaces when necessary.

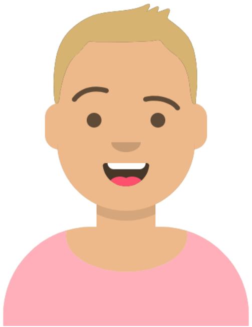
Resources

- BusinessOwners.xlsx
- Complete workspace

Video

Videos are provided in this module if you prefer to watch instead of reading the text below. Note that some Quiz answers might require you to read the text.

Starting an FME Project



Sven works as a Planning Analyst for a city's Economic Development department. He has been given an Excel spreadsheet containing point locations of businesses and needs to load it into an Esri geodatabase. He will use this business data to create guides for each neighborhood of the city to provide to residents, prospective business owners, and tourists.

Before beginning a new project with FME, he knows it is important to consider the different ways people interact with the FME platform. FME users can fall into one or multiple user roles:

- **Authors** create workspaces using FME Desktop and can publish them to FME Server An author can also create automated or notification-driven workflows in FME Server.

- **End-users** run workspaces using FME Workbench, FME Server, or an application powered by FME, but they do not create or edit workspaces. An end-user doesn't need much FME experience and, in the case of applications powered by FME behind the scenes, they don't even need to be aware they are interacting with FME.
- **FME Server Developers** implement features and services offered by FME Server into other applications. They primarily interact with the REST API and likely have a strong development-focused background. For example, an FME Server Developer might create an open data portal that lets the public pick datasets to download. This application is powered by FME behind the scenes, but the end-user doesn't know that, as they just interact with the website.
- **Administrators** install and maintain the FME platform. For FME Desktop, they just have to install the software and might help others with licensing. Administrating FME Server is a bit more involved, including managing security, scalability, and performance.

For this project, Sven is an **author**. He will create a workspace for himself using FME Desktop and doesn't plan on sharing it. Maybe later on he will upload it to FME Server for others to use, but not yet.

Connect to Your Data

Sven is starting with an Excel workbook (BusinessOwners.xlsx) with a single sheet. Each row is a separate business and has information about the business, including the name of the owner, the company name, the business license number, and the location of the business' primary address. He wants to keep all this information in his new geodatabase. Here are two sample rows of his data.

First	Last Name	Company	License Number	Longitude	Latitude
Elvis	Clay	Diam Industries	B347A2	-123.101472	49.2480941
Noelani	Curry	Mus Donec Associates	1991FF	-123.1318356	49.28042851

Sven begins his data integration project by opening FME Workbench (2022.0 or later) and clicking New to create a new workspace in FME.

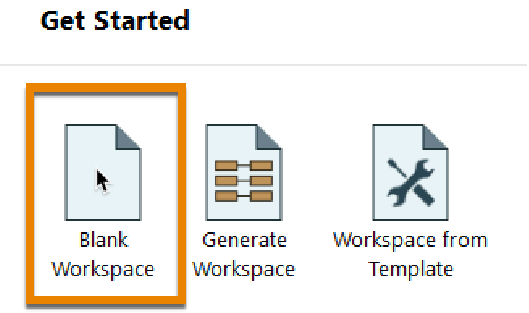
Most FME Academy modules assume you have access to FME Desktop and FME Server. **You should follow along with their actions using your own installation of FME (2022.0 or later)** or request an on-demand virtual machine in the footer link below. Some units contain an additional exercise challenging you to take additional steps on your own.

If you need access to FME, you can:

- Sign up for a free trial
- Request an on-demand virtual machine in the footer link below to use a virtual machine to gain temporary access to FME
- Ask your administrator for a license
- Purchase FME

If you don't have access to FME, some modules provide step-by-step tutorials you can follow to see the workflow in action. However, some Quizzes will require FME access to answer the questions.

All the data you need for the training is provided as links in the relevant units. If you want a collection of sample data, you can use the FME data repository (also on GitHub). Unless otherwise stated, the data used here originates from open data made available by the City of Vancouver, in British Columbia. It contains information licensed under the Open Government License - Vancouver.

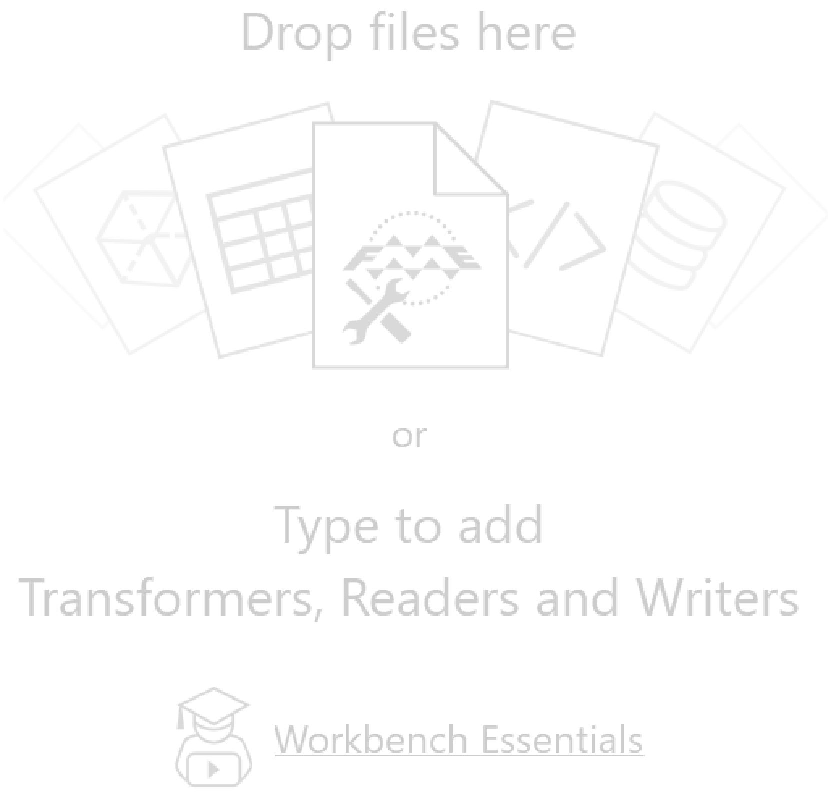


Clicking **Blank Workspace** opens the canvas, where a translation can be created. The canvas is currently blank, but a **reader** can be easily added to read the Excel dataset into FME.

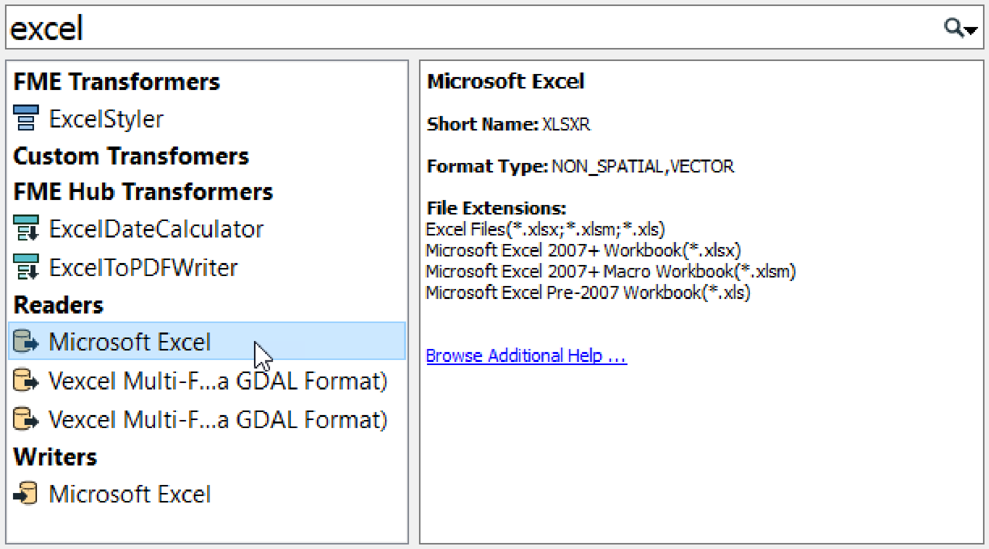
A **reader** is a component of a workspace that enables you to read a particular dataset.

Learn More

☆ **New for 2022.0**: the Start page has been revamped. You can now browse recent workspaces, favorite them, and download workspaces directly from FME Server.



Sven types “Excel” on the canvas. As soon as Sven starts typing, the Quick Add dialog appears, and starts suggesting objects that match “Excel”. Listed are all objects that can be added to the canvas: transformers, readers, and writers. Under the **Readers** heading, Sven selects the **Microsoft Excel** reader.



excel

FME Transformers
- ExcelStyler

Custom Transfomers

FME Hub Transformers
- ExcelDateCalculator
- ExcelToPDFWriter

Readers
- Microsoft Excel
- Vexcel Multi-F...a GDAL Format)
- Vexcel Multi-F...a GDAL Format)

Writers
- Microsoft Excel

Microsoft Excel

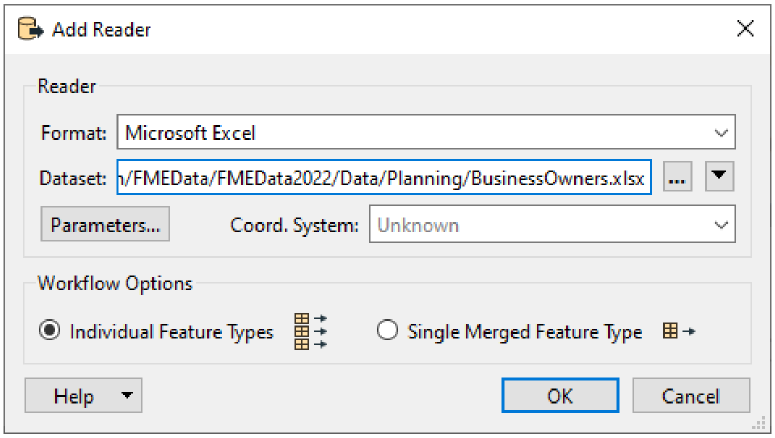
Short Name: XLSXR

Format Type: NON_SPATIAL,VECTOR

File Extensions:
Excel Files(*.xlsx;*.xlsm;*.xls)
Microsoft Excel 2007+ Workbook(*.xlsx)
Microsoft Excel 2007+ Macro Workbook(*.xlsm)
Microsoft Excel Pre-2007 Workbook(*.xls)

Browse Additional Help ...

The **Add Reader** dialog appears. The **Format** is already set as Excel, but the other parameters still need to be set. Sven sets the **Dataset** parameter to the location of the Excel file, pasting in a URL: https://s3.amazonaws.com/FMEData/FMEData2022/Data/Planning/BusinessOwners.xlsx.



Add Reader

Reader

Format: Microsoft Excel

Dataset: ı/FMEData/FMEData2022/Data/Planning/BusinessOwners.xlsx

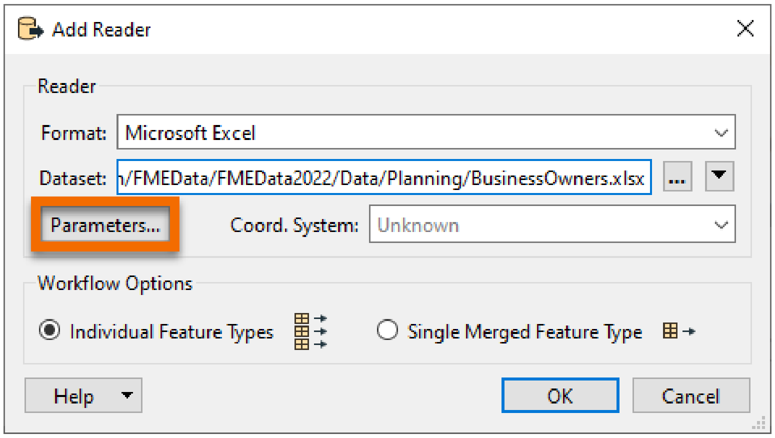
Coord. System: Unknown

Workflow Options

● Individual Feature Types ○ Single Merged Feature Type

The **Dataset** parameter can accept URLs or paths to files stored on your computer.

Sven then clicks on the **Parameters...** button.



Add Reader

Reader

Format: Microsoft Excel

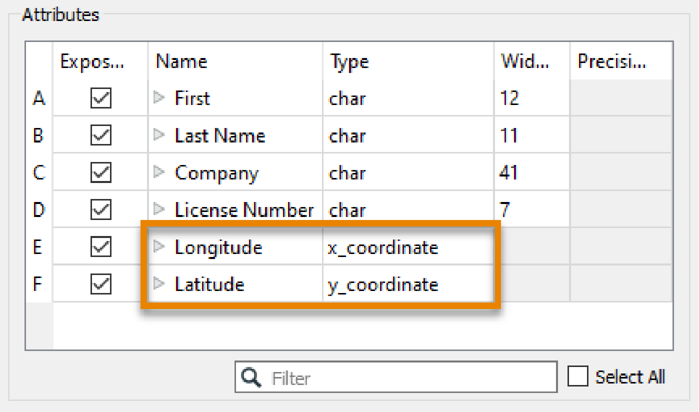
Dataset: ı/FMEData/FMEData2022/Data/Planning/BusinessOwners.xlsx

Coord. System: Unknown

Workflow Options

● Individual Feature Types ○ Single Merged Feature Type

The **Microsoft Excel Parameters** dialog controls how the Excel file will be read, including which sheet(s) to read. The **Preview** section displays how FME currently sees the data, while the **Attributes** section displays the attributes (spreadsheet columns, in this case) that FME has detected. The **Longitude** and **Latitude** attributes have been automatically detected as X and Y coordinates and set appropriately (under the **Type** column). They can be set manually if necessary. FME will automatically create points using these attributes when the spreadsheet is read.



Attributes

	Expos...	Name	Type	Wid...	Precisi...
A	☑	First	char	12	
B	☑	Last Name	char	11	
C	☑	Company	char	41	
D	☑	License Number	char	7	
E	☑	Longitude	x_coordinate		
F	☑	Latitude	y_coordinate		

Some datasets store geometry information and some do not. In this case, the source Excel file contained spatial data (latitude and longitude coordinates) describing the location of the address associated with each business license. However, in Excel these coordinates are just stored as numbers. To create a spatial dataset that can be analyzed and manipulated using FME or a GIS, the dataset needs to store geometry separate from its attributes.

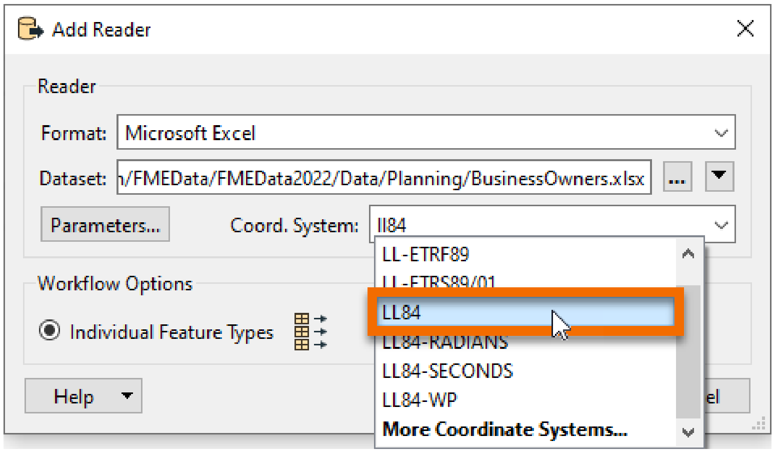
Learn More

The parameters all look correct, so Sven hits **OK** to close the **Excel Parameters** dialog. The final parameter to set is the **Coord. System** parameter.

The **Coord. System** parameter sets the coordinate system of the data. For some formats, the coordinate system information is contained in the data itself, and FME can read that automatically. The **Coord. System** currently says "Unknown", so Sven needs to set it.

Setting the coordinate system is not mandatory. It is necessary to set the coordinate system to use background maps when inspecting the data, to compare the data to other datasets in different coordinate systems, or to write to formats that require a coordinate system. FME will use the coordinate system stored by the dataset, if it exists. The **Coord. System** parameter will display "Read from source" if the dataset is capable of storing coordinate system information and will display "Unknown" if it is not. If you are unsure of the coordinate system of your data, check the metadata or contact the creator of the dataset.

Sven wants to use a background map when inspecting the data, so he sets the **Coord. System** to "LL84", a commonly used global coordinate system.



Add Reader

Reader

Format: Microsoft Excel

Dataset: ı/FMEData/FMEData2022/Data/Planning/BusinessOwners.xlsx

Coord. System: ll84

This coordinate system corresponds to the common 1984 World Geodetic System used by Global Positioning Systems (GPS).

Learn More

Now that all required parameters are set, Sven hits **OK** to close the dialog and add the reader to the canvas. The canvas now shows the single worksheet from the Excel spreadsheet: **BusinessOwners**.

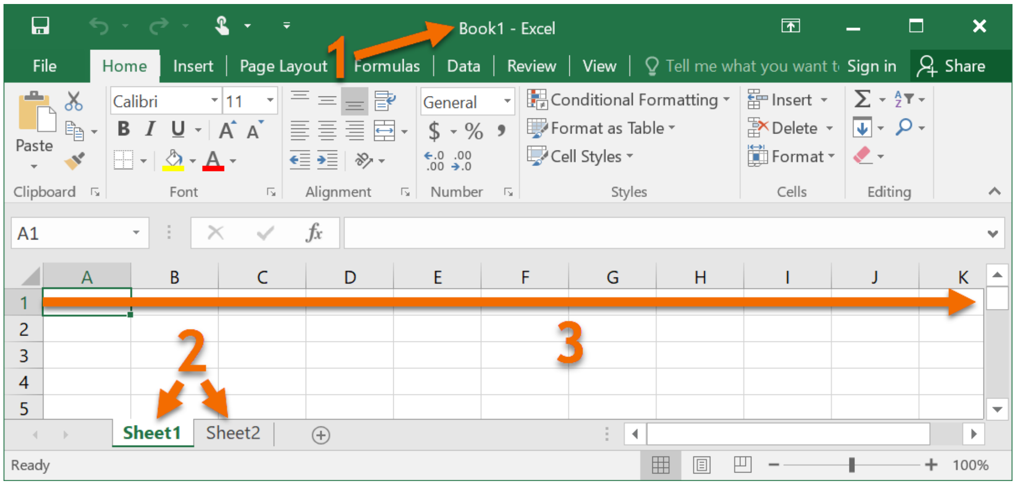
When you add a reader to FME Workbench, you choose which **feature types** you want to add to the canvas. A **feature type** in FME corresponds to a single sheet in a spreadsheet, table in a database, or a single layer (geometry plus attributes) in spatial data. For nested markup formats like JSON or XML, users create feature types by extracting a portion of the data or exploding it into a table. Because the Excel file only has one worksheet, Sven sees a single feature type added to his canvas. Feature types are children of readers or writers, so we will usually refer to them as reader feature types or writer feature types.

Feature types contain **features**. For spreadsheet or database data, a feature corresponds to a single row. For spatial data, it corresponds to a single piece of geometry (point, line, polygon, etc.) and its attributes.

Learn More

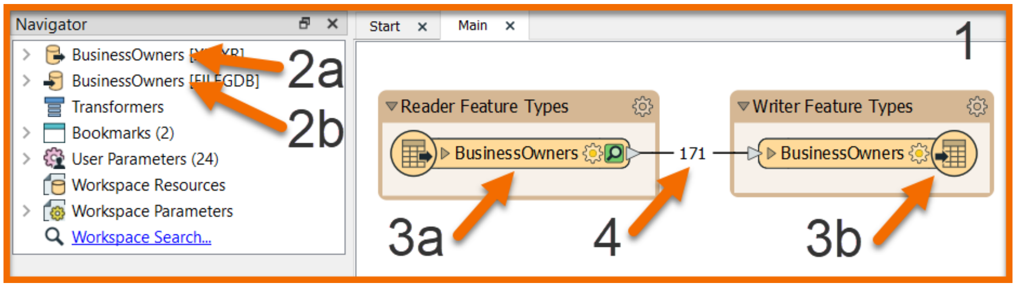
Here is a visual example of how FME components relate to Excel components. In the image below:

1. The **dataset** is the XLS or XLSX file (a.k.a. the workbook)
2. The **feature types** are the sheets (a.k.a. tables)
3. The **features** are the rows (the columns are the **attributes)**

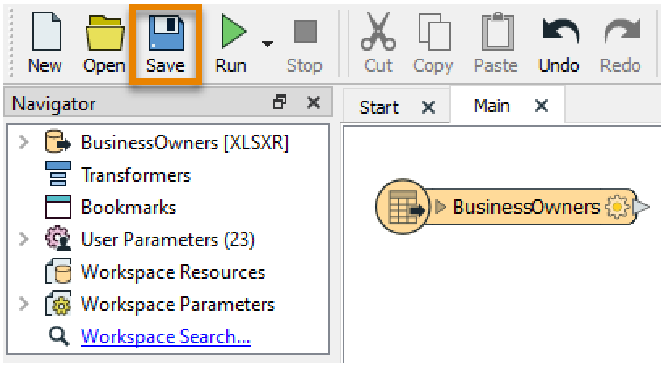


The components of a workspace are represented like this in FME Workbench. In the image below:

1. The entire **workspace**, consisting of the contents of the **canvas** and the **Navigator**.
2. **Readers** (a) and **writers** (b) at the top of the **Navigator**.
3. **Reader** (a) and **writer** (b) **feature types**, shown on the **canvas** and under their respective reader and writer in the **Navigator**.
4. **Features** (rows in a table or single pieces of geometry with associated attributes), shown as feature counts on connection lines after a workspace is run.



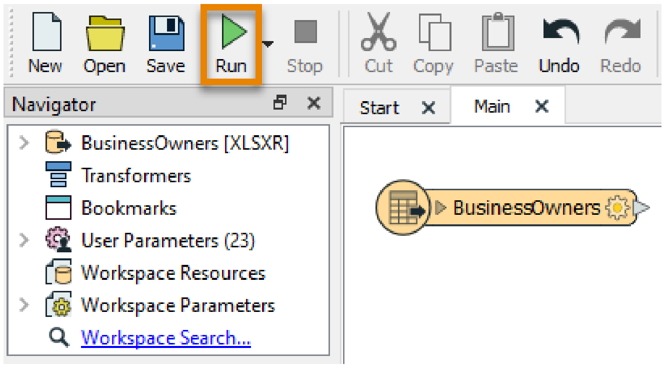
Sven saves the workspace by clicking on the **Save** button in the toolbar (the row of icons right below the menu bar) and using the default values provided. Now Sven can run the workspace.



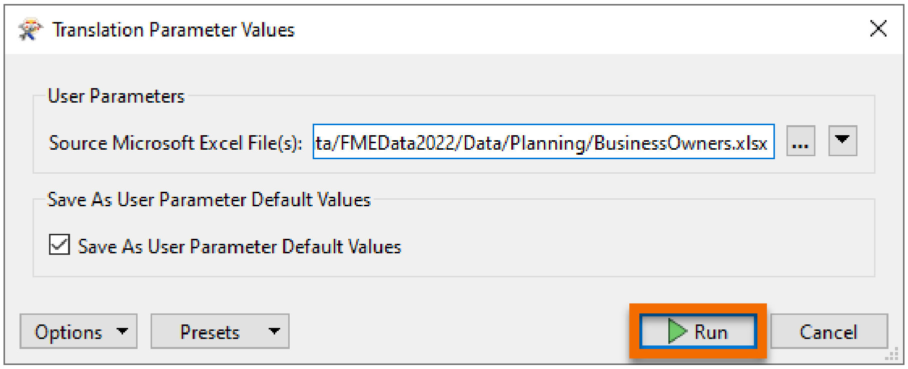
If you are following along, feel free to save your workspace wherever you like. We recommend saving often, including every time before you run your workspace. Note that workspace files do not contain any datasets.

Run the Workspace

Sven clicks on **Run** in the toolbar to run the translation.



A **Translation Parameter Values** dialog appears to confirm some parameters. This dialog can be helpful if Sven wants to change parameters before he runs his workspace.


Translation Parameter Values

User Parameters

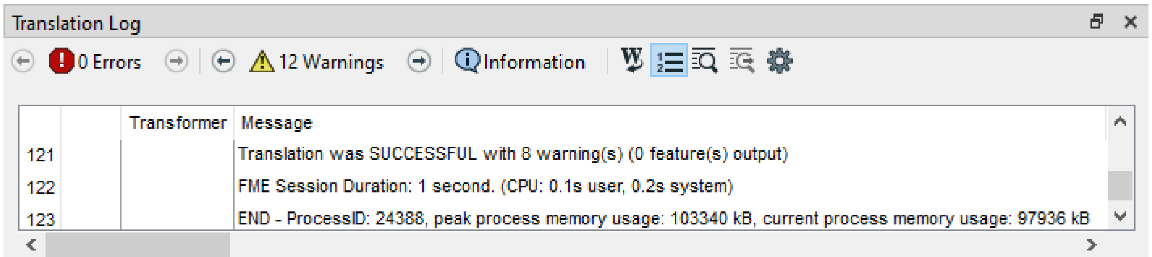
Source Microsoft Excel File(s): ta/FMEData2022/Data/Planning/BusinessOwners.xlsx

Save As User Parameter Default Values

☑ Save As User Parameter Default Values

Being prompted to fill in the parameter values can be useful if you want to rerun a workspace with different parameters. For example, you may be converting several datasets using the same workspace running multiple times, or testing if a workspace runs successfully with different input data. You may disable this prompt by clicking the drop-down triangle beside **Run** in the toolbar and deselecting **Prompt for User Parameters**.

After the workspace runs and the data is read, the **Translation Log** appears and reports what FME did during the translation and whether the translation was successful or not.

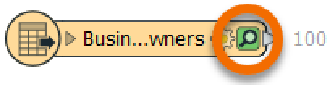


	Transformer	Message
121		Translation was SUCCESSFUL with 8 warning(s) (0 feature(s) output)
122		FME Session Duration: 1 second. (CPU: 0.1s user, 0.2s system)
123		END - ProcessID: 24388, peak process memory usage: 103340 kB, current process memory usage: 97936 kB

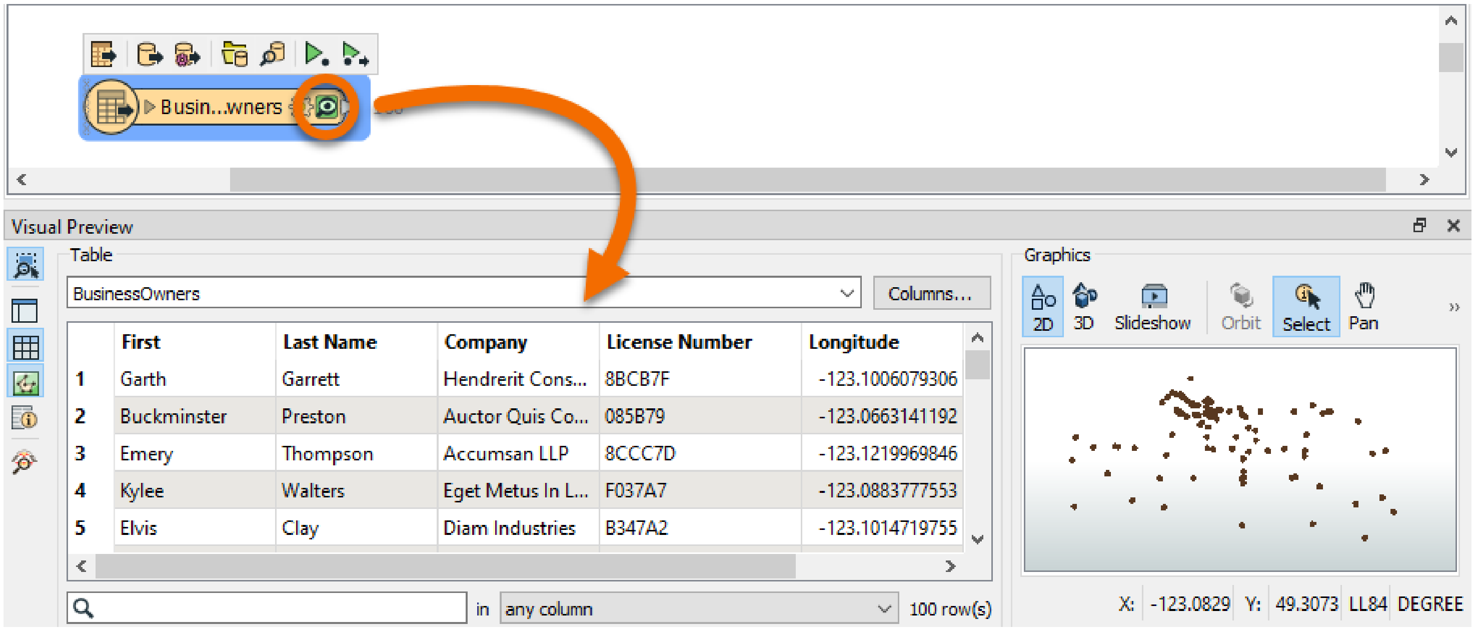
☆ **New for 2022.0**: the Translation Log is now displayed as a sortable table. Click the hyperlinked transformer name to navigate to the element on the canvas that's producing the message. This ability to identify where errors are occurring will make debugging workspaces more efficient.

View Your Data

On the **BusinessOwners** feature type, a copy of all of the features in the spreadsheet has been cached and can be inspected. Caches are a store of all of the features coming out of a particular port and are represented by the green magnifying glass icon. Sven clicks on the green magnifying glass icon to inspect the cached data.



Visual Preview displays a table containing all of the spreadsheet data. The total number of rows (features) is shown in the bottom right of **Table View**. **Graphics View** displays the spatial data, which in this case are points.



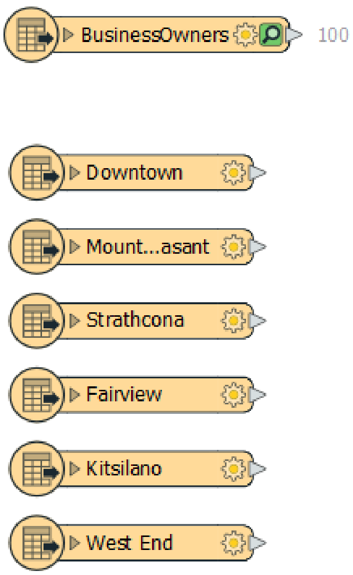
	First	Last Name	Company	License Number	Longitude
1	Garth	Garrett	Hendrerit Cons...	8BCB7F	-123.1006079306
2	Buckminster	Preston	Auctor Quis Co...	085B79	-123.0663141192
3	Emery	Thompson	Accumsan LLP	8CCC7D	-123.1219969846
4	Kylee	Walters	Eget Metus In L...	F037A7	-123.0883777553
5	Elvis	Clay	Diam Industries	B347A2	-123.1014719755

Sven notices that the **BusinessOwners** name in the feature type is truncated. He resizes it by double-clicking on its right edge.

Exercise

Now it’s your turn! Follow Sven's steps above to add the **BusinessOwners** feature type. Sven also wants to include data about public art in his neighborhood guides. Help him out by adding another Excel reader to connect to all the sheets (feature types) in this public art Excel workbook. Use **LL84** for the **Coord. System** again.

The workbook contains one sheet per neighborhood. Each row is a public art installation and contains information about the location, the title of the piece, and its longitude and latitude. **Tip**: you don’t need to download the file; you can just paste the URL into the **Dataset** parameter of the **Add Reader** dialog. Your canvas should now look like the image below.



1While you create an FME workspace for your own use, you are an:

○ A.Author
○ B.End-user
○ C.FME Server Developer
○ D.Administrator

2The Coordinate System parameter for the Microsoft Excel reader should be set if:

○ A.The data will be viewed with a background map
○ B.The data will be compared to other data in a different coordinate system
○ C.The writer format requires a coordinate system
○ D.All of the above

3For Microsoft Excel data being read by FME Workbench, a feature type is:

○ A.A Microsoft Excel XLS or XLSX file
○ B.A sheet in the Microsoft Excel file
○ C.A row in the sheet
○ D.A column in the sheet

4How many rows of data are in the BusinessOwners feature type?

○ A.56
○ B.100
○ C.125
○ D.268

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Write Data & Basic Troubleshooting

Learning Objectives

After completing this unit, you'll be able to:

- Create a new dataset.
- Connect objects on the canvas.
- Interpret the translation log to identify problems with your workspace.
- Fix an error by setting a writer feature type parameter.

Resources

- Starting workspace
- Complete workspace

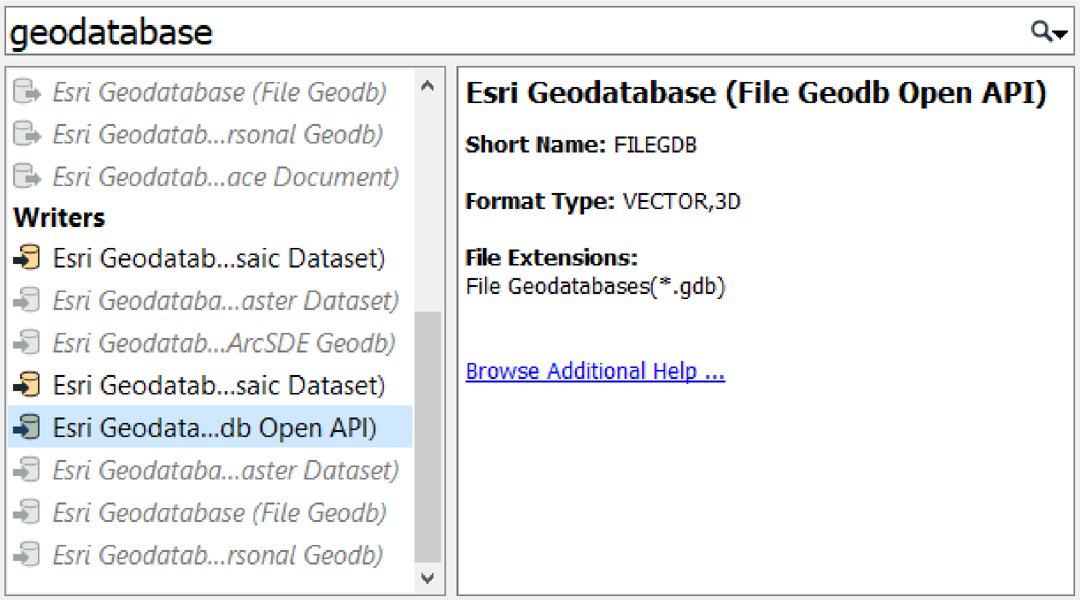
Don't forget to follow along with the steps described below using your own copy of FME. You'll need to complete the steps to answer quiz questions.

Video

Write Out Your Data

Now that Sven has read in his source Excel business license data, he has to add an Esri geodatabase writer to create his new dataset.

He opens the workspace he was working on in FME Workbench (2022.0 or later) -- it has readers for the "business owners" and "public art" datasets. He needs to add an Esri geodatabase writer. He clicks on the canvas and starts typing “geodatabase”. A large number of writer formats match “geodatabase”.


geodatabase
Esri Geodatabase (File Geodb)
Esri Geodatab...rsonal Geodb)
Esri Geodatab...ace Document)
Writers
Esri Geodatab...saic Dataset)
Esri Geodataba...aster Dataset)
Esri Geodatab...ArcSDE Geodb)
Esri Geodatab...saic Dataset)
Esri Geodata...db Open API)
Esri Geodataba...aster Dataset)
Esri Geodatabase (File Geodb)
Esri Geodatab...rsonal Geodb)

Esri Geodatabase (File Geodb Open API)

Short Name: FILEGDB

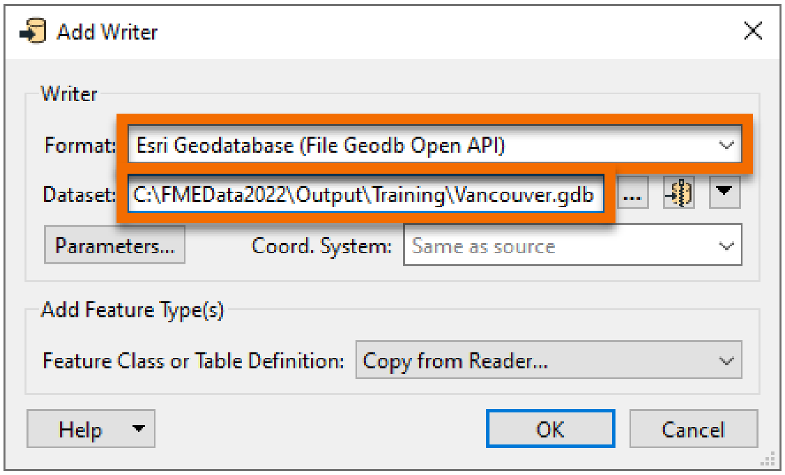
Format Type: VECTOR,3D

File Extensions:
File Geodatabases(*.gdb)

Browse Additional Help ...

FME supports many versions of Esri Geodatabases, however, many of them are only accessible to users who have ArcGIS licenses. The version we are using, Esri Geodatabase (File Geodb Open API) can be used by anyone.

Sven selects the Esri Geodatabase (File Geodb Open API) writer and starts filling out the **Add Writer** dialog. Under **Dataset** Sven enters the location he wants his data to be saved. He enters "C:\FMEData2022\Output\Training\Vancouver.gdb". All of the source datasets Sven will be working with are within the City of Vancouver and all of the writer feature types Sven creates with this workspace will be stored in the Vancouver.gdb file geodatabase.



Format: Esri Geodatabase (File Geodb Open API)

Dataset: C:\FMEData2022\Output\Training\Vancouver.gdb

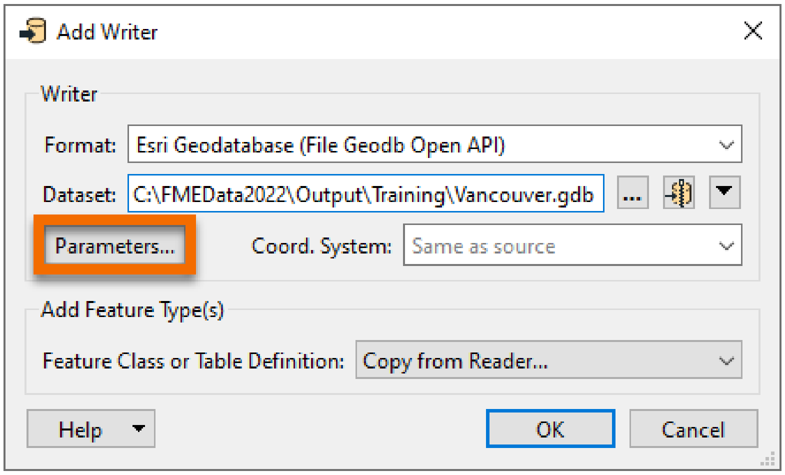
Coord. System: Same as source

Feature Class or Table Definition: Copy from Reader...

For readers, the **Dataset** parameter points to one or more datasets to read. For writers, it points to the file or folder where new data will be created or an existing dataset will be updated. If you want to change this after adding a reader or writer, you can do so in the **Navigator**.

You can use whatever path you want to write out your data, just remember your paths might be different from Sven's if you don't copy them exactly.

Next, Sven clicks **Parameters**.



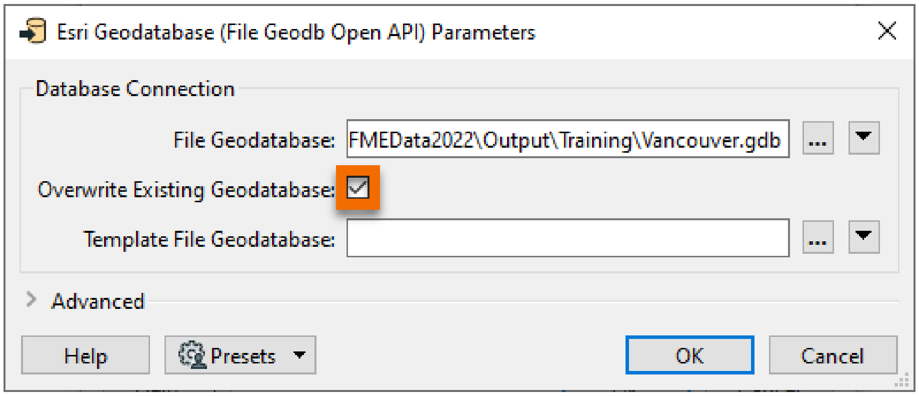
Format: Esri Geodatabase (File Geodb Open API)

Dataset: C:\FMEData2022\Output\Training\Vancouver.gdb

Coord. System: Same as source

Feature Class or Table Definition: Copy from Reader...

Clicking the **Parameters** button brings up the **Esri Geodatabase (File Geodb Open API) Parameters** dialog. Because Sven will be running this workspace often, and might be changing the schema of the output data due to changing requirements, Sven checks the **Overwrite Existing Database** option so that the geodatabase will be overwritten each time the workspace is run. Sven clicks **OK** to accept the changes made in the dialog.



Esri Geodatabase (File Geodb Open API) Parameters

Database Connection

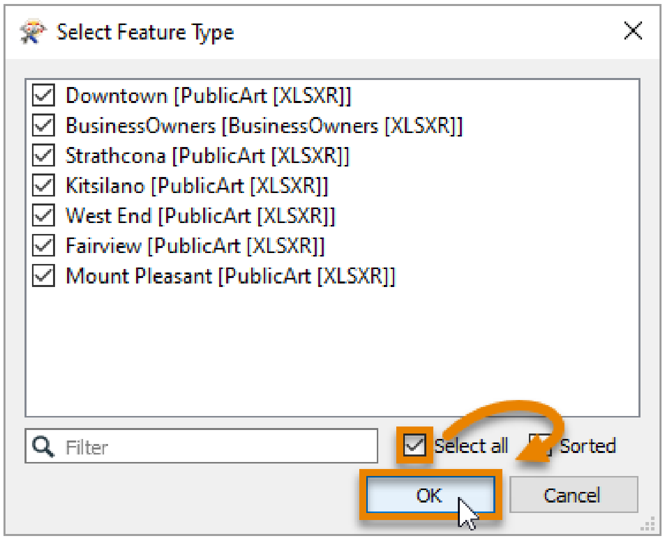
File Geodatabase: FMEData2022\Output\Training\Vancouver.gdb

Overwrite Existing Geodatabase: ☑

Template File Geodatabase:

The **Coord. System** is set as "Read from source". This means the written data will take the coordinate system of the input data, which is LL84. This information will be stored in the geodatabase.

Sven hits **OK** and is then prompted to select a reader feature type to copy the writer feature type definitions from. A writer feature type will be created for each reader feature type selected here. He clicks **Select All** to select all the PublicArt and BusinessOwners feature types and clicks **OK**.

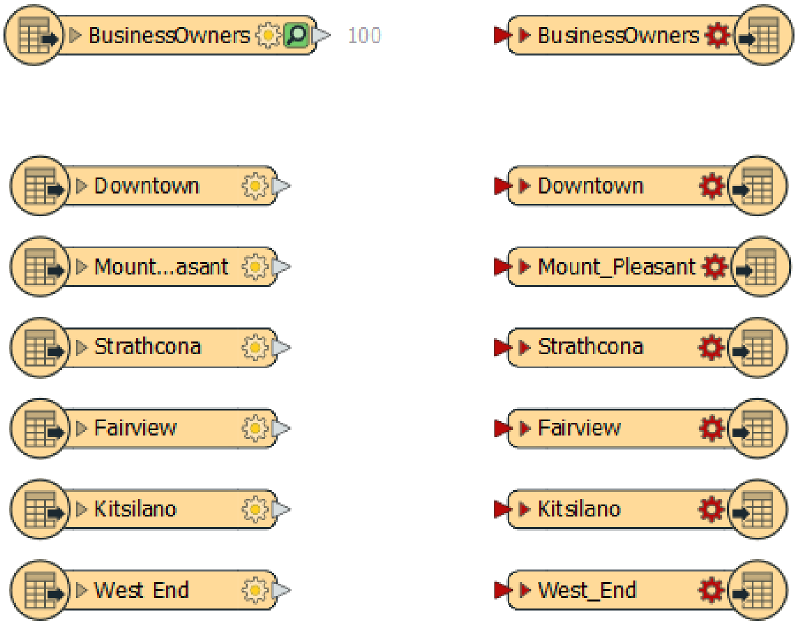


Select Feature Type

☑ Downtown [PublicArt [XLSXR]]
☑ BusinessOwners [BusinessOwners [XLSXR]]
☑ Strathcona [PublicArt [XLSXR]]
☑ Kitsilano [PublicArt [XLSXR]]
☑ West End [PublicArt [XLSXR]]
☑ Fairview [PublicArt [XLSXR]]
☑ Mount Pleasant [PublicArt [XLSXR]]

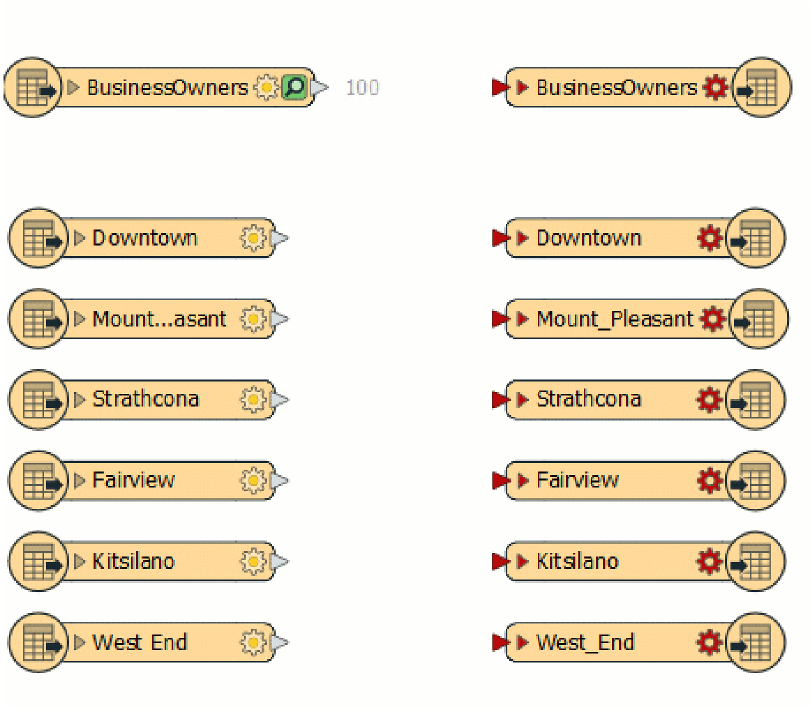
☑ Select all ☐ Sorted

The writer feature types are then added to the canvas.



Control the Flow of Data

Now Sven needs to connect the reader feature types to the writer feature types. Sven connects the **BusinessOwners** reader feature type to the corresponding writer feature type by clicking the reader feature type output port (gray triangle), and dragging his cursor to the writer feature type input port (red triangle). A new connection line appears.



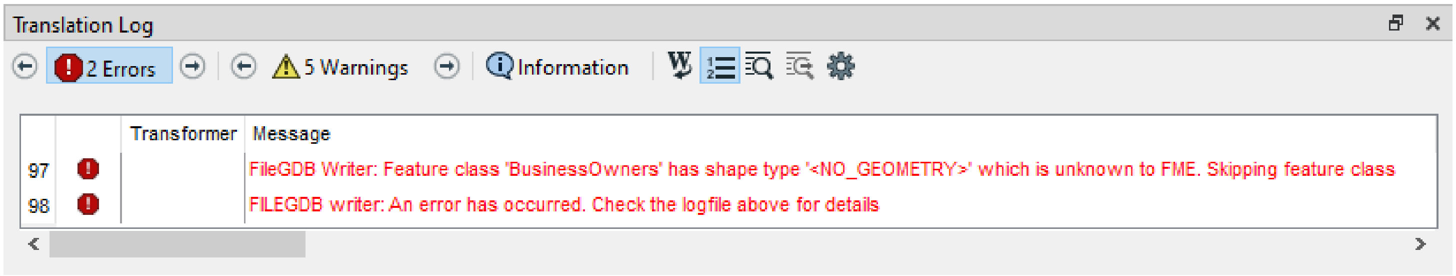
Sven ignores the PublicArt feature types for now. He wants to make sure the BusinessOwners feature type is working first. The workspace will still run with the PublicArt feature types disconnected. Sven then runs the workspace by clicking on the **Run** button on the toolbar.

Identify Problems Using the Translation Log

After the workspace finishes running, Sven notices that there are red entries in the log file shown in the **Translation Log**, meaning that errors have been reported. Sven clicks on **Errors** in the **Translation Log** to display errors only. Here Sven sees:

```
FileGDB Writer: Feature class 'BusinessOwners' has shape type '<NO_GEOMETRY>' which is unknown to FME. Skipping feature class
```

Copy



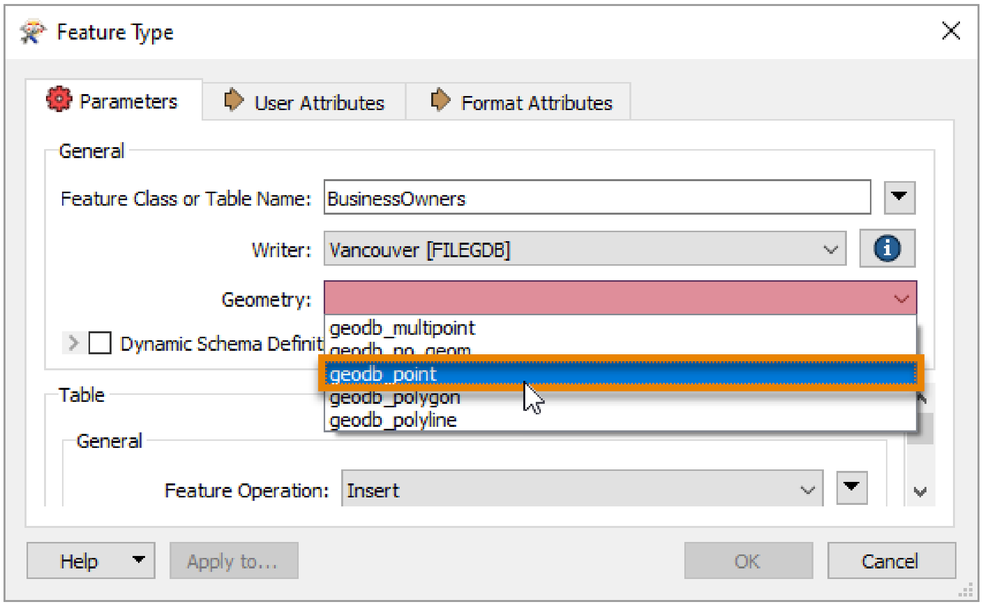
		Transformer	Message
97	❗		FileGDB Writer: Feature class 'BusinessOwners' has shape type '<NO_GEOMETRY>' which is unknown to FME. Skipping feature class
98	❗		FILEGDB writer: An error has occurred. Check the logfile above for details

This error occurred because the newly added BusinessOwners feature type requires a geometry type of geodb_point, but this has not been set yet. You can see it isn't set properly because the log reports its geometry type as "NO_GEOMETRY".

Fatal errors cause the workspace to stop running immediately. The "A fatal error has occurred" errors are caused by the first error in this case. Fixing the geometry will resolve all four errors. Errors that cause the workspace to stop running will often create additional error messages as the translation fails. Resolving the cause of the first error message will often resolve the subsequent error messages as well.

Fix the Error By Setting Writer Feature Type Parameters

To fix this error, Sven double-clicks on the **BusinessOwners** writer feature type to bring up the **Feature Type** dialog. He sets **Geometry** to "geodb_point" (the addresses are represented as points), and then clicks **OK**.



Feature Class or Table Name: BusinessOwners

Writer: Vancouver [FILEGDB]

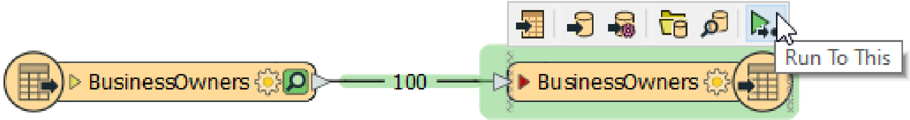
Geometry:

geodb_multipoint
geodb_no_geom
geodb_point
geodb_polygon
geodb_polyline

Feature Operation: Insert

Not all formats are limited to a single geometry type per feature type. Most formats will support different types of geometry on the same layer or feature type. But the Esri geodatabase formats only allow a single geometry type per feature class, so this parameter must be set.

He runs the workspace again by clicking on the "Run to this" icon on the writer feature type.



This time there are no errors reported in the **Translation Log**.

Exercise

Make sure you have followed along with Sven’s steps.

1The Feature Type dialog box can be opened by:

○ A.Double-clicking on the feature type on the canvas
○ B.Clicking on the cog-wheel button on the feature type on the canvas
○ C.Right-clicking on the feature type on the canvas and selecting "Properties" in the drop-down menu
○ D.All of the above

2In the Translation Log window, ERROR entries will be colored:

○ A.Black
○ B.Blue
○ C.Green
○ D.Red

3The error in Sven’s workspace was caused by:

○ A.FME being unable to convert Microsoft Excel data to an Esri file geodatabase.
○ B.Not connecting the correct feature types.
○ C.Not setting the Geometry parameter on the writer feature type.
○ D.Connecting features with incompatible geometry types to the writer feature type.

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Bring Together Multiple Streams

Learning Objectives

After completing this unit, you'll be able to:

- Set the schema for your output data.
- Combine data from different sources.
- Find your written data.

Resources

- Starting workspace

Video

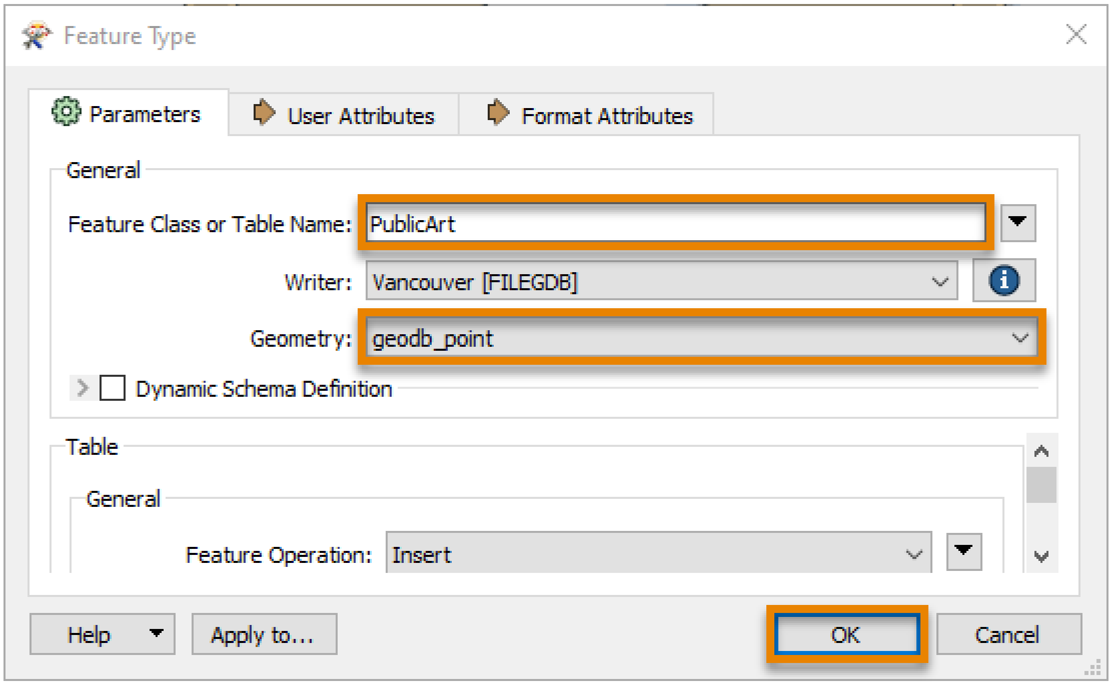
Combine Your Data Streams

When multiple streams connect to the same input port, the features accumulate. This operation is often called a **union**.

Sven opens FME Workbench (2022.0 or later) and continues to work on his Excel to geodatabase workspace. He realizes that right now, with a separate writer feature type for each public art feature type, he will end up with many layers in the geodatabase, one for each neighborhood's public art features. This corresponds to how the Excel file was structured, with one sheet per neighborhood. However, in the geodatabase, he'd prefer all the public art to be stored in one layer called PublicArt.

To create a single feature type, he could add a new feature type, but he prefers to just edit an existing one and then delete the others. He double-clicks on the Downtown writer feature type to open the **Feature Type** dialog and changes the **Feature Class or Table Name** to "PublicArt".

Sven also notices that the **Geometry** setting in the dialog is red like it was with the **BusinessOwners** feature type. He needs to define what kind of geometry he wants this feature type to use. He sets it to "geodb_point", as each art installation is represented by a point.



Feature Type

Parameters | User Attributes | Format Attributes

General

Feature Class or Table Name: PublicArt

Writer: Vancouver [FILEGDB]

Geometry: geodb_point

□ Dynamic Schema Definition

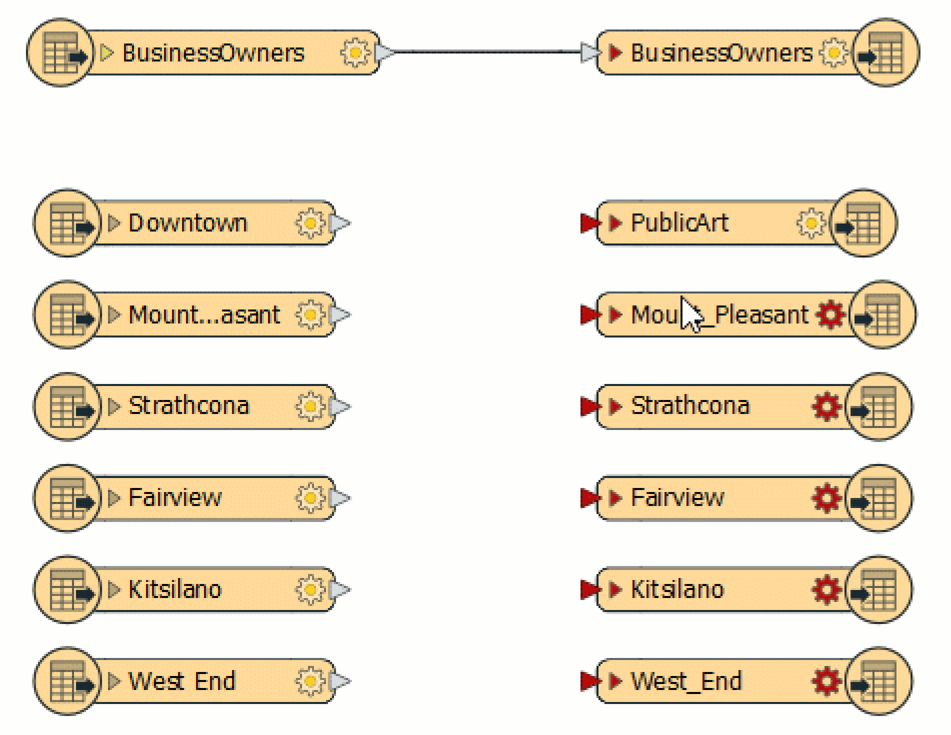
Table

General

Feature Operation: Insert

Sven clicks **OK** to accept the changes in the **Feature Type** dialog.

Sven then connects **PublicArt** to all of the neighborhood reader feature types by clicking on the red triangle going into PublicArt, holding down the CTRL key (⌘ or Shift on Mac), and clicking on each triangle coming out of the neighborhood feature types (**Downtown**, **Mount Pleasan**t, **Strathcona**, **Fairview**, **Kitsilano**, and **West End**).



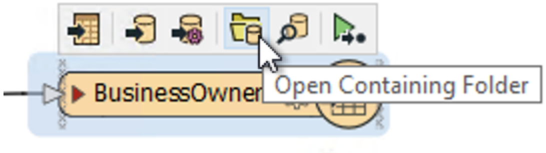
By creating these feature connection lines, Sven is telling FME to route all features from these sources into a single destination. When the workspace runs, features travel from left to right across the feature connection lines.

Sven can now delete the writer feature types that are not connected (**Mount_Pleasant**, **Strathcona**, **Fairview**, **Kitsilano**, and **West_End**). He clicks-and-drags a rectangle around them, then right-clicks one of them and selects **Delete**.

The workspace is now ready to be run. Sven clicks on **Run** in the toolbar.

Locate Your Output Data

Sven selects the **BusinessOwners** writer feature type and then clicks the **Open Containing Folder** icon. This opens the C:\FMEData2022\Output\Training folder where the data was written.



Sven sees the **Vancouver.gdb** file which contains the BusinessOwners and PublicArt feature classes (Feature classes are what feature types are called in a geodatabase). This confirms that the geodatabase was created in the correct folder and Sven expects that the BusinessOwners and PublicArt feature classes are within the geodatabase. In the next unit, he'll inspect it to ensure the contents of Vancouver.gdb are correct by checking that there are two feature types (BusinessOwners and PublicArt), they have the correct schema (attribute names, types, and allowed geometries), and they contain the correct number of features.

Exercise

Make sure you have followed along with Sven's steps.

1The name of a feature type always sets the name of the output file.

○ A.True
○ B.False

2Once you add writer feature types, you are free to connect them or delete them as you wish.

○ A.True
○ B.False

3After clicking on a port, holding down the CTRL key (⌘ or Shift on Mac) allows you to connect a feature connection to multiple ports.

○ A.True
○ B.False

4Navigating to the location of your written data requires remembering your file path and manually finding it using your operating system's file browser.

○ A.True
○ B.False

5What is the size of the written geodatabase to the nearest KB? Choose the closest answer, as results will vary slightly.

○ A.171
○ B.498
○ C.830
○ D.2,932

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Document Your Workspace

Learning Objectives

After completing this unit, you'll be able to:

- Understand why you should document your workspace.
- Add a bookmark to explain the purpose of sections of your workspace.
- Add annotations to your workspace to explain the purpose of specific feature types or transformers in your workspace.

Resources

- Starting workspace

Video

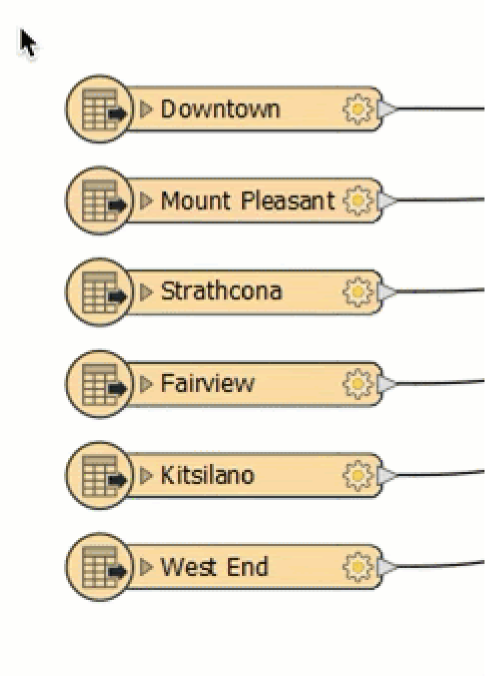
Why Document Your Workspaces?

Sven has created a workspace that converts his data from Excel to geodatabase. However, before he finishes working on this workspace, Sven needs to add bookmarks and annotation. Bookmarks and annotation provide documentation for your workspace. If you have any experience programming, documenting your workspace is like commenting your code. Taking these steps ensures that if your main FME author goes on vacation or leaves the company, documentation will be available to help the next FME author maintain, troubleshoot, and expand your organization's FME workspaces.

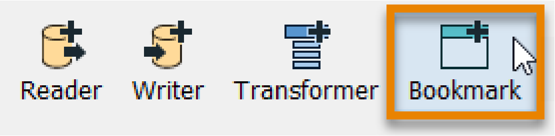
Adding Bookmarks

As a first step, Sven will use bookmarks to describe sections of his workspace. Bookmarks help you organize your workspace into logical sections. They make it easier to tell at a glance what each section of the workspace is doing. They also have some advanced functionality that is useful when working with large workspaces, such as the ability to zoom to a particular bookmark or to collapse bookmarks to reduce visual clutter.

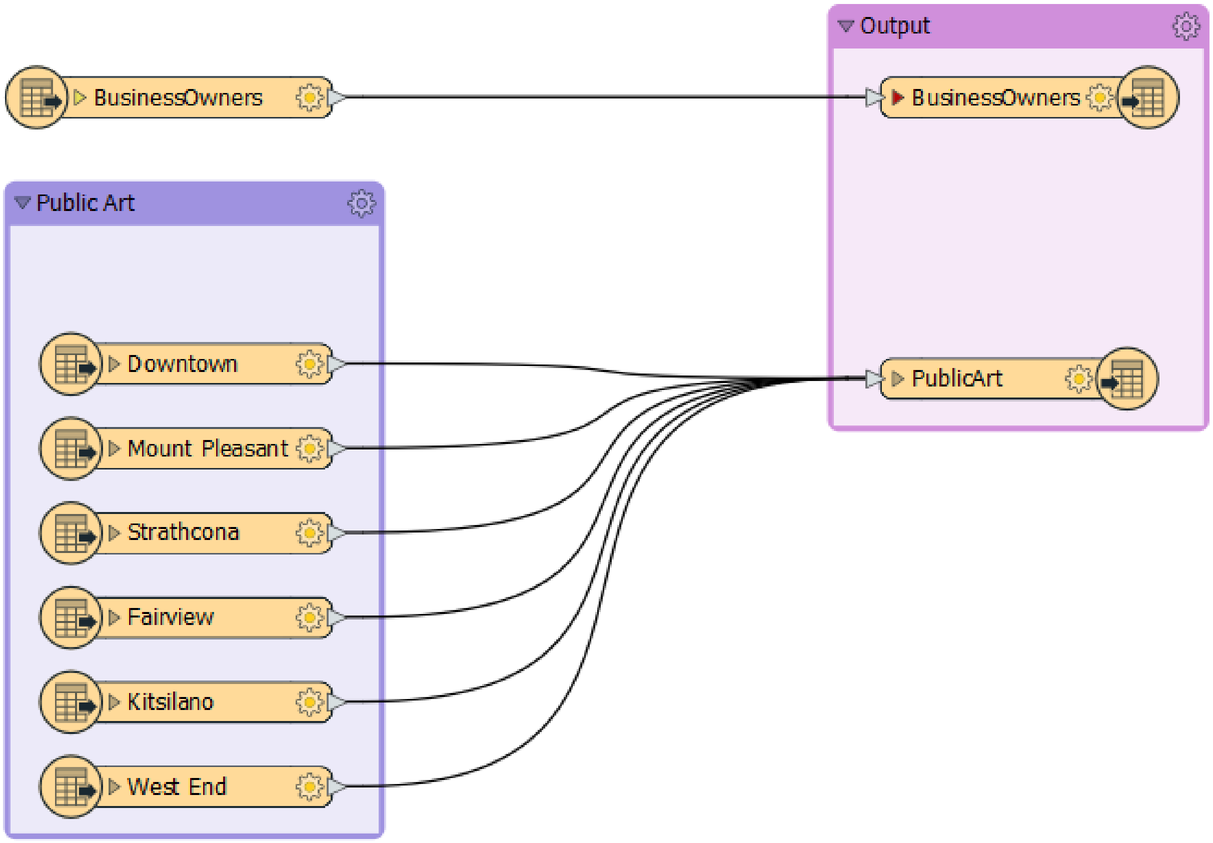
Sven opens FME Workbench (2022.0 or later) and adds two bookmarks to his workspace to surround his reader and writer feature types. He does this by selecting all the feature types he wants to include in a bookmark.



Then he clicks on **Bookmark** in the toolbar.

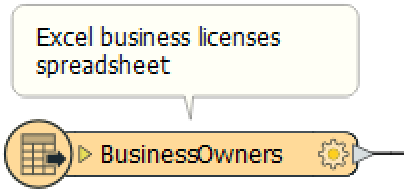


After clicking the button, his feature types are surrounded by the colored bookmark. He can then type in a name. He names the Excel reader feature type bookmark “Public Art” and the writer feature type bookmark “Output”.



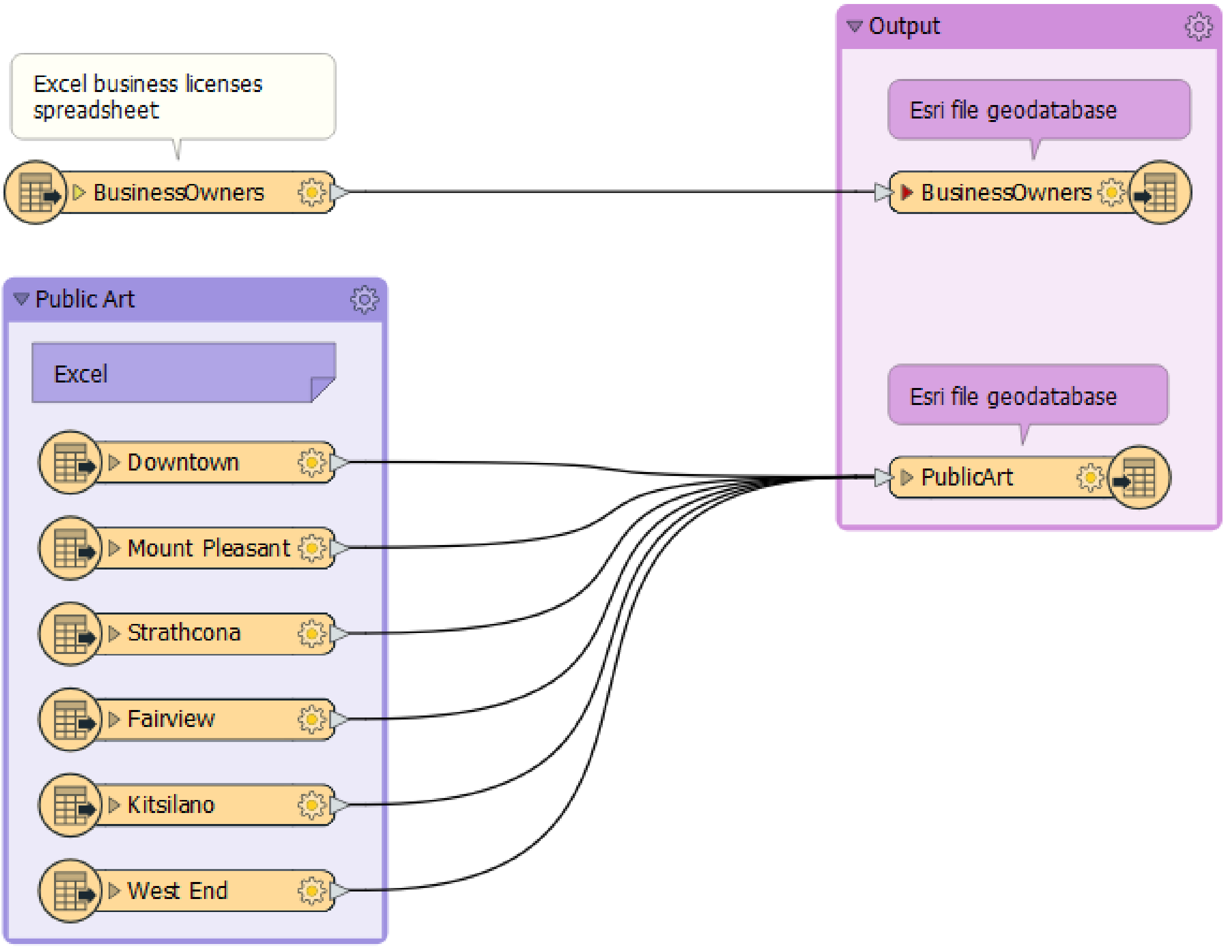
Adding Annotations

He also wants to add a note to describe what formats are being read. To do this, he adds an annotation by right-clicking on the **BusinessOwners** reader feature type and then clicks **Attach Annotation**. He edits it to say: “Excel business licenses spreadsheet”.



Excel business licenses spreadsheet

This annotation makes it easy to see what data is being read and written. He repeats the process for his other reader feature types, resizing his bookmarks by clicking and dragging their edges as needed.



Exercise

Make sure you have followed along with Sven’s steps.

1Bookmarks document sections of your workspace, while annotations document specific objects.

○ A.True
○ B.False

2Who might benefit from your use of annotation and bookmarks as part of best practice/style?

○ A.Other FME users in your organization
○ B.Customers to whom you deliver FME workspaces
○ C.Yourself in the future, coming back to edit the workspace
○ D.All of the above

Check the Quiz to Earn 50 Points

Second attempt earns 25 points. Three or more earns 12 points.

View Data With a Background Map

Learning Objectives

After completing this unit, you'll be able to:

- View your written data.
- View spatial data with a background map for context.

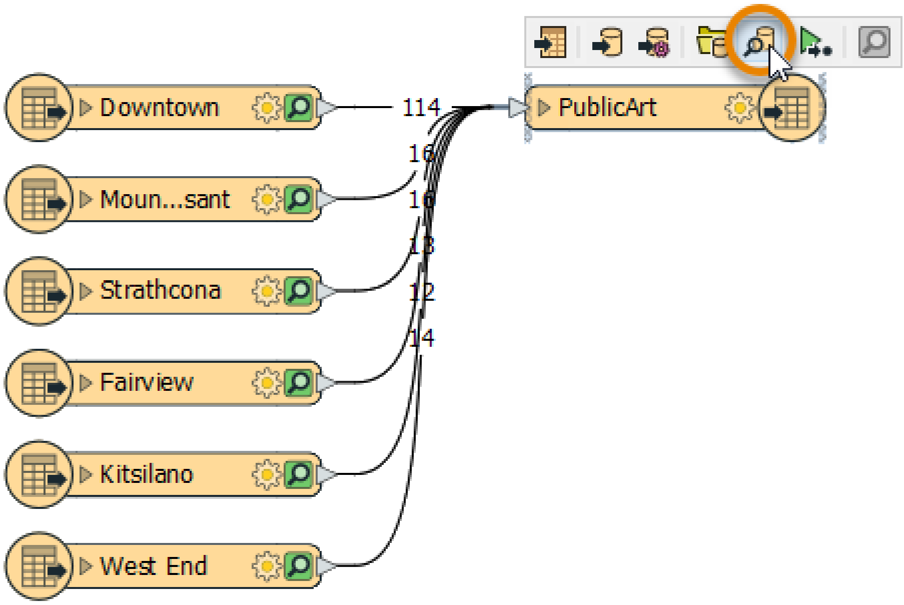
Resources

- Starting workspace

Video

View Your Written Data

Continuing with his workspace, Sven clicks **View Written Data** on the PublicArt writer feature type to open the geodatabase in **Visual Preview**.



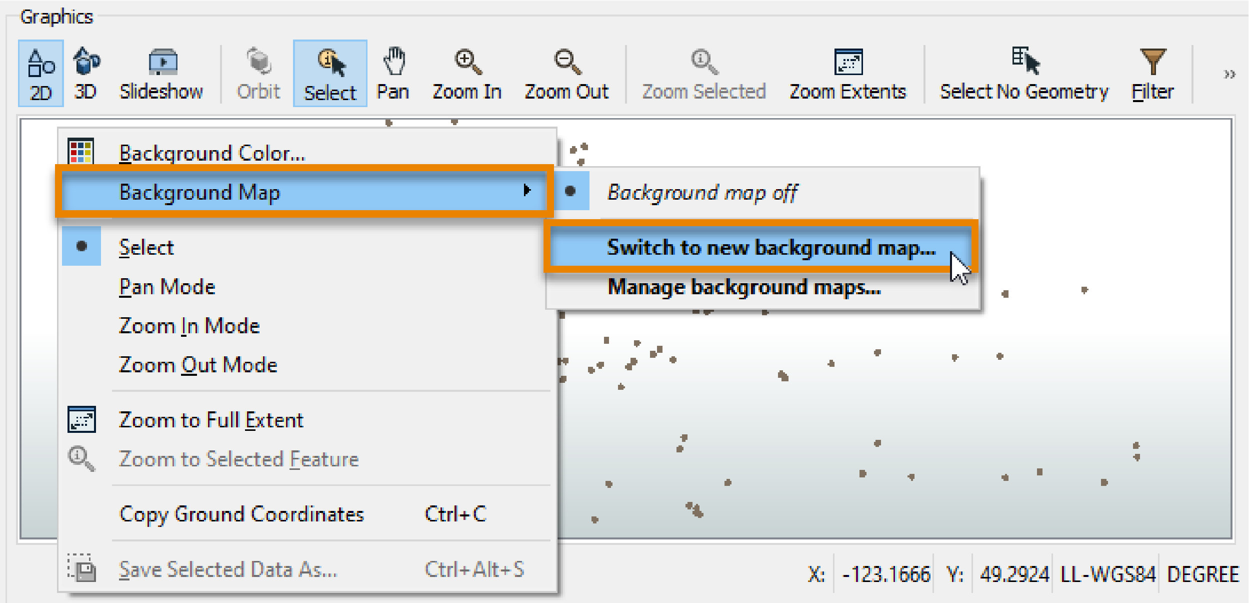
Add a Background Map

Sven can view the data now, but it is displayed without much context. He wants to get a general sense of where most art installations are located. He knows from experience that viewing his data with a background map can help with this task. In general, Sven likes to add a background map to spatial data when he wants to:

- see data in the context of common data he doesn't have on hand, like water bodies, streets, and parks
- compare locations against one another
- visually check for basic densities, aggregations, or patterns in the data

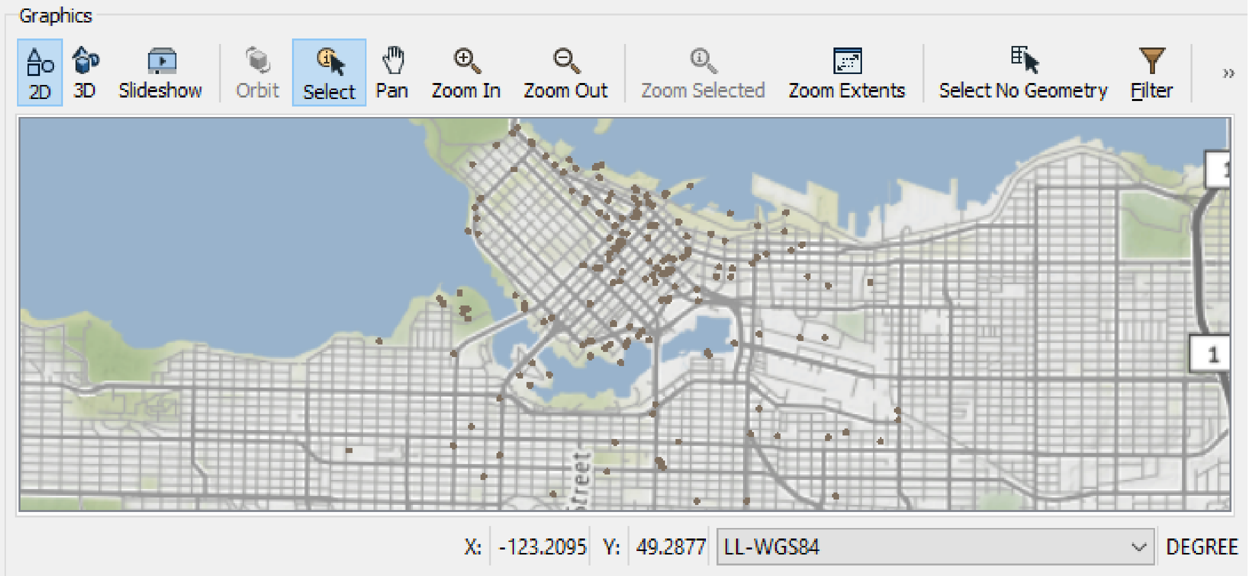
To use a background map, your dataset needs to have a known coordinate system either stored in the data itself or specified explicitly. If your data doesn't look as you expect when you add a background map, chances are you have a coordinate system problem.

To add a background map to **Visual Preview**, Sven right-clicks on the **Graphics View** pane and selects **Background Map** > **Switch to new background map...**



He sees many background map services available under the **Source** dropdown menu. Sven selects **Stamen Maps** because it is the only provider that does not require an account. For **Map**, Sven clicks the ellipsis **[...]** button to load the options and selects "terrain".Then he clicks **OK** followed by **Save**.

The background map appears in the **Graphics View**. Sven can quickly tell that it appears the majority of art installations are located in the Downtown neighborhood, the central and northern area displayed below.

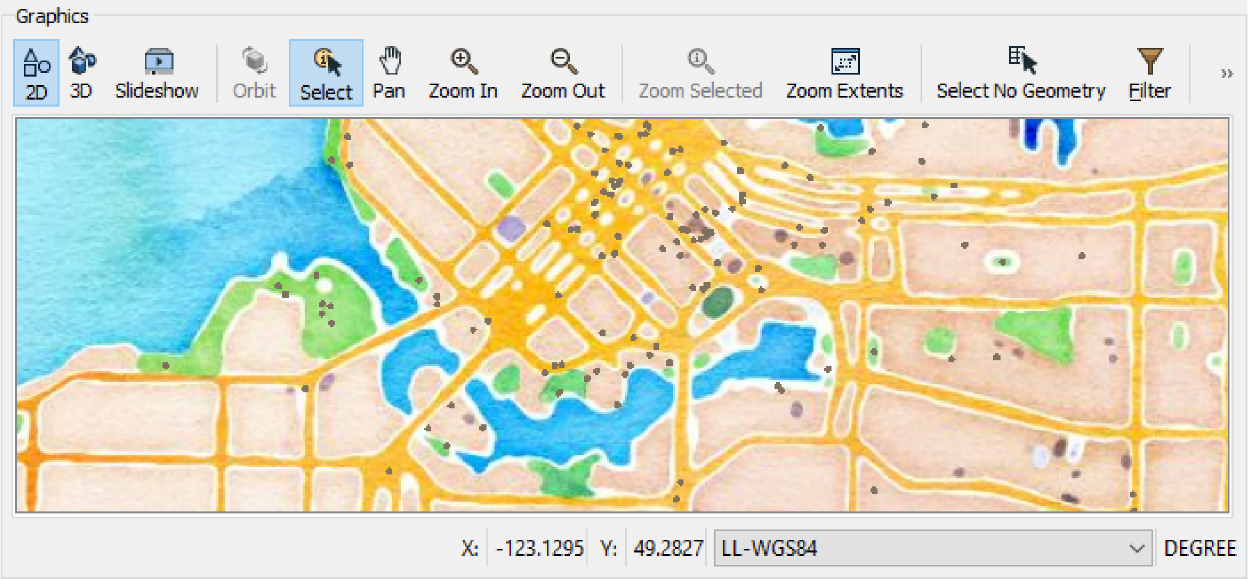


X: -123.2095 Y: 49.2877 LL-WGS84 DEGREE

Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.

Exercise

In the FME Workbench menu bar, use **Tools** > **FME Options** > **Background Maps** (or **FME Workbench** > **Preferences** > **Background Maps** on a Mac) to swap to a different Stamen Maps background map using the “watercolor” layer instead of the “terrain” layer. Your map should look like this:



X: -123.1295 Y: 49.2827 LL-WGS84 DEGREE

Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.

Next, try using the **View Source Data** button on the **Mount Pleasant** reader feature type and take note of the number of features it contains.

1In which FME Workbench window does your data with a background map appear?

○ A.Transformer Gallery

○ B.Navigator

○ C.Visual Preview

○ D.Canvas

2How many features belong to the Mount Pleasant reader feature type?

○ A.12

○ B.13

○ C.14

○ D.16

○ E.114

Check the Quiz to Earn 50 Points

Second attempt earns 25 points. Three or more earns 12 points.

Interactively View Spatial Data

Learning Objectives

After completing this unit, you'll be able to:

- View tabular data using Visual Preview.
- Interact with spatial data using Visual Preview.

Resources

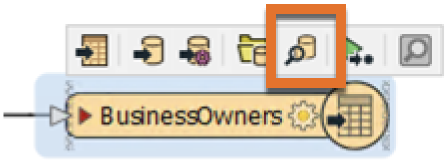
- Starting workspace

Video

Explore Visual Preview

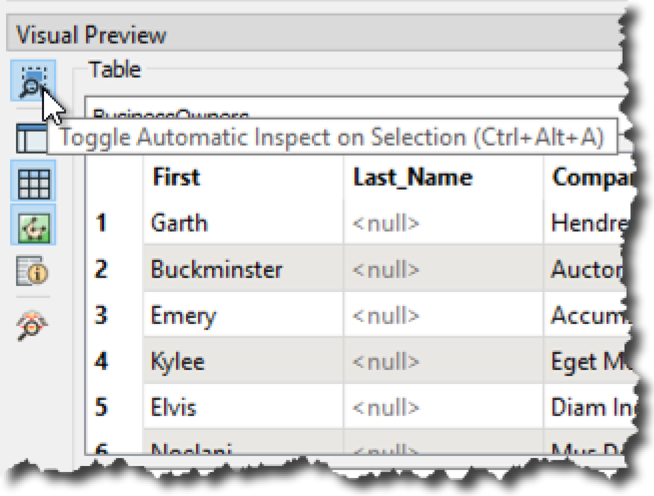
An important step in a data transformation process is data inspection. It is important to inspect the output dataset to ensure that the process was successful.

Continuing with his workspace, Sven uses FME Workbench (2022.0 or later) to open the BusinessOwners written data in **Visual Preview** by selecting it and then clicking **View Written Data**.



Now that the BusinessOwners feature class (a geodatabase feature class is the equivalent of an FME feature type) in the Vancouver.gdb geodatabase is loaded into **Visual Preview**, Sven uses the Visual Preview toolbar to control the display. The **Visual Preview** will display the geometry of features in the **Graphics View** and the attributes in the **Table View.**

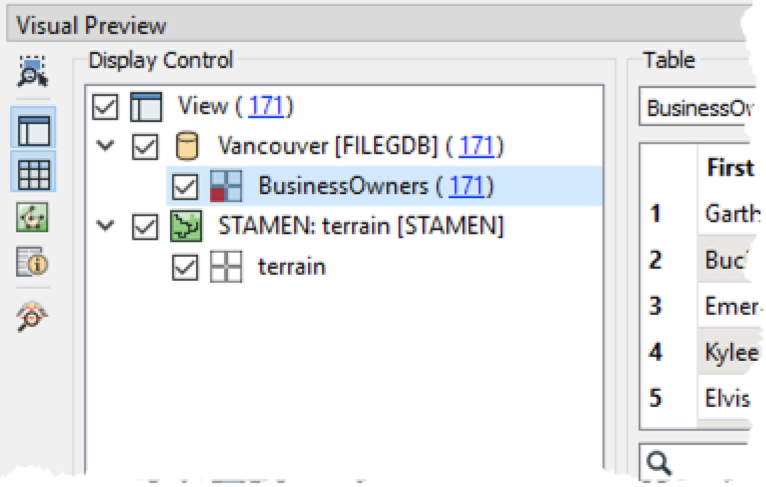
By default, **Visual Preview** will display whichever object (feature type or transformer) is selected on the canvas automatically. This means that if Sven selects another object on the canvas, **Visual Preview** will display that data. Sven doesn't want that to happen because he'll lose the view of the BusinessOwners feature type, so he clicks the **Toggle Automatic Inspect on Selection** button on the left side of **Visual Preview** to turn it off.



	First	Last_Name	Compa
1	Garth	<null>	Hendre
2	Buckminster	<null>	Auctor
3	Emery	<null>	Accum
4	Kylee	<null>	Eget M
5	Elvis	<null>	Diam In
6	Noelani	<null>	Mus D

Change How Features Are Displayed

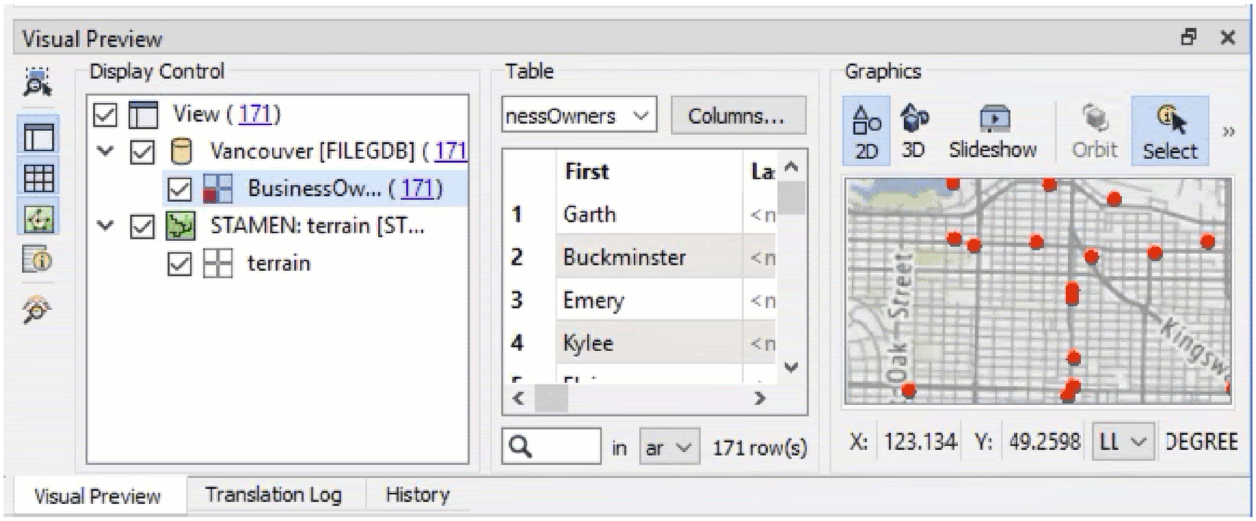
Sven wants to change the color and size of the points shown in the **Graphics View** to make the points easier to see against the background map. He clicks the **Display Control** button to open **Display Control**, which lets him see a list of the layers being inspected. Sven unchecks the box next to **STAMEN** to turn off the background map for now; this will allow the points to be seen more clearly. Clicking on the grid icon next to **BusinessOwners** opens the **Geometry Styles** dialog. Here Sven can change the display symbology and color of the points. Sven selects the color red, and increases the point size to 8, then turns the background map back on.



☑ View (171)
☑ Vancouver [FILEGDB] (171)
☑ BusinessOwners (171)
☑ STAMEN: terrain [STAMEN]
☑ terrain

	First
1	Garth
2	Buc
3	Emer
4	Kylee
5	Elvis

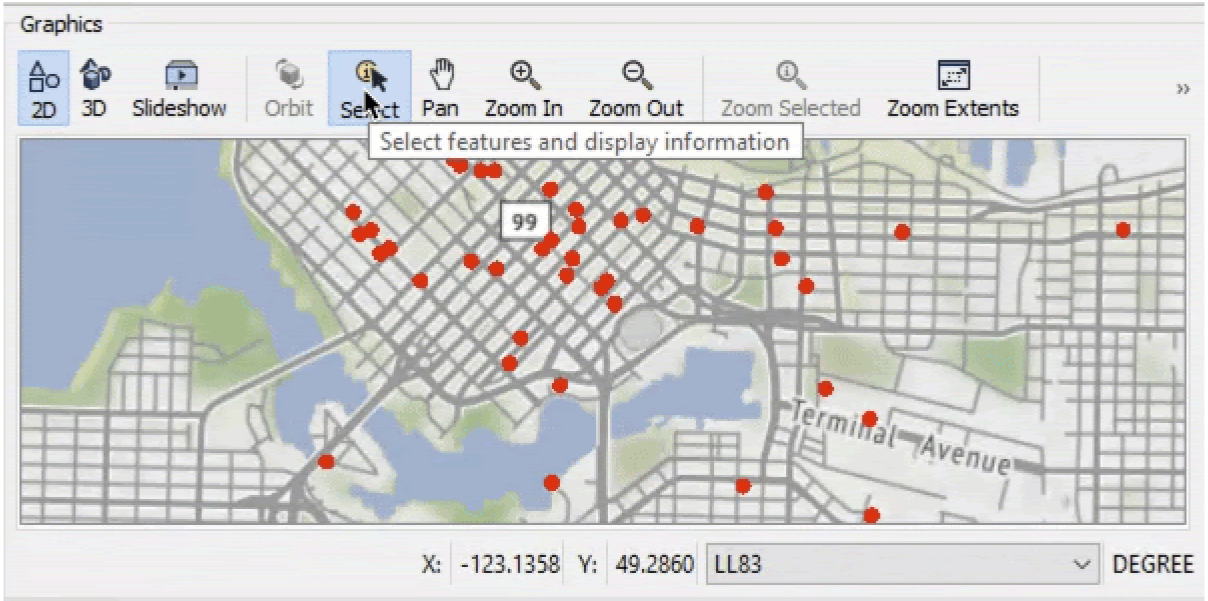
To allow more space for the **Graphics View**, Sven toggles off **Display Control** and **Table View**.



	First	La
1	Garth	<n
2	Buckminster	<n
3	Emery	<n
4	Kylee	<n

Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.

Sven clicks on **Zoom Extents** to see all of the data points.



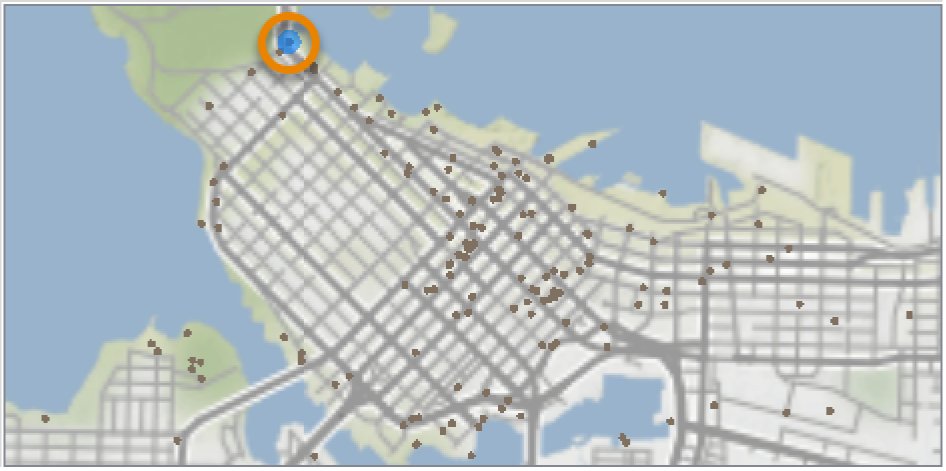
Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.

Visual Preview and the stand-alone application **Data Inspector** are not a Geographic Information System; they can not be used to create polished cartographic output, conduct interactive spatial analysis, or edit data. The purpose of **Visual Preview** and **Data Inspector** is simply to inspect data.

Exercise

Use **Pan** and **Zoom** to find the northernmost public art installation. Use **Select** to select it.

Use the **Table View** or **Feature Information** to take note of the title of the installation.



Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.

1Visual Preview and Data Inspector are examples of a GIS

○ A.True
○ B.False

2The default behaviour of the Visual Preview is to display the cache of the selected object on the canvas. This can be turned off by clicking which Visual Preview button?

○ A.Display Control
○ B.Open in Data Inspector
○ C.Show/Hide Feature Information window
○ D.Automatic Inspect on Selection

3What is the title of the northernmost artwork in the Public Art dataset?

○ A.Welcome Figures
○ B.Aerodynamic Forms in Space
○ C.Gate to the Northwest Passage
○ D.Flame of Peace

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

View Data as a Table

Learning Objectives

After completing this unit, you'll be able to:

- View your data as a table.
- Use Visual Preview to verify the results of a workspace.

Resources

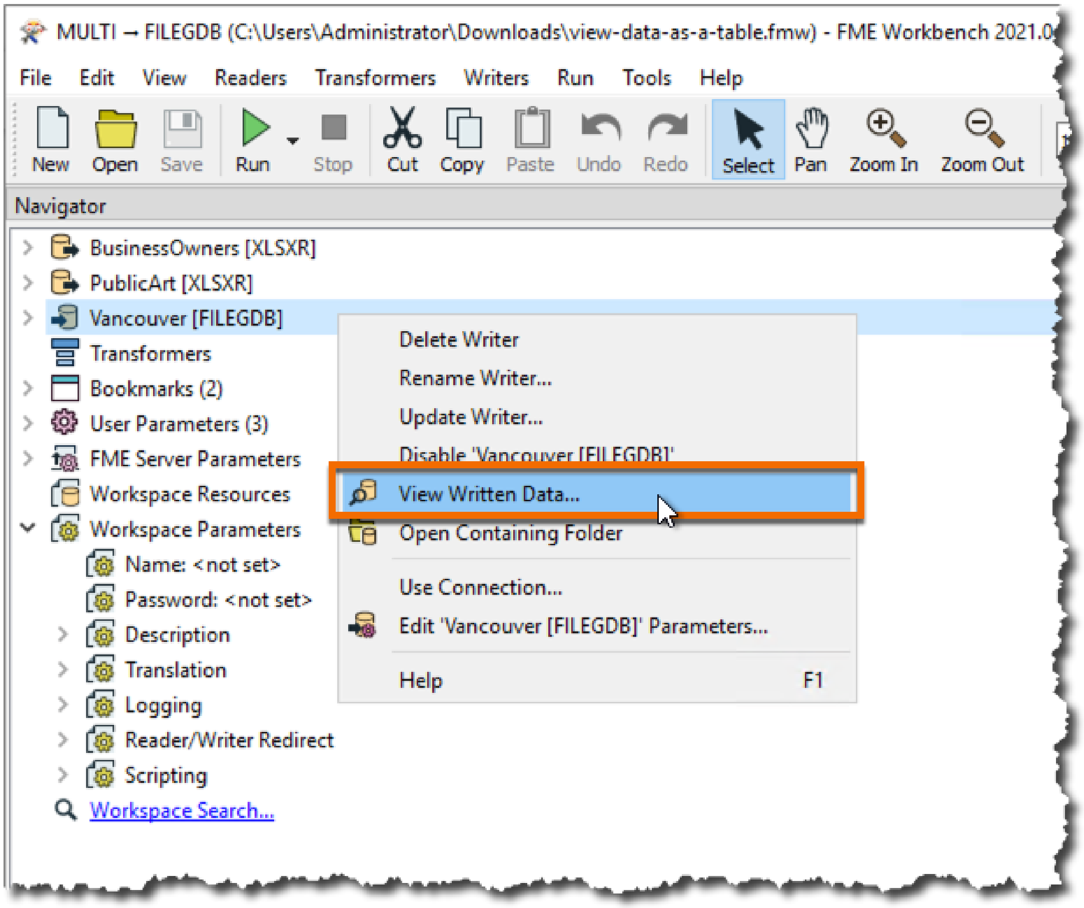
- Starting workspace

Video

Open the Written Dataset

In addition to 2D or 3D view modes, you can view attribute data for features in table format.

Sven continues to work on his Excel to geodatabase workspace using FME Workbench (2022.0 or later) and inspects the Vancouver.gdb output by right-clicking on the **Vancouver [FILEGDB]** writer in the **Navigator** and selecting **View Written Data...**. This will show BusinessOwners and PublicArt in Visual Preview. If Sven had selected **View Written Data** on a writer feature type, only that feature type would be displayed in Visual Preview.



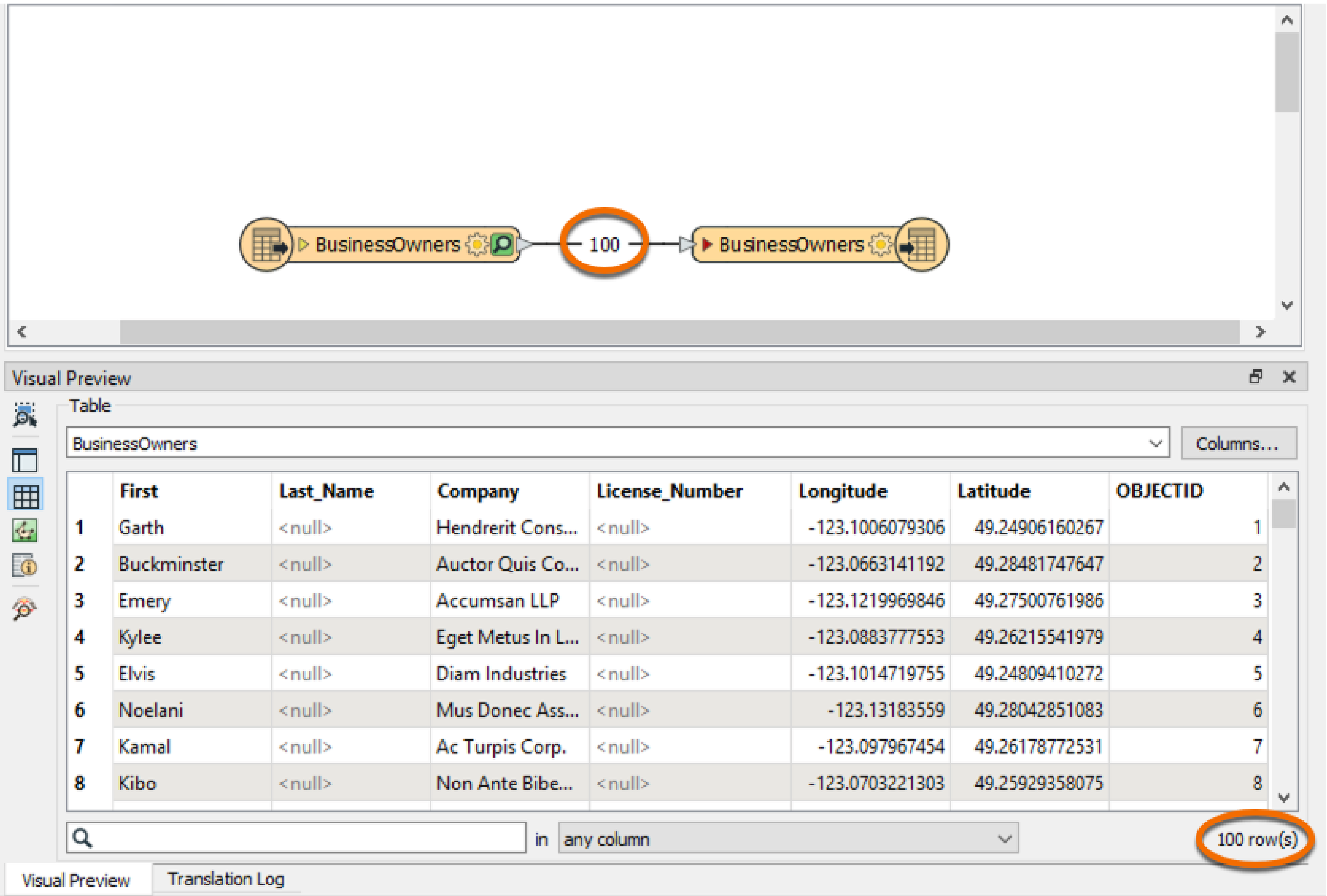
The **Table** view now displays **BusinessOwners**.

The Navigator window is a structured list of parameters that represent and control all of the components of a translation. The Navigator displays an overview of the source (reader) and destination (writer) information, bookmarks, workspace parameters, and transformers.

In general, it is a hierarchical view of the information in the graphical pane. You can adjust most dataset, feature type, and attribute parameters from here, as well as directly on the canvas.

Inspect Data Quantity

After running the translation, the bottom-right corner of the **Table** view shows the number of rows (100). This number matches the feature count on the feature connection line on the canvas. Sven uses this count to confirm that all his features have been written. He sees that 100 features were read and confirms that 100 exist in the written data.



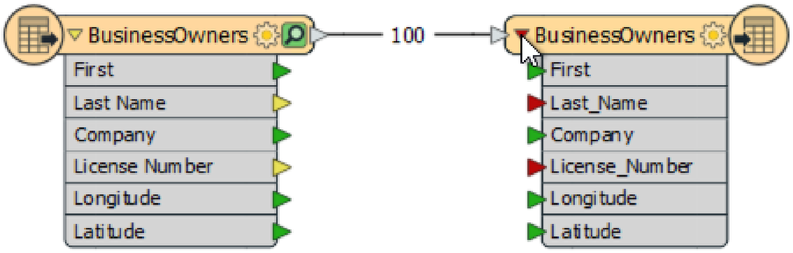
	First	Last_Name	Company	License_Number	Longitude	Latitude	OBJECTID
1	Garth	<null>	Hendrerit Cons...	<null>	-123.1006079306	49.24906160267	1
2	Buckminster	<null>	Auctor Quis Co...	<null>	-123.0663141192	49.28481747647	2
3	Emery	<null>	Accumsan LLP	<null>	-123.1219969846	49.27500761986	3
4	Kylee	<null>	Eget Metus In L...	<null>	-123.0883777553	49.26215541979	4
5	Elvis	<null>	Diam Industries	<null>	-123.1014719755	49.24809410272	5
6	Noelani	<null>	Mus Donec Ass...	<null>	-123.13183559	49.28042851083	6
7	Kamal	<null>	Ac Turpis Corp.	<null>	-123.097967454	49.26178772531	7
8	Kibo	<null>	Non Ante Bibe...	<null>	-123.0703221303	49.25929358075	8

Remember that a **feature** is a single row in a table or a single piece of geometry (point, line, polygon, etc.) and its associated attributes.

Inspect Attributes

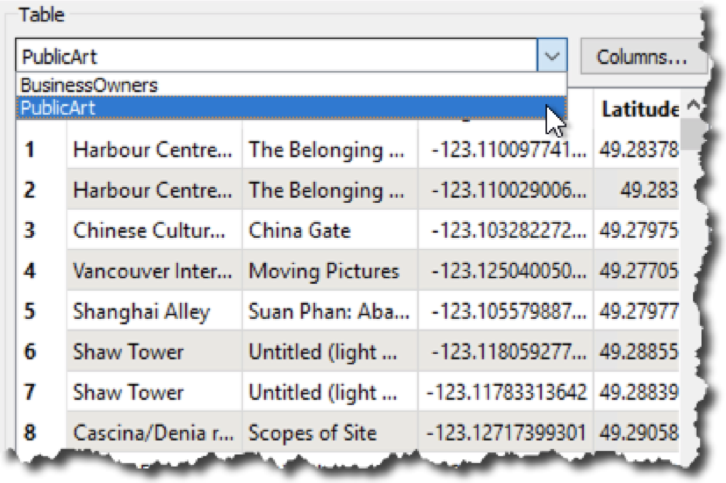
The **Table** view shows an extra attribute: OBJECTID. Additional attributes may be created by the writer if they are required by the format, as in this case. All Esri geodatabase datasets contain an OBJECTID attribute. An ObjectID is a unique, not null integer field used to uniquely identify rows in tables in a geodatabase.

Sven checks that all the desired columns (feature attributes) are present and have values. Sven notices that the entries for **Last_Name** and **License_Number** are all <null>. This is typically a bad sign; null means that there is no value for that attribute. On the canvas, Sven clicks on the triangle on the **BusinessOwners** writer feature type to see the feature type's attributes. The triangles are red beside the attributes with null values. The red triangles indicate that the attribute is not mapped to a source attribute. This occurred because the spaces in **Last Name** and **License Number** were converted to underscores. The spaces were converted to underscores because the Esri geodatabase format does not support spaces in attribute names.



View a Different Table

Vancouver.gdb also contains the PublicArt feature class. Sven clicks on the drop-down triangle beside BusinessOwners and changes the table to display data for PublicArt.



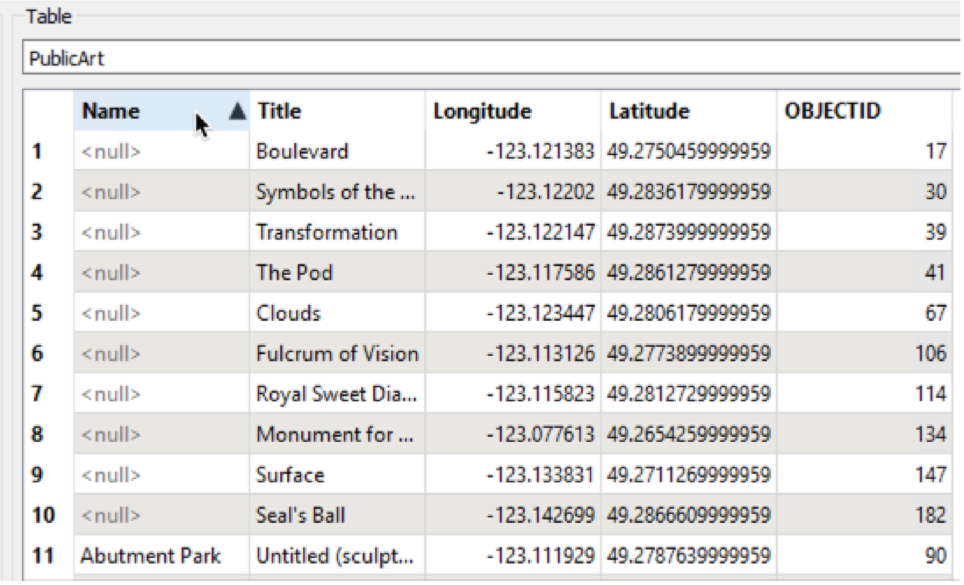
Table

PublicArt

				Latitude
1	Harbour Centre...	The Belonging ...	-123.110097741...	49.28378
2	Harbour Centre...	The Belonging ...	-123.110029006...	49.283
3	Chinese Cultur...	China Gate	-123.103282272...	49.27975
4	Vancouver Inter...	Moving Pictures	-123.125040050...	49.27705
5	Shanghai Alley	Suan Phan: Aba...	-123.105579887...	49.27977
6	Shaw Tower	Untitled (light ...	-123.118059277...	49.28855
7	Shaw Tower	Untitled (light ...	-123.11783313642	49.28839
8	Cascina/Denia r...	Scopes of Site	-123.12717399301	49.29058

Sort By Attributes

Sven notices that PublicArt is not sorted alphabetically by name. He clicks on the **Name** heading to sort the column alphabetically. This is also a good way to find any <null> attribute values. Null attributes will appear first.



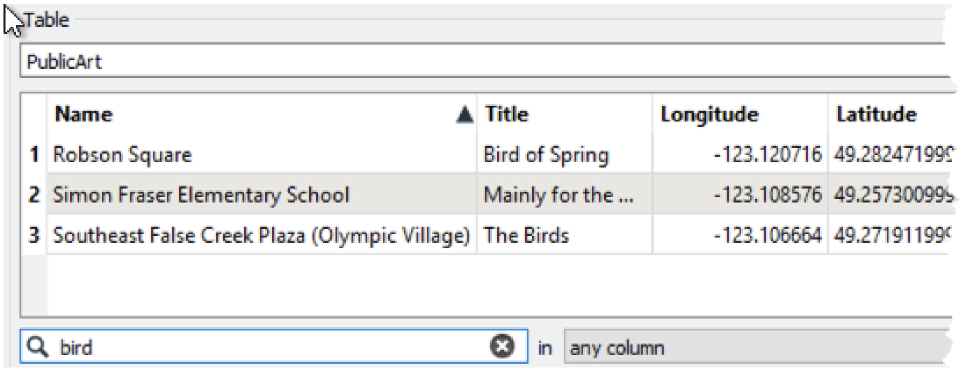
Table

PublicArt

	Name ▲	Title	Longitude	Latitude	OBJECTID
1	<null>	Boulevard	-123.121383	49.2750459999959	17
2	<null>	Symbols of the ...	-123.12202	49.2836179999959	30
3	<null>	Transformation	-123.122147	49.2873999999959	39
4	<null>	The Pod	-123.117586	49.2861279999959	41
5	<null>	Clouds	-123.123447	49.2806179999959	67
6	<null>	Fulcrum of Vision	-123.113126	49.2773899999959	106
7	<null>	Royal Sweet Dia...	-123.115823	49.2812729999959	114
8	<null>	Monument for ...	-123.077613	49.2654259999959	134
9	<null>	Surface	-123.133831	49.2711269999959	147
10	<null>	Seal's Ball	-123.142699	49.2866609999959	182
11	Abutment Park	Untitled (sculpt...	-123.111929	49.2787639999959	90

Filter by Attribute Value

Sven’s colleagues are really interested in birds and he is curious if there is any bird-related public art. He enters "bird" in the filter at the foot of the Table view. The Table view then only displays the three features that have "bird" in an attribute value.



Table

PublicArt

	Name ▲	Title	Longitude	Latitude
1	Robson Square	Bird of Spring	-123.120716	49.28247199
2	Simon Fraser Elementary School	Mainly for the ...	-123.108576	49.25730099
3	Southeast False Creek Plaza (Olympic Village)	The Birds	-123.106664	49.27191199

bird in any column

Exercise

Make sure you have followed along with Sven’s steps.

1It is possible to show or hide columns from the Table view.

○ A.True
○ B.False

2Additional attributes may be created by the writer if they are required by the format.

○ A.True
○ B.False

3Right-clicking on column headings in Table view allows you to sort them:

○ A.Natural
○ B.Alphabetical
○ C.Numeric
○ D.All of the above

Check the Quiz to Earn 50 Points

Second attempt earns 25 points. Three or more earns 12 points.

View Information About a Specific Feature

Learning Objectives

After completing this unit, you'll be able to:

- Query features for additional information.
- View coordinate system information about features.

Resources

- Starting workspace

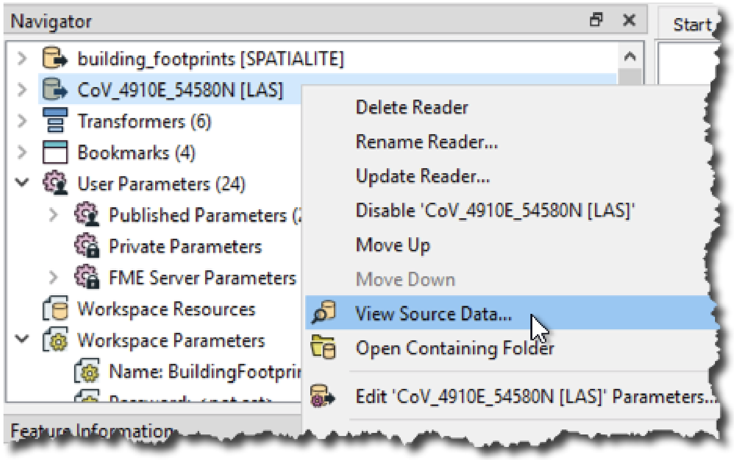
Video

Inspect Source Data

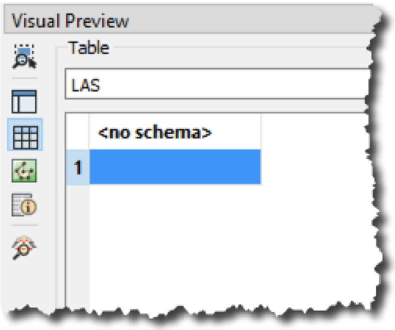
Sven's colleague Amar is starting an FME project converting 2D building footprints to 3D models to compare them with a point cloud. He opens his workspace using FME Workbench (2022.0 or later). This workspace takes 2D building footprints and extrudes them to make 3D buildings. He also wants to compare the resulting extruded 3D buildings to a point cloud to see if the extrusion is roughly correct compared to what is observed in the point cloud.

Amar has not yet inspected the point cloud dataset, and is curious about what units of distance are used in the point cloud data (feet or meters) and what software was used to generate the point cloud—TopoDOT had been used previously but Terrascan was used for the most recent point clouds.

The point cloud format is ASPRS Lidar Data Exchange Format (LAS). The reader is listed in the Navigator window as [LAS]. Amar right-clicks on the LAS reader and selects **View Source Data...** to inspect the original dataset.



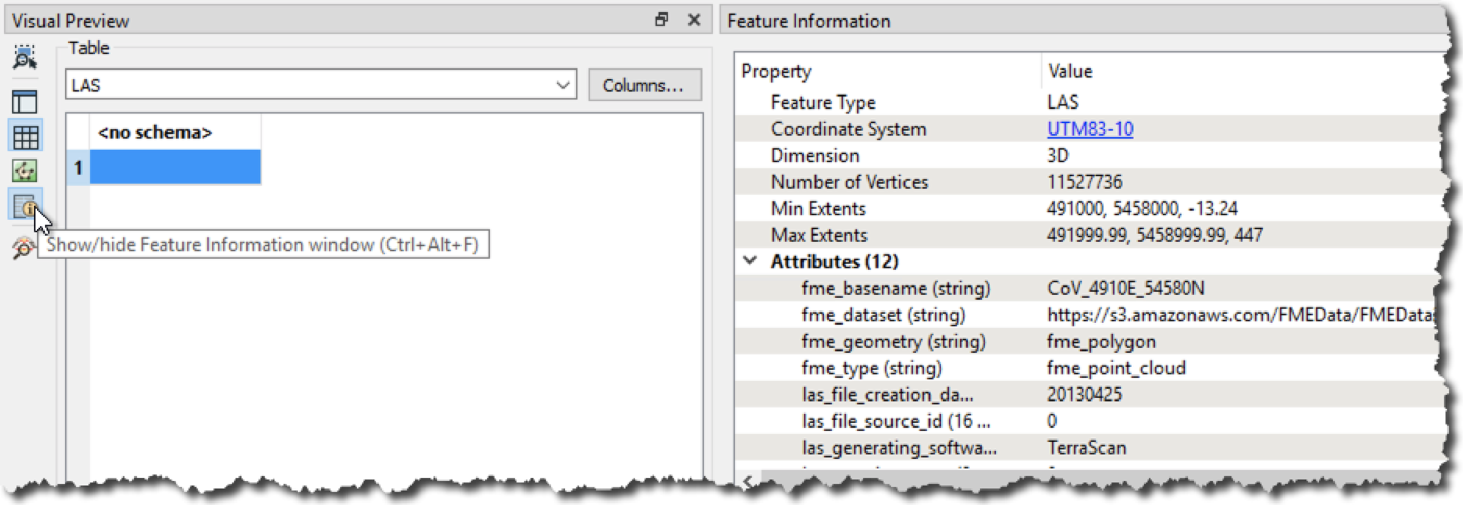
The Table view in Visual Preview shows no user-defined attributes. This indicates that there are no user-defined attributes in the dataset, but that does not mean there are no attributes at all. The features will have format attributes but format attributes are not displayed in the Table view by default.



A quick check of the ASPRS LiDAR Data Exchange Format (LAS) "Quick Facts" in the documentation confirms that this format does not support user-defined attributes. But, Amar knows that there are more attributes than just user-defined attributes. Amar selects the single feature in the table view and opens the Feature Information window.

Feature Information Window

The Feature Information window displays everything that FME knows about the feature selected in the **Table** or **Graphics View**. Here you can find all the user-defined attributes, format attributes, coordinate system information, and geometry information.

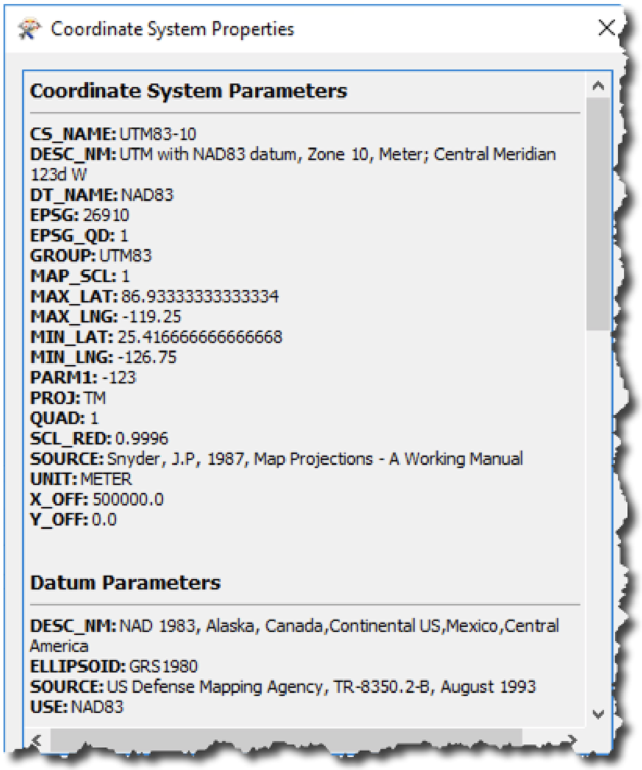


Property	Value
Feature Type	LAS
Coordinate System	UTM83-10
Dimension	3D
Number of Vertices	11527736
Min Extents	491000, 5458000, -13.24
Max Extents	491999.99, 5458999.99, 447
Attributes (12)	
fme_basename (string)	CoV_4910E_54580N
fme_dataset (string)	https://s3.amazonaws.com/FMEData/FMEData
fme_geometry (string)	fme_polygon
fme_type (string)	fme_point_cloud
las_file_creation_da...	20130425
las_file_source_id (16 ...	0
las_generating_softwa...	TerraScan

Amar checks the upper part of **Feature Information**, above **Attributes**, where he finds:

- The feature type (layer/table) the selected feature belongs to
- If the feature is spatial:
  - Its coordinate system
  - Summary information about its geometry (dimension, number of vertices, and extents)

The coordinate system catches Amar's eye because it’s a link. Clicking on UTM83-10 opens the Properties dialog for the coordinate system. Here, Amar can see that the units for UTM83-10 are meters. This is important for Amar to know especially if he plans to do any calculations such as measuring a distance or an area. By default, FME will use the coordinate system’s unit as measurement, and the result would sure look odd if those units were in degrees!



Coordinate System Parameters

CS_NAME: UTM83-10
DESC_NM: UTM with NAD83 datum, Zone 10, Meter; Central Meridian 123d W
DT_NAME: NAD83
EPSG: 26910
EPSG_QD: 1
GROUP: UTM83
MAP_SCL: 1
MAX_LAT: 86.93333333333334
MAX_LNG: -119.25
MIN_LAT: 25.416666666666668
MIN_LNG: -126.75
PARM1: -123
PROJ: TM
QUAD: 1
SCL_RED: 0.9996
SOURCE: Snyder, J.P, 1987, Map Projections - A Working Manual
UNIT: METER
X_OFF: 500000.0
Y_OFF: 0.0

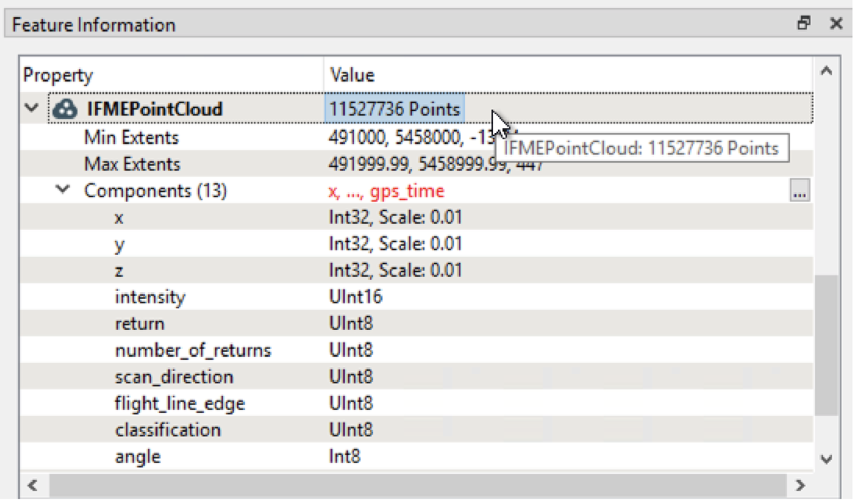
Datum Parameters

DESC_NM: NAD 1983, Alaska, Canada,Continental US,Mexico,Central America
ELLIPSOID: GRS1980
SOURCE: US Defense Mapping Agency, TR-8350.2-B, August 1993
USE: NAD83

Amar closes the **Coordinate System Properties** dialog and then checks the items under the **Attributes** header in the **Feature Information** window. This section reports the attributes associated with the feature, including user attributes and format attributes (for example fme_type).

Here, Amar can see that this dataset was generated by TerraScan, not TeraDOT, so this is the more recent point cloud dataset. Amar can also see the creation date of this LAS file (las_file_creation_date (string): 20130425), the LAS version (las_version (string): 1.2), and the location of the dataset (fme_dataset (string): https://s3.amazonaws.com/FMEData/FMEData2022/Data/PointClouds/CoV_4910E_54580N.laz).

Amar then checks the **Geometry** section in the lower part of the **Feature Information** window. This section reports the geometry of the feature. It includes the geometry type and a list of the coordinates that make up the feature. Here Amar can see the extents of the data, and the number of points in the point cloud.



Property	Value
IFMEPointCloud	11527736 Points
Min Extents	491000, 5458000, -13
Max Extents	491999.99, 5458999.99, 447
Components (13)	x, ..., gps_time
x	Int32, Scale: 0.01
y	Int32, Scale: 0.01
z	Int32, Scale: 0.01
intensity	UInt16
return	UInt8
number_of_returns	UInt8
scan_direction	UInt8
flight_line_edge	UInt8
classification	UInt8
angle	Int8

While the **Table** view typically only shows exposed user-defined attributes, the **Feature Information** window also displays FME-generated format attributes and list attributes. Any user-defined attributes or fme_type attributes visible in **Feature Information** can also be exposed for display in the table, allowing Amar to use them in his workspace.

Using **Feature Information** is one of the fastest ways to find out what units you are working with when calculating areas or distances.

Exercise

Make sure you have followed along with Amar’s steps.

1What is the unit of the coordinate system used by the LAS file?

○ A.Meters
○ B.Feet
○ C.Decimal Degrees

2How many points are in the point cloud?

○ A.1,054,031
○ B.491,384
○ C.5,458,240
○ D.11,527,736

3Feature Information shows:

○ A.User defined attributes
○ B.Format attributes
○ C.List attributes
○ D.All of the above

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Interactively View 3D Data

Learning Objectives

After completing this unit, you'll be able to:

- Swap between 2D and 3D views of your data.
- View and interact with 3D data.

Resources

- Starting workspace template

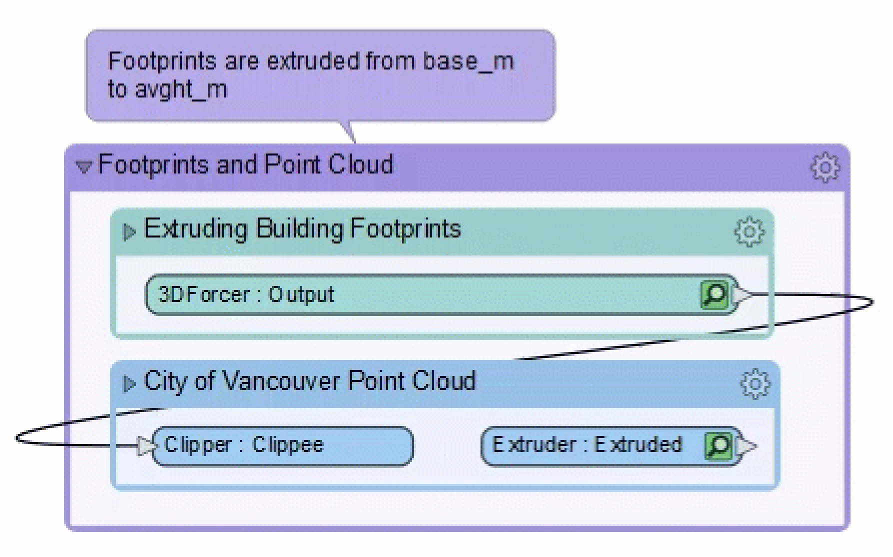
Video

Inspecting 3D Data

Amar has made some progress with his 2D to 3D building workspace but feels that there is something wrong with the extrusions. Amar would like Sven to review his progress so far, so Amar has sent an FME Workbench template to Sven to examine.

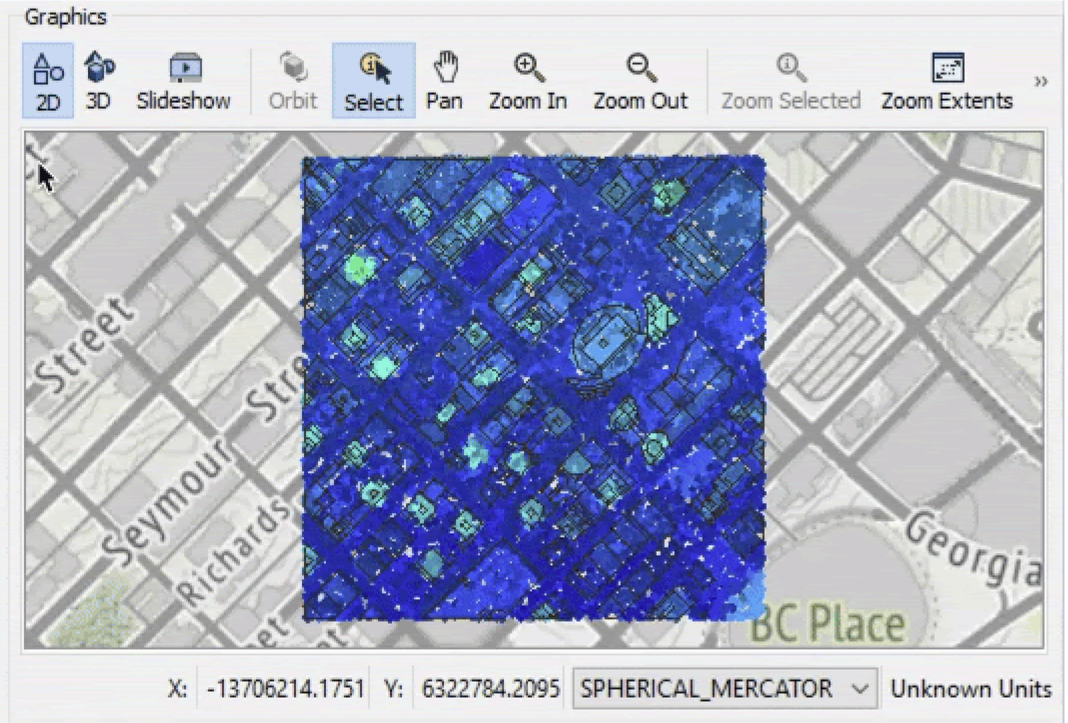
Sven opens Amar's workspace template in FME Workbench (2022.0 or later) and runs the workspace to create the feature caches. The workspace displays some examples of best practices in authoring a workspace; collapsed bookmarks simplify the view and annotations explain what Sven is looking at.

Sven wants to inspect extruded building footprints and the point cloud at the same time to see if the building footprints have been extruded to an appropriate height by comparing the extrusions to the point cloud. Holding down the CTRL key (CMD on Mac) while clicking on caches allows Sven to select multiple caches to view in Visual Preview. Sven holds down CTRL and selects the **Extruded Building Footprints** cache and the **City of Vancouver Point Cloud** cache.


Footprints are extruded from base_m to avght_m
Footprints and Point Cloud
Extruding Building Footprints
3DForcer : Output
City of Vancouver Point Cloud
Clipper : Clippee
Extruder : Extruded

The 2D view doesn't show much of interest, but a rotated 3D view will. Sven clicks the 3D button in the Graphics window.

The background map disappears because background maps are not supported in the 3D view. Sven zooms in to an area of interest and then rotates the view using the **Orbit** tool.



Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under CC BY SA.

Sven can see that many buildings are not extruded accurately. He guesses that this is a result of the buildings being extruded by an average height instead of a maximum height.

If you have a mix of 2D and 3D and want to switch back and forth, you can use the **2D** and **3D** buttons in the **Graphics View** toolbar.

Exercise

Make sure you have followed along with Sven’s steps.

1Background maps, if selected, will display in both 2D and 3D views

○ A.True
○ B.False

2A template can contain:

○ A.A workspace
○ B.Source datasets
○ C.Caches
○ D.All of the above

3The 3D view in the Graphics View can be rotated using the:

○ A.Zoom Extents tool
○ B.Zoom In tool
○ C.Pan tool
○ D.Orbit tool

Check the Quiz to Earn 50 Points

Second attempt earns 25 points. Three or more earns 12 points.

Quickly Create a Conversion Workflow

Learning Objectives

After completing this unit, you'll be able to:

- Identify when to use Generate Workspace versus starting with a blank workspace.
- Connect to a database.
- Quickly create a workflow using Generate Workspace.

Learning content in the FME Academy presents the story of a user addressing their data integration challenges with FME. **You should follow along with their actions using your own installation of FME (2022.0 or later)** or request an on-demand virtual machine in the footer link below. Some units will require you to follow their steps or to take additional steps on your own to answer a quiz question.

The **Resources** section will provide you with links to interactive tutorials and starting workspaces when necessary.

Resources

- Interactive tutorial to help you follow along
- Complete workspace

Video

Videos are provided in this module if you prefer to watch instead of reading the text below. Note that some Quiz answers might require you to read the text.

Jumpstart Your Data Integration Workflows

Jennifer is a GIS Specialist working for a local government. She needs to read in a table of business license data from a PostGIS database and write it to an Esri geodatabase, a widely-used spatial database. The PostGIS database is used internally, while the geodatabase is used in public-facing applications. Therefore, she wants to edit the attribute names and filter out some features with revoked business licenses in the final geodatabase in order to have appropriate data for displaying to the public.

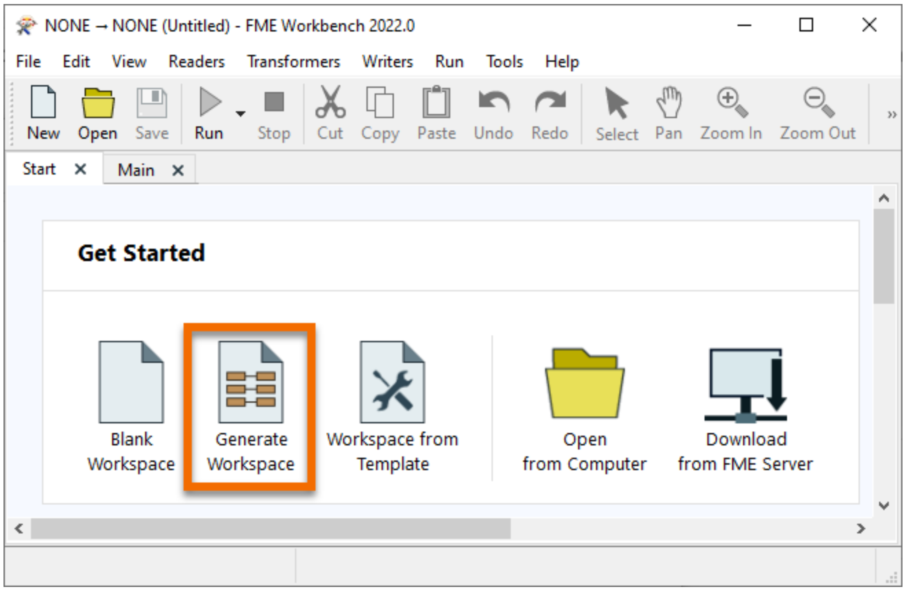
Before she conducts those steps, however, she needs to get started with a basic conversion workspace. She could add her readers and writers manually on a blank canvas, or she could use the **Generate Workspace** dialog to set up her workspace faster.

Jennifer is working with the same BusinessOwners dataset as Sven from the previous module, but she's reading it from a database instead of an Excel spreadsheet.

Jennifer knows that the **Generate Workspace** dialog will let her quickly create a workspace. She just has to supply the reader and writer format and dataset locations, and FME does the rest. It will automatically do its best to duplicate the reader schema (formal definition of a dataset's structure) on the writer, handling differences in data types or restrictions on geometry types or attribute name lengths.

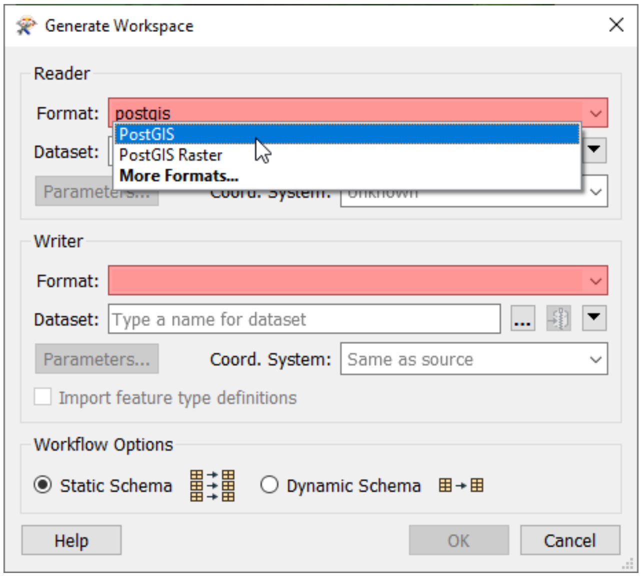
To generate a workspace, Jennifer starts FME Workbench (2022.0 or later) and clicks **Generate** in the Start tab.

Follow along with Jennifer's steps using your own version of FME.


NONE → NONE (Untitled) - FME Workbench 2022.0
File Edit View Readers Transformers Writers Run Tools Help
New Open Save Run Stop Cut Copy Paste Undo Redo Select Pan Zoom In Zoom Out
Start
Main
Get Started
Blank Workspace
Generate Workspace
Workspace from Template
Open from Computer
Download from FME Server

Connect to a Database

With the **Generate Workspace** dialog open, Jennifer fills out the **Reader** section first to define her source data. She chooses PostGIS by typing in the **Format** field and then selecting "PostGIS" from the drop-down options.


Generate Workspace

Reader

Format: postgis

PostGIS
PostGIS Raster
More Formats...

Dataset:

Parameters... Coord. System: Unknown

Writer

Format:

Dataset: Type a name for dataset

Parameters... Coord. System: Same as source

□ Import feature type definitions

Workflow Options

● Static Schema ○ Dynamic Schema

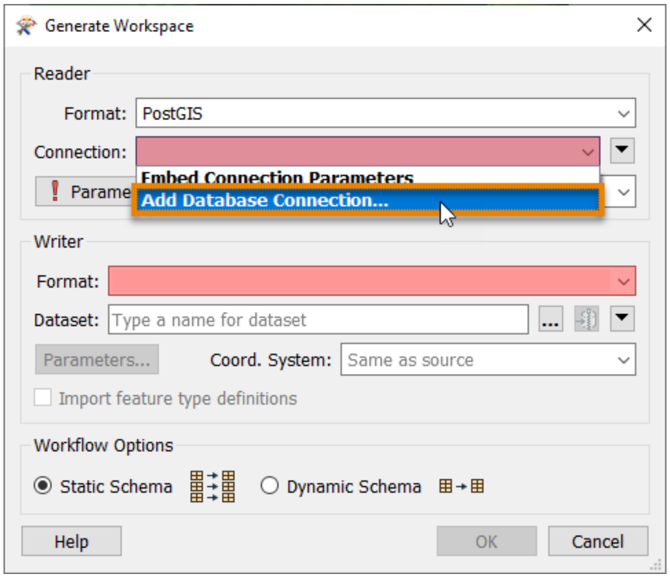
Help OK Cancel

Next, she needs to connect to the database. She can do that with a database connection.

Database connections save authentication information for databases. FME also has **web connections** to connect to web services and APIs. They are stored on the user's operating system profile, so authentication information can be stored separately from the workspace. They can also be published to FME Server to allow multiple users to share them without exposing any passwords.

Learn More

Jennifer clicks the **Connection** drop-down menu and chooses **Add Database Connection**.


Generate Workspace

Reader

Format: PostGIS

Connection:

Embed Connection Parameters

Add Database Connection...

Parameters...

Writer

Format:

Dataset: Type a name for dataset

Parameters... Coord. System: Same as source

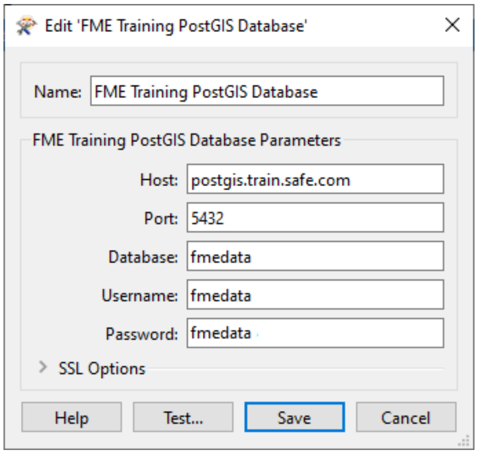
□ Import feature type definitions

Workflow Options

● Static Schema ○ Dynamic Schema

Help OK Cancel

From here, she fills in her database connection details.


Edit 'FME Training PostGIS Database'

Name: FME Training PostGIS Database

FME Training PostGIS Database Parameters

Host: postgis.train.safe.com

Port: 5432

Database: fmedata

Username: fmedata

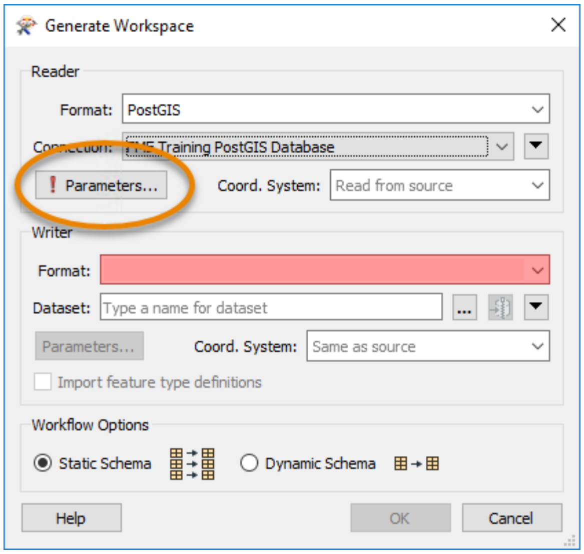
Password: fmedata

› SSL Options

Help Test... Save Cancel

She clicks **Test** and then **Save**. FME tests the connection and confirms that it is working.

Because Jennifer doesn't want to read the entire database, she clicks the **Parameters** button, which will allow her to choose the table she wants to read.


Generate Workspace

Reader

Format: PostGIS

Connection: FME Training PostGIS Database

Parameters... Coord. System: Read from source

Writer

Format:

Dataset: Type a name for dataset

Parameters... Coord. System: Same as source

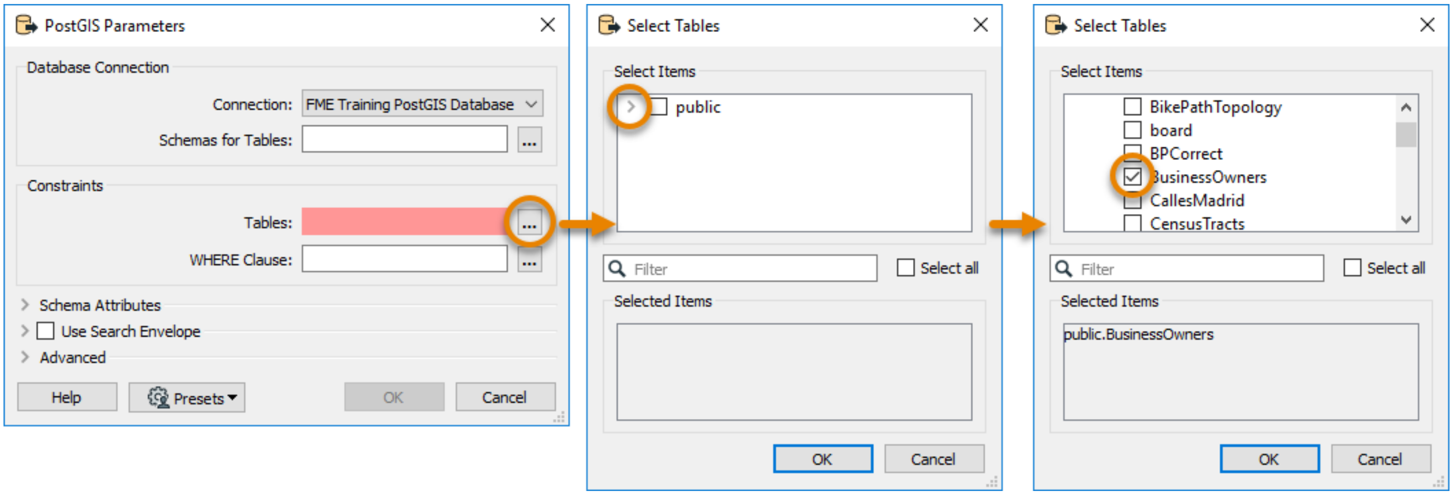
□ Import feature type definitions

Workflow Options

● Static Schema ○ Dynamic Schema

Help OK Cancel

In the **PostGIS Parameters** dialog, she clicks the ellipsis **[...]** button next to the **Tables** parameter. She clicks the **> arrow** next to **public** to expand the list of tables and then picks the **BusinessOwners** table from the list. This step ensures only the BusinessOwners table will be read.



PostGIS Parameters

Database Connection

Connection: FME Training PostGIS Database

Schemas for Tables:

Constraints

Tables:

WHERE Clause:

Schema Attributes

□ Use Search Envelope

Advanced

Select Tables

Select Items

□ public

Filter □ Select all

Selected Items

Select Tables

Select Items

□ BikePathTopology
□ board
□ BPCorrect
☑ BusinessOwners
□ CallesMadrid
□ CensusTracts

Filter □ Select all

Selected Items

public.BusinessOwners

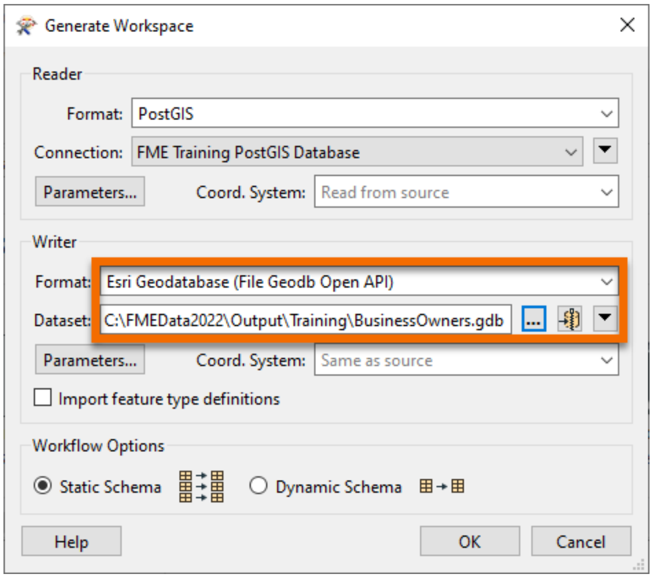
Once the **BusinessOwners** table is selected, she saves her changes by clicking **OK**.

If Jennifer wanted a subset of features, she could use SQL in the **Where Clause** parameter or a transformer like the SQLCreator to let the database do the work.

If the **BusinessOwners** table is not in the list, you should follow these instructions to reset the public database.

Generate a Workspace

Jennifer needs these business tables in an Esri geodatabase so she can use them in ArcGIS. To create a new Esri geodatabase for this data, she fills out the **Writer** section of the **Generate Workspace** dialog. She selects the **Format** "Esri Geodatabase (File Geodb Open API)" and enters the file path "C:\FMEData2022\Output\Training\BusinessOwners.gdb" for **Dataset**. BusinessOwners.gdb will be created in the Training folder once this workspace is run.



Generate Workspace

Reader

Format: PostGIS

Connection: FME Training PostGIS Database

Parameters... Coord. System: Read from source

Writer

Format: Esri Geodatabase (File Geodb Open API)

Dataset: C:\FMEData2022\Output\Training\BusinessOwners.gdb

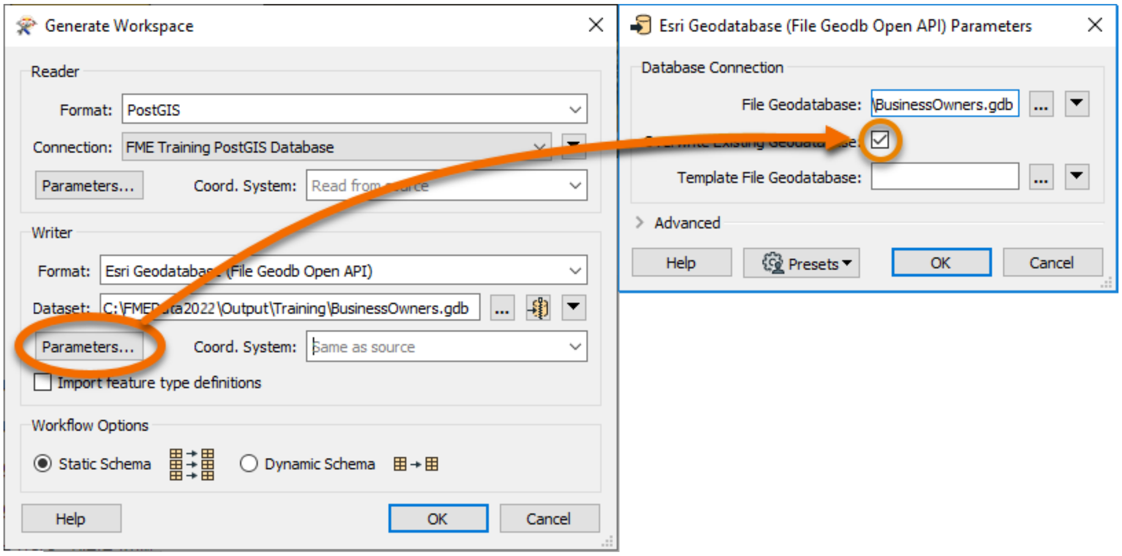
Parameters... Coord. System: Same as source

□ Import feature type definitions

Workflow Options

● Static Schema ○ Dynamic Schema

Before clicking **OK**, there is one extra step. Jennifer needs to review the parameters of the readers and writers she's using to ensure that her data is being transformed properly. Jennifer knows that while authoring her workspace, she might want to run this workspace more than once for testing purposes. And if she does that without handling the parameters correctly, especially the database operations, she could end up appending the same data to the original output from the first run of the translation. To avoid this mistake, she opens the **Parameters** and enables **Overwrite Existing Database**.


Generate Workspace

Reader

Format: PostGIS

Connection: FME Training PostGIS Database

Coord. System: Read from source

Writer

Format: Esri Geodatabase (File Geodb Open API)

Dataset: C:\FMEData2022\Output\Training\BusinessOwners.gdb

Coord. System: Same as source

☐ Import feature type definitions

Workflow Options

● Static Schema ○ Dynamic Schema

Esri Geodatabase (File Geodb Open API) Parameters

Database Connection

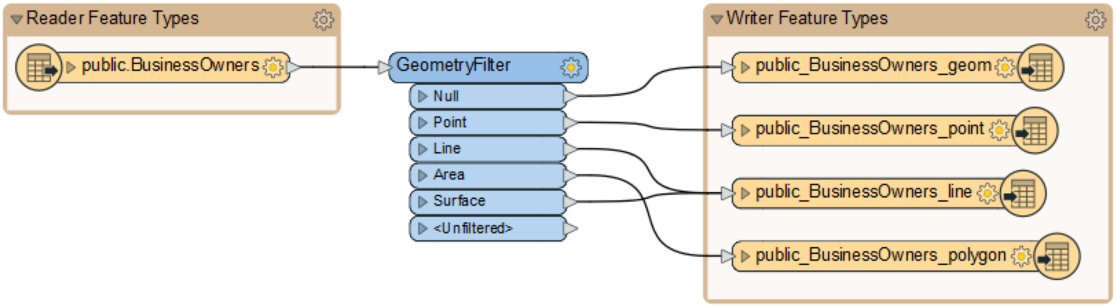
File Geodatabase: \BusinessOwners.gdb

Overwrite Existing Geodatabase: ☑

Template File Geodatabase:

This step ensures that if the workspace runs more than once the entire geodatabase is rewritten.

She saves her changes by clicking **OK**, and FME Workbench generates the workspace.



It is generally a good idea to look at the parameters before adding a reader or writer. If you ever are working with your data and find it looks odd, there is a good chance a reader or writer parameter needs to be changed.

Learn More

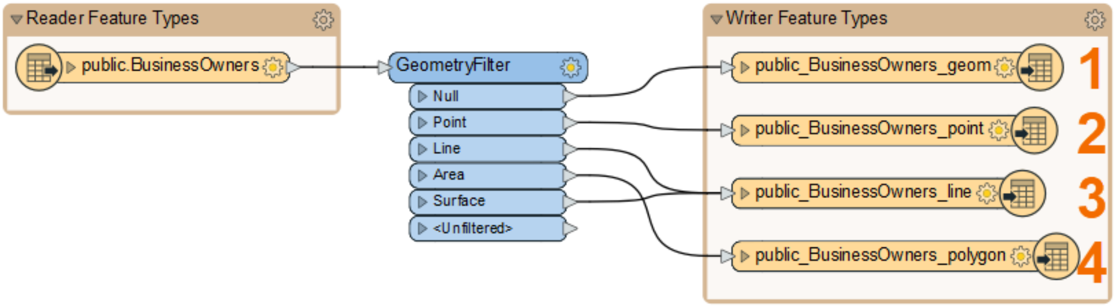
Understanding a Generated Workspace

Jennifer notices that FME has automatically added a transformer called the **GeometryFilter** to the canvas. The **GeometryFilter** is one of the hundreds of transformers available in FME that allow users to modify their data without having to code. FME automatically added a **GeometryFilter** to filter incoming PostGIS features by geometry type, writing points, lines, polygons, and collections to corresponding writer feature types. This transformer is needed because a geodatabase feature class can only have one type of geometry, and this limitation is inherent to the data format structure.

Transformers are objects on the canvas that modify features. We will discuss transformers more in the next unit.

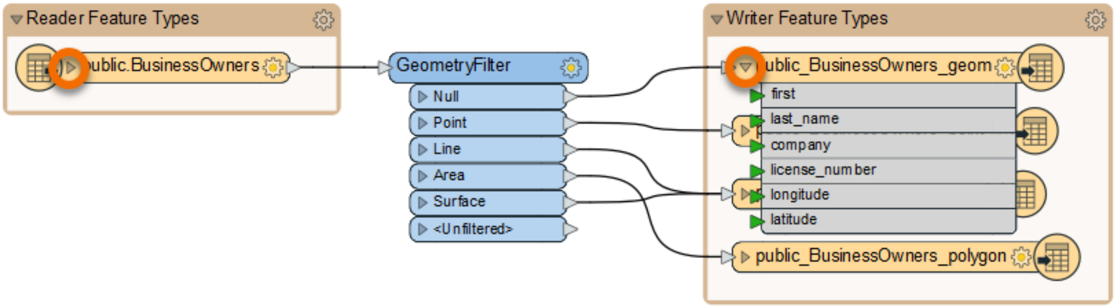
Remember the problem Sven ran into in the earlier unit, Understand and Fix Problems With Your Workspace? He added a geodatabase writer manually, but he ran into problems when he tried to write his data. The source of the error was that he had not manually set the geometry type for each geodatabase writer feature type. Using **Generate Workspace** helps you avoid errors from format limitations like this one by automatically duplicating the source schema on the destination and adding transformers where necessary.

Jennifer sees that Generate Workspace has created four feature types (feature classes in the geodatabase), one for each kind of geometry supported by the geodatabase format.



Remember, feature types correspond to tables for spreadsheets or databases or layers (geometry plus attributes) for spatial data. Each feature type belongs to a reader or writer.

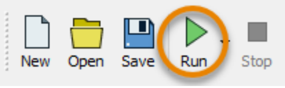
By clicking the triangle on the left of her reader and one writer feature type, Jennifer can see that FME duplicates the reader schema (e.g., attribute names) as best it can on the writer, taking into account format issues like different data types and restrictions on attribute names. These automated steps are one reason she used **Generate Workspace**.



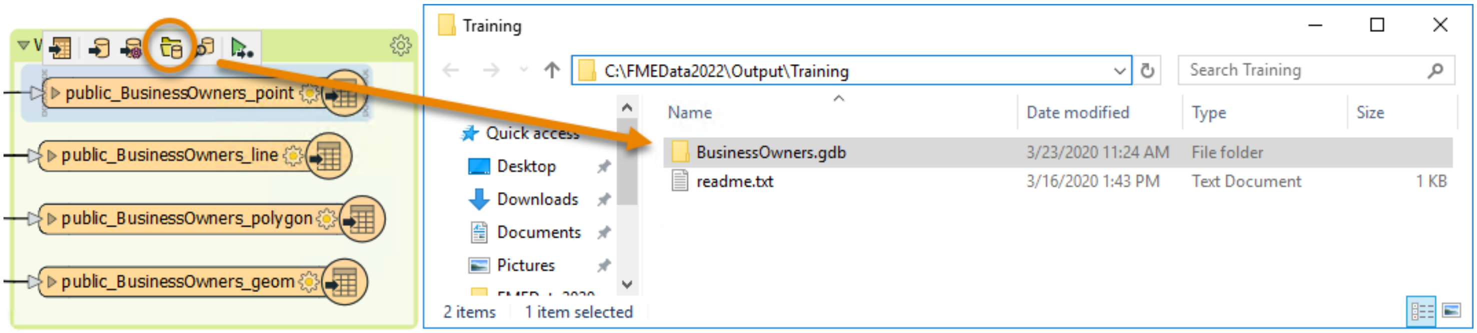
After she quickly examines the attribute names, she clicks the triangles again to hide them. She'll examine the schema in more detail later.

Run the Workspace

Jennifer clicks **Run** to run her workspace and convert her data.



After the workspace has run, the **Translation Log** reports that the "Translation was Successful". Jennifer selects one of the writer feature types and clicks **Open Containing Folder** to confirm the geodatabase has been created.


public_BusinessOwners_point
public_BusinessOwners_line
public_BusinessOwners_polygon
public_BusinessOwners_geom
Training
C:\FMEData2022\Output\Training
Search Training
Quick access
Desktop
Downloads
Documents
Pictures
2 items
1 item selected

Name	Date modified	Type	Size
BusinessOwners.gdb	3/23/2020 11:24 AM	File folder	
readme.txt	3/16/2020 1:43 PM	Text Document	1 KB

Jennifer is off to a good start. She has created a new geodatabase from her PostGIS dataset with the business owner data loaded into it. Next, she has to edit the schema.

Exercise

After following along with Jennifer's steps, find the size of your BusinessOwners.gdb output to answer the quiz question below. You can check the size on Windows by right-clicking the folder in the **File Explorer**, clicking **Properties**, and checking the **General** tab and the number under **Size**. On Mac, click the folder in **Finder** to select it, then use **File** > **Get Info** and check the **Size**.

1Generate Workspace automatically duplicates the reader schema on the writer schema, taking format limitations into account.

○ A.True
○ B.False

2When using the Generate Workspace dialog, which of the following is not mandatory?

○ A.Reader Format
○ B.Reader Dataset
○ C.Writer Format
○ D.Writer Parameters

3Which of the following is not an advantage of using a Database Connection?

○ A.The connection file is associated with the user, not the workspace.
○ B.The reader will perform better when using a database connection.
○ C.You can share a workspace without having to share database permissions.
○ D.You can upload a workspace to FME Server and optionally include the database permissions.

4The GeometryFilter created by Generate Workspace is a kind of:

○ A.Feature Class
○ B.Writer
○ C.Transformer
○ D.Feature Type

5How big is the resulting geodatabase (rounded to the nearest kilobyte)?

○ A.58
○ B.155
○ C.553
○ D.917

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Edit Data’s Schema

Learning Objectives

After completing this unit, you’ll be able to:

- View your data’s schema.
- Edit your data’s schema, including feature type and attribute names.

Resources

- Interactive tutorial to help you follow along
- Starting workspace

Video

View Source Schema

Jennifer is continuing to work on her workspace in FME Workbench (2022.0 or later). This workspace creates an Esri geodatabase from a PostGIS layer. Jennifer would like her geodatabase to have a different schema than her source PostGIS data. She’d like to make the following changes:

- Change some attributes so they make more sense to the public:
  - “first” renamed to “first_name” to match “last_name”
  - Move “last_name” to be the first column and sort it alphabetically, matching common conventions when working with names
  - Remove “latitude” and “longitude” attributes

- Rename feature types to better reflect what they contain:
  - “public_BusinessOwners_point” to “BusinessOwners”
  - “public_BusinessOwners_geom” to “BusinessOwners_incomplete”. This feature type should contain any features missing geometry so the Quality Assurance team can fix them by adding geometry.

- Remove empty feature types:
  - “public_BusinessOwners_line”
  - “public_BusinessOwners_polygon”

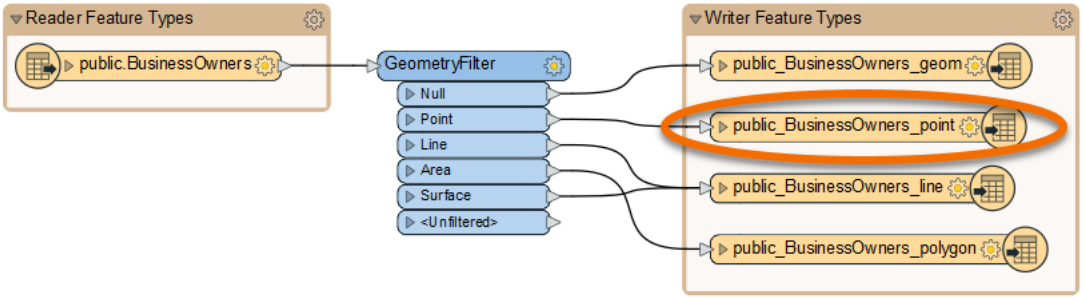
Manipulating schema (the formal definition of a dataset’s structure) is a key process in FME. When creating an FME workspace that modifies schema, there are generally two steps:

1. Edit the schema: define what schema you want on the written data. In FME, this is done by changing the feature type parameters to reflect what you want, e.g. changing attribute or feature type names.

2. Map the schema: define the relationship between the source schema (what you have) and the destination schema (what you want). In FME this is accomplished by using transformers to define how the old and new schema are related.

We’ll cover these two steps in this unit and the next.

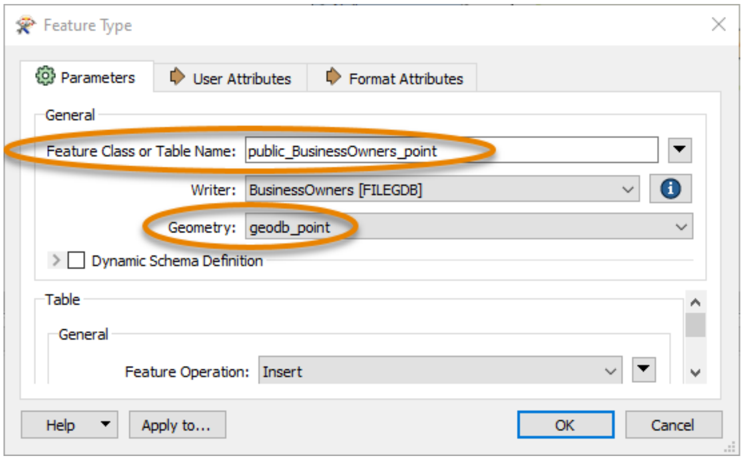
Generate Workspace has created a duplicate of the reader schema on the writer schema in Jennifer’s workspace. Now she has to edit the schema to get the results she wants. The first step is to view the writer schema. She does that by double-clicking on the **public_BusinessOwners_point** feature type to open its dialog.



The writer **Feature Type** dialog contains all of the data’s schema information:

- Feature type name (in this case, **Feature Class or Table Name**, but it varies by format)
- Allowed geometries (if any, in this case under **Geometry**)

- Attribute names (in the **User Attributes t**ab under the **Name** column)
    - They are now all lowercase, matching PostGIS specifications
- Attribute data types (in the **User Attributes** tab under the **Type** column)



General

Feature Class or Table Name: public_BusinessOwners_point

Writer: BusinessOwners [FILEGDB]

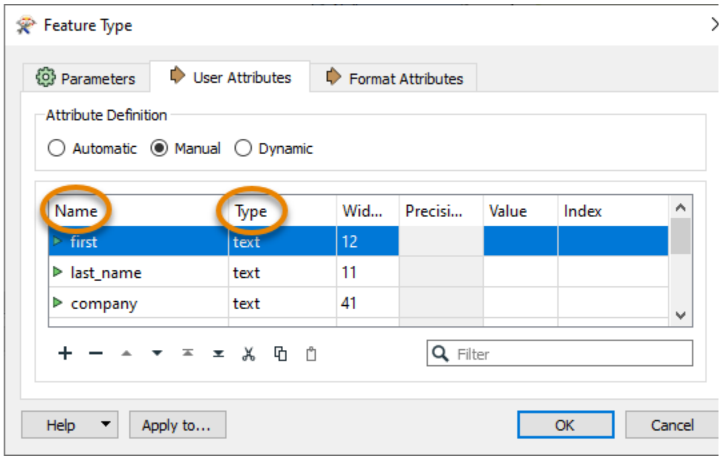
Geometry: geodb_point

□ Dynamic Schema Definition

Table

General

Feature Operation: Insert



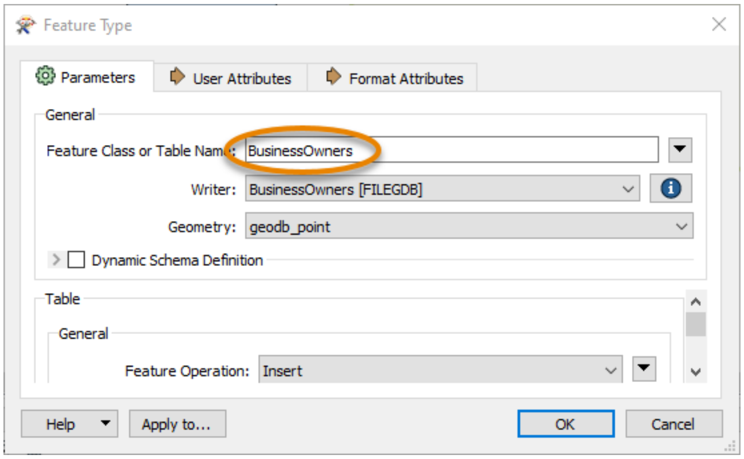
Attribute Definition

○ Automatic ● Manual ○ Dynamic

Name	Type	Wid...	Precisi...	Value	Index
first	text	12			
last_name	text	11			
company	text	41			

Rename Output

Jennifer wants to rename the feature types to “BusinessOwners” and “BusinessOwners_incomplete” to distinguish features with and without geometry. She does this by renaming **Feature Class or Table Name** to “BusinessOwners”.



General

Feature Class or Table Name: BusinessOwners

Writer: BusinessOwners [FILEGDB]

Geometry: geodb_point

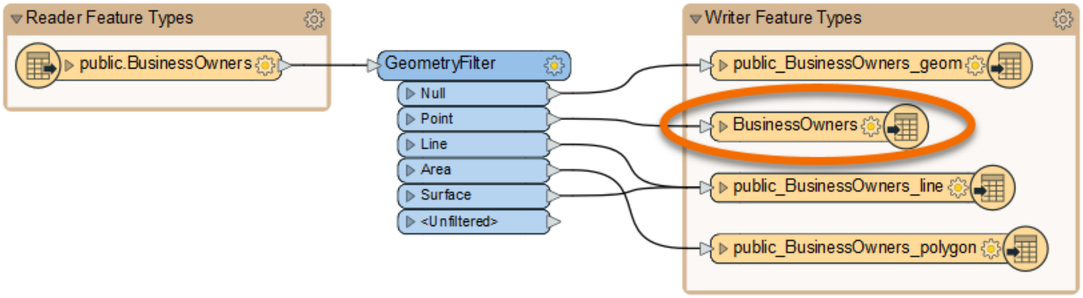
□ Dynamic Schema Definition

Table

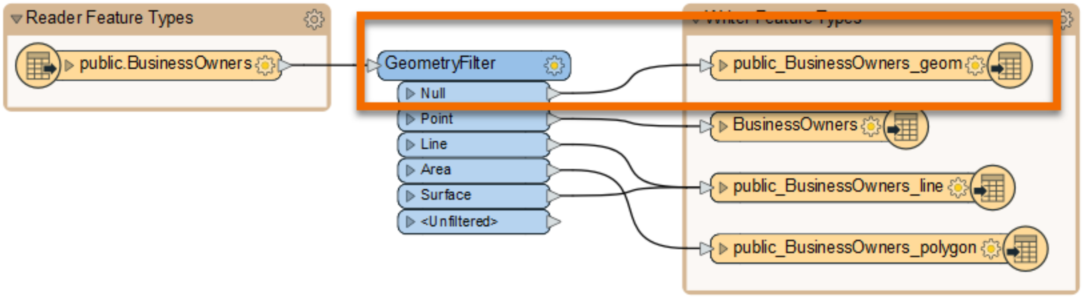
General

Feature Operation: Insert

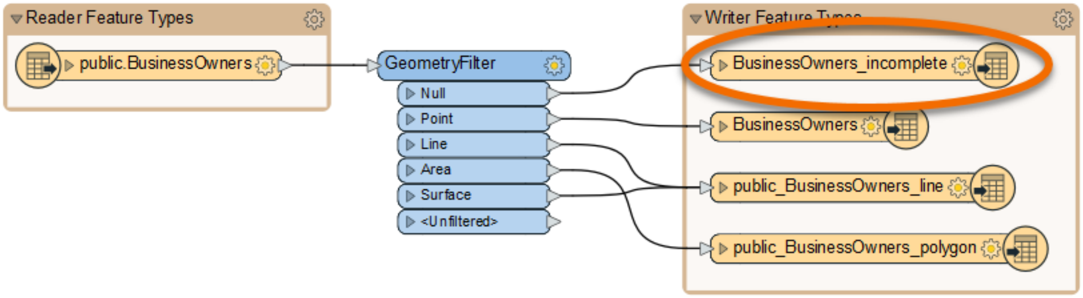
After changing the text, she clicks **OK**. The name of the feature type is updated on the canvas.



Jennifer can see some of her data is missing geometry. She can tell because the **GeometryFilter** has results coming out of the **Null** port, which means that those features are neither points, lines, or polygons.



She wants to route the features missing geometry to their own feature class so the Quality Assurance team can fix them. Therefore, she renames the **public_BusinessOwners_geom** feature type to “BusinessOwners_incomplete”.

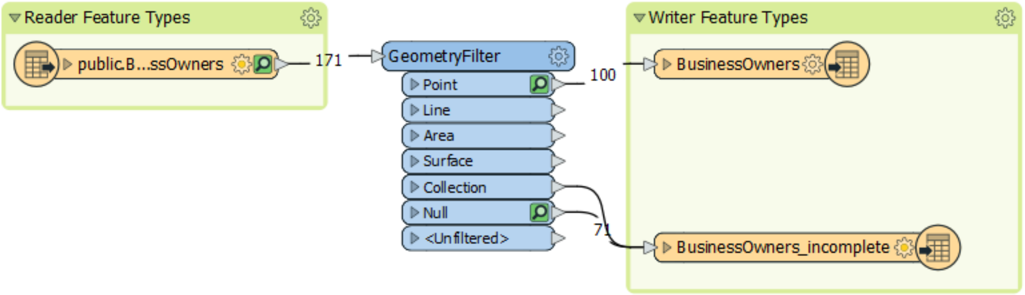


Now when the geodatabase is written, it will have these names for its feature classes.

In some cases, the name of a feature type will be the same name of the file created. In other cases, such as with files that have multiple tables or layers, the feature type will indicate the name of the table or layer, not the file. The first parameter in all writer feature type dialogs indicates what will be created, e.g. **Feature Class or Table Name** for geodatabases or **Sheet Name** for Excel. You can set the output file or folder name in the first parameter under the writer in the **Navigator**.

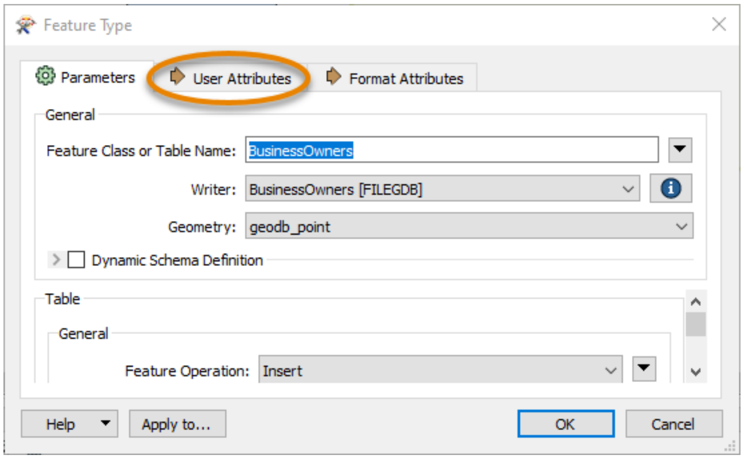
Remove Empty Writer Feature Types

Jennifer takes a look at the feature counts along the connection lines coming out of the **GeometryFilter**. She notices there are no features coming out of the **Line**, **Area**, **Surface**, **Collection**, or **<Unfiltered>** ports. That means there are only point and null geometries in the source data. It is therefore unnecessary to handle features from the **Line**, **Area**, and **Surface** output ports, as FME won’t create empty feature classes. Since nothing is being written to these feature types, she deletes the **public_BusinessOwners_line** and **public_BusinessOwners_polygon** feature types by right-clicking them and selecting **Delete**. Now her workspace looks like this:



Edit Writer Feature Type Attributes

Now that Jennifer has edited the feature type names, she would also like to change the attributes. She double-clicks the **BusinessOwners** feature type to open the dialog. She clicks on the **User Parameters** tab to view the attributes she wants to edit.



General

Feature Class or Table Name: BusinessOwners

Writer: BusinessOwners [FILEGDB]

Geometry: geodb_point

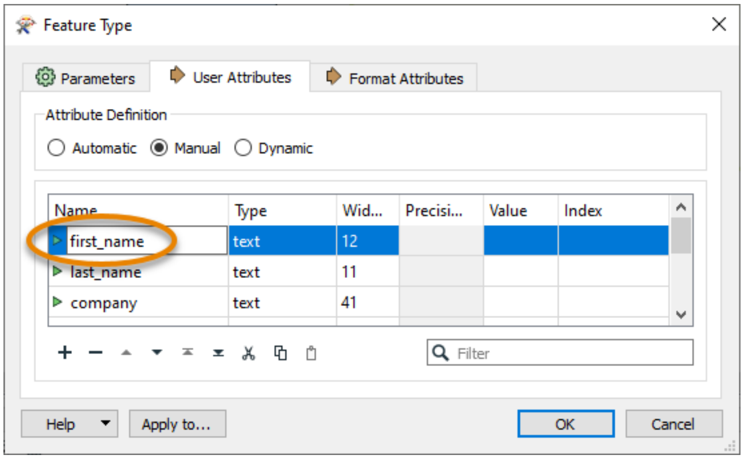
□ Dynamic Schema Definition

Table

General

Feature Operation: Insert

The table listing them can be edited. She can rename them, change their type, reorder them, or add a new attribute. She renames "first" to "first_name" by typing in the **Name** cell of the table.

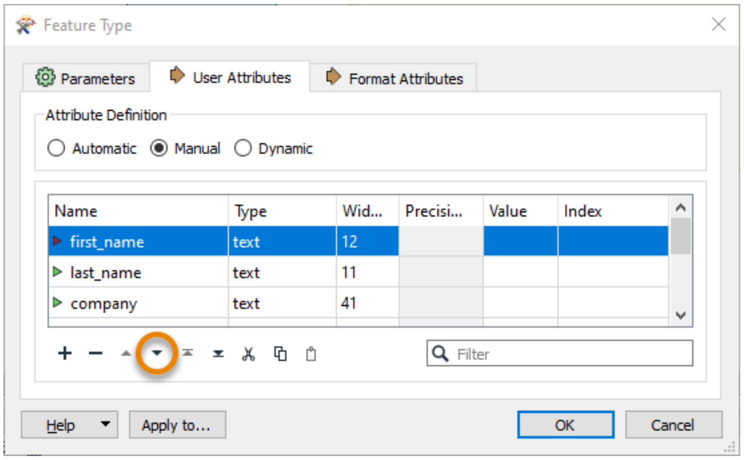


Attribute Definition

○ Automatic ● Manual ○ Dynamic

Name	Type	Wid...	Precisi...	Value	Index
first_name	text	12			
last_name	text	11			
company	text	41			

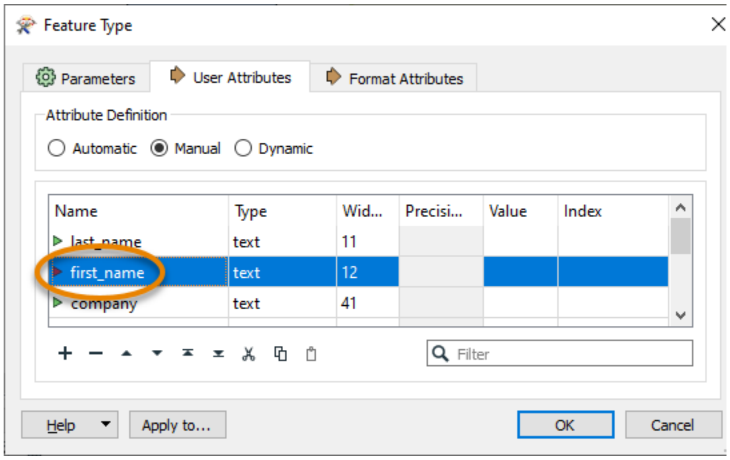
She also changes the order of the attributes, clicking on a row to select it and then using the up and down triangle buttons at the bottom of the table to change their order. She uses the **Move Down** button to move "first_name" down one spot so it comes after "last_name".



Attribute Definition

○ Automatic ● Manual ○ Dynamic

Name	Type	Wid...	Precisi...	Value	Index
first_name	text	12			
last_name	text	11			
company	text	41			

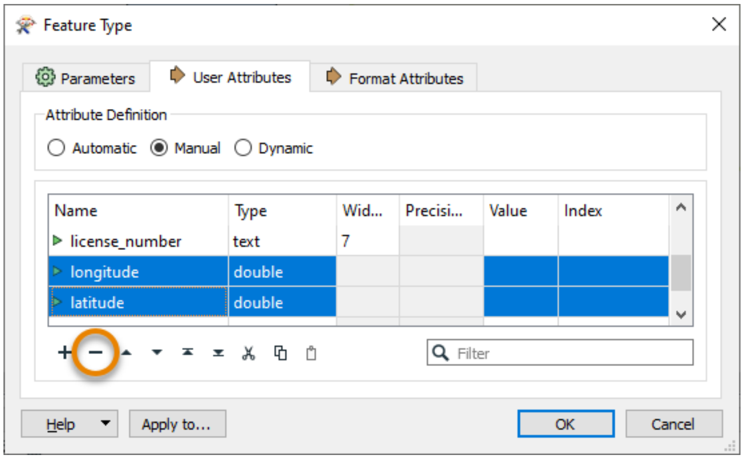


Attribute Definition

○ Automatic ● Manual ○ Dynamic

Name	Type	Wid...	Precisi...	Value	Index
last_name	text	11			
first_name	text	12			
company	text	41			

Jennifer no longer wants the latitude and longitude attributes written to her feature class, so she deletes those attributes from the User Attributes tab of the writer feature type dialog. She selects their rows and clicks the **Remove Row** button.



Attribute Definition

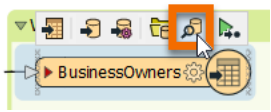
○ Automatic ● Manual ○ Dynamic

Name	Type	Wid...	Precisi...	Value	Index
license_number	text	7			
longitude	double				
latitude	double				

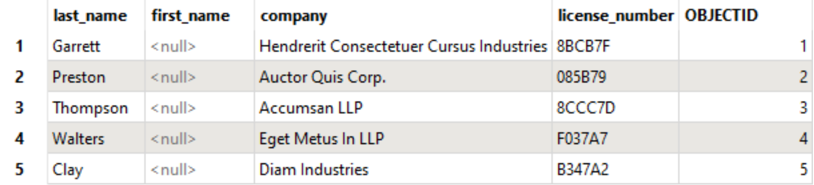
She clicks **OK** to exit the dialog and apply the changes.

View Data with Edited Schema

Jennifer wants to preview her data with the edited schema. She clicks **Run** and once the workspace is finished, she clicks her **BusinessOwners** feature type once to select it and then clicks **View Written Data**.



The data appears in **Visual Preview**. She can see in **Table View** that the schema has been edited, but some of the data is missing. For example, “last_name” is in the correct position, but “first_name” doesn’t have any values.



	last_name	first_name	company	license_number	OBJECTID
1	Garrett	<null>	Hendrerit Consectetuer Cursus Industries	8BCB7F	1
2	Preston	<null>	Auctor Quis Corp.	085B79	2
3	Thompson	<null>	Accumsan LLP	8CCC7D	3
4	Walters	<null>	Eget Metus In LLP	F037A7	4
5	Clay	<null>	Diam Industries	B347A2	5

In order to provide the right values to the written data, Jennifer has to map her schema, connecting the source and destination schema.

Exercise

Make sure you have followed along with Jennifer’s steps.

1Where can you view the schema of your source data, i.e. the data you start with?

○ A.The Transformer Gallery
○ B.The writer entry in the Navigator
○ C.The reader feature type dialog
○ D.The writer feature type dialog

2Opening the writer feature type dialog and adding a new attribute is a form of:

○ A.Schema mapping
○ B.Schema editing
○ C.Structure componentizing
○ D.Data refreshing

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Map Data's Schema

Learning Objectives

After completing this unit, you'll be able to:

- Explain the conceptual role of a transformer in FME.
- Use a transformer to edit features' schema.
- Automatically adopt features' schema by using Automatic Attribute Definition mode.

Resources

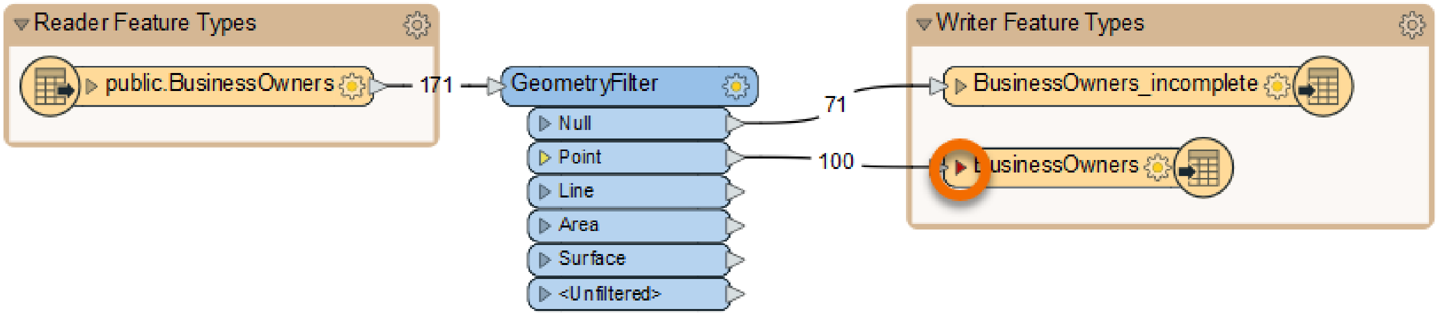
- Interactive tutorial to help you follow along
- Starting workspace

Video

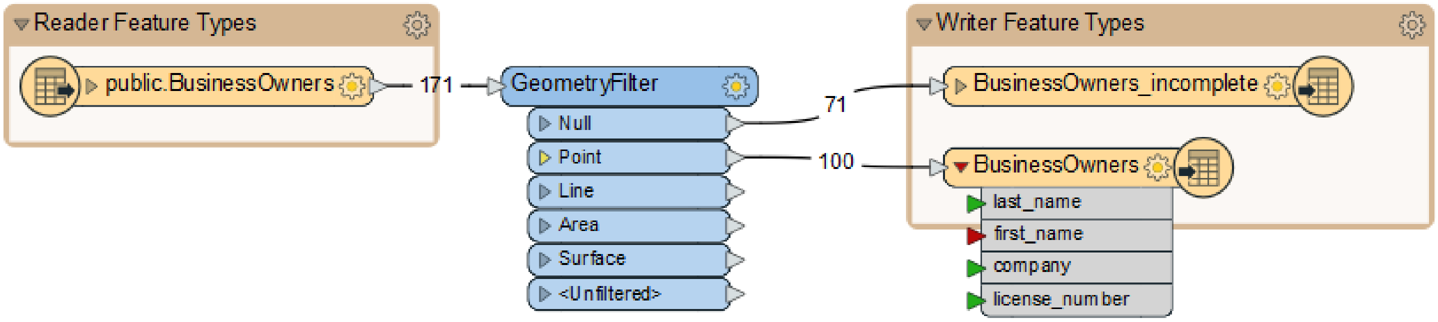
Determine Schema Mapping Goals

Now that Jennifer has created a workspace that edits the PostGIS data's schema, she has to map it, telling FME how the original and new schemas are related.

She opens the workspace in FME Workbench (2022.0 or later) and notices the triangle icon on her writer feature type has turned red.



She clicks the triangle to expand the list of attributes entering the feature type.



FME uses colored ports to indicate the status of schema mapping. In this case, the red port means an attribute exists on the writer feature type, but not on the features entering the feature type. An attribute with a red port will not cause an error when the workspace is run, but it will not have any values in the written data.

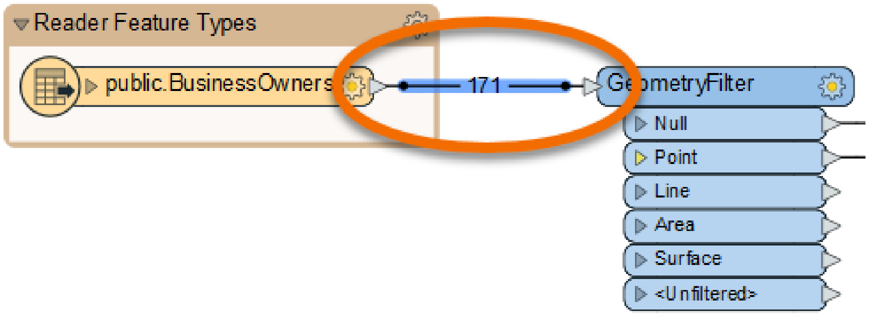
To ensure her new **first_name** attribute gets the same values as her existing **first** attribute, we have to map the new schema onto the old schema. We can use the AttributeManager transformer for that.

Colored ports are used to aid schema mapping visually:

- Green ▶: this attribute is connected.
- Yellow ▶: this reader feature type attribute is not mapped to any writer feature type; therefore, this attribute will not be in the output.
- Red ▶: this writer feature type attribute is not connected. While it exists in the schema, it will not receive any data and therefore will not have any values in the written data.

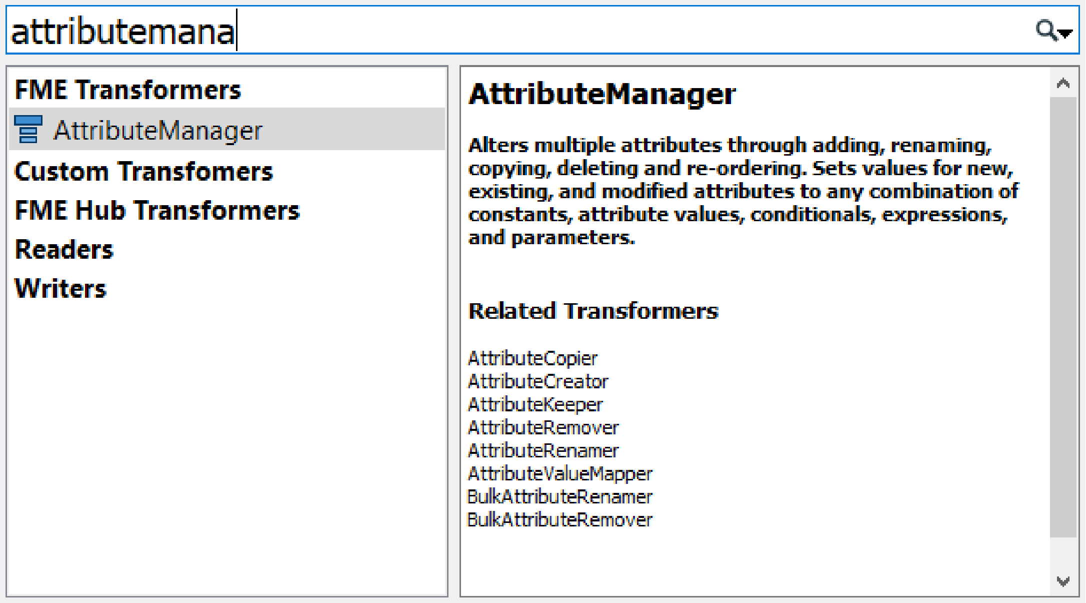
Add an AttributeManager

Jennifer clicks the black **feature connection line** between the reader feature type and the **GeometryFilter** to select it. She observes the light blue highlight indicating the line is selected.



Feature connection line: these lines connect feature types and transformers on the canvas and control the flow of features from left to right.

When any object on the canvas is selected, using Quick Add will automatically connect the new object. With the feature connection line selected, Jennifer types in “AttributeManager”. The Quick Add dialog appears, letting her search transformers, readers, and writers. She finds the AttributeManager and presses Enter to add it.



attributemana

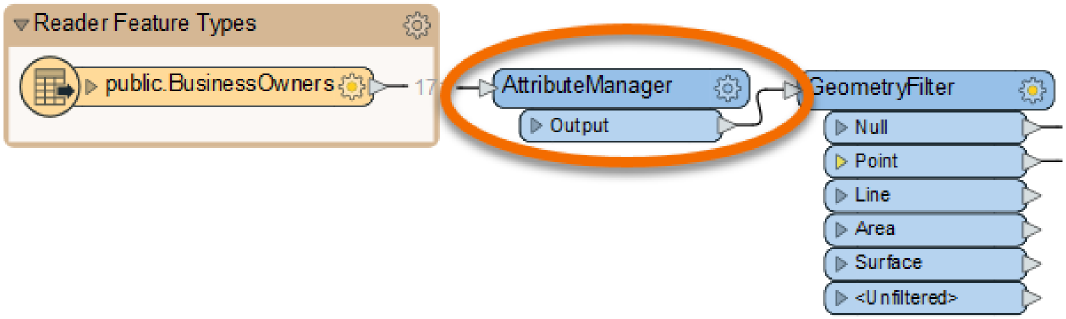
AttributeManager

Alters multiple attributes through adding, renaming, copying, deleting and re-ordering. Sets values for new, existing, and modified attributes to any combination of constants, attribute values, conditionals, expressions, and parameters.

Related Transformers

AttributeCopier
AttributeCreator
AttributeKeeper
AttributeRemover
AttributeRenamer
AttributeValueMapper
BulkAttributeRenamer
BulkAttributeRemover

The AttributeManager appears on the canvas. Jennifer double-clicks it to open its Parameters dialog.



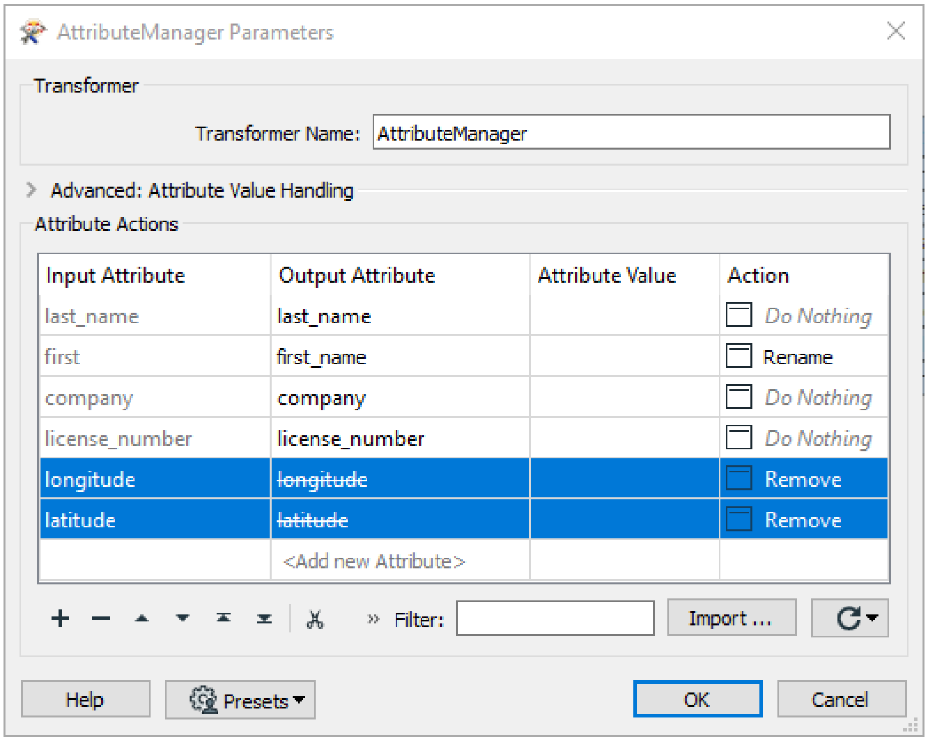
All transformers have parameters, which control how they operate. These parameters are unique to each transformer. The parameters button is also color-coded. Red means that there are parameters that must be set, yellow means that default settings will be used, but you haven't reviewed them, and blue means that you've set or reviewed the parameter settings.

Map Schema with an AttributeManager

The **AttributeManager** parameters are a table that defines how to modify attributes. It allows you to create new attributes, edit existing attribute names, change their order, and set their values.

Jennifer plans to use this transformer to change incoming features so their schema matches the writer feature type.

First, she clicks in the **Output Attribute** column for the first attribute and renames it “first_name”. Then she clicks the **Move Down** button to change the attribute order. Then she Shift + clicks the **latitude** and **longitude** attribute rows and clicks the **Remove Row** button to delete them. Her dialog looks like this:



AttributeManager Parameters

Transformer

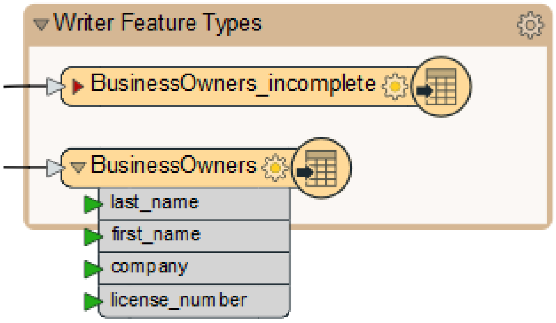
Transformer Name: AttributeManager

› Advanced: Attribute Value Handling

Attribute Actions

Input Attribute	Output Attribute	Attribute Value	Action
last_name	last_name		□ *Do Nothing*
first	first_name		□ Rename
company	company		□ *Do Nothing*
license_number	license_number		□ *Do Nothing*
longitude	~~longitude~~		□ Remove
latitude	~~latitude~~		□ Remove
	<Add new Attribute>		

She clicks **OK**. Now the attribute ports on the writer feature type all turn green, showing her schema is mapped.

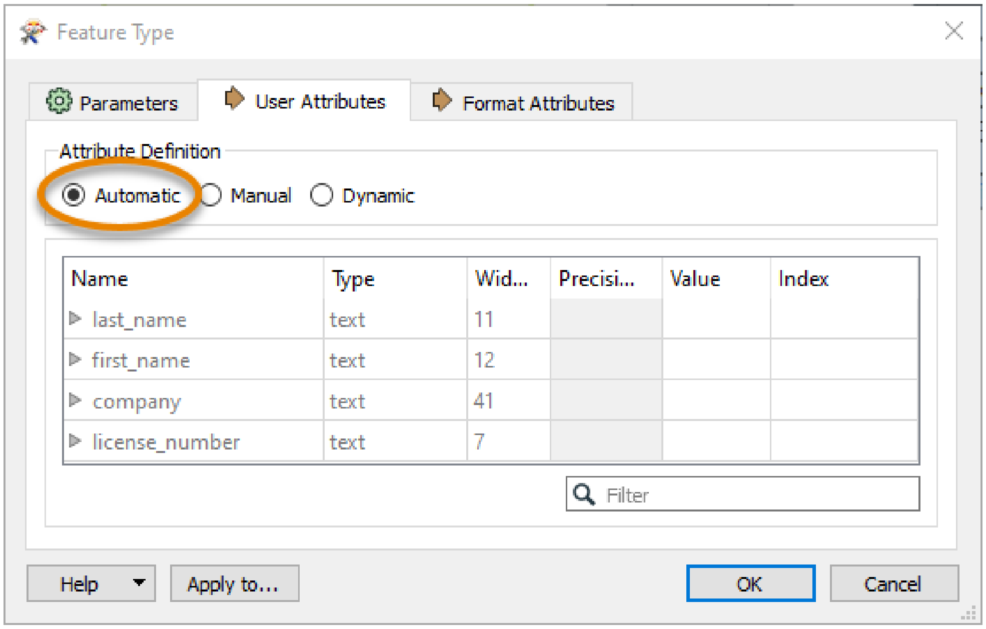


However, Jennifer notices a slight problem: **BusinessOwners** looks good, but **BusinessOwners_incomplete** still has its original schema. How can she quickly update the schema to match the incoming features?

Jennifer knows she can use **Automatic Attribute Definition** mode for this task. She double-clicks on **BusinessOwners_incomplete** and then clicks the **User Attributes** tab. The first section of this dialog is **Attribute Definition**. By default, this is in **Manual** mode. However, changing this to **Automatic** mode will tell the feature type to simply adopt the schema of whatever features are connected to it.

Why didn't Jennifer do this first if it was so easy? Well, she had to first edit the schema upstream using the **AttributeManager**. As the data flows from the reader feature type, it gets edited in the **AttributeManager**. These changes will be reflected automatically in the writer feature type when it is set to **Automatic** mode. If she left it on **Manual**, she would have to repeat the edits on the writer feature type.

Jennifer clicks **Automatic** and the schema is updated.



Attribute Definition

● Automatic ○ Manual ○ Dynamic

Name	Type	Wid...	Precisi...	Value	Index
last_name	text	11			
first_name	text	12			
company	text	41			
license_number	text	7			

If she then wants to make changes, she could click **Manual** again. She leaves it in **Automatic** mode and clicks **OK**. Then she clicks **Run**.

Dynamic mode is a more advanced **Attribute Definition** mode that lets you write out data without knowing the schema in advance. Both of these modes can also be set when you use **Generate Workspace** or add a writer.

Learn More

Exercise

Make sure you have followed along with Jennifer's steps.

1What are objects in FME that let you change the content or structure of your data?

○ A.Feature Types
○ B.Tools
○ C.Transformers
○ D.Editors
○ E.Readers and writers

2Why did Jennifer use the AttributeManager in her workspace?

○ A.The features going into the writer feature types did not match the edited schema.
○ B.The reader feature type had the correct schema.
○ C.The workspace was running slowly.
○ D.She edited the schema on the reader feature type and needed to confirm it with the AttributeManager.

3Why did Jennifer change the Attribute Definition mode from Manual to Automatic on the BusinessOwners_incomplete feature type?

○ A.This mode automatically improves data quality.
○ B.She wanted to skip using the AttributeManager for this feature type.
○ C.The Manual mode is incompatible with the Esri geodatabase format
○ D.She wanted to automatically update the schema for the BusinessOwners_incomplete feature type.

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Author and Debug Workspaces Efficiently

Learning Objectives

After completing this unit, you'll be able to:

- Inspect features at any point in your workflow with feature caching.
- Speed up workspace authoring with partial runs.
- Quickly author and debug your workspaces.

Resources

- Interactive tutorial to help you follow along
- Starting workspace
- Complete workspace

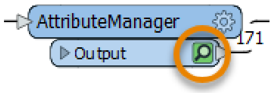
Video

View Your Data at Any Point in the Workspace

Jennifer has edited her data's schema using an **AttributeManager**. She knows that the data changed because she can see the colored attribute ports, which show successful schema mapping. However, Jennifer wants to see the changes to her data in **Visual Preview**.

Jennifer can do this using feature caching. Feature caching is an authoring mode set on the **Run** menu and enabled by default. When enabled, a local cache of data is stored at every output port in the workspace. These caches let you view and compare data at any point in your workspace.

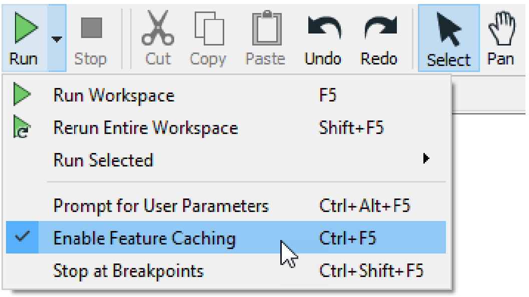
Jennifer continues to work in FME Workbench (2022.0 or later). Now that she has set up her **AttributeManager** and run the workspace, Jennifer sees a cache on the transformer. Caches are indicated by a green magnifying glass icon. She clicks it to open that cache in **Visual Preview**.



As covered in a previous module, you can also inspect caches by clicking any object with a cache (feature type or transformer) as long as **Automatic Inspect on Selection** is enabled. Note, however, that this inspects all the caches on that object at once. You'll have to use the **Display Control** window and the drop-down menu in **Table View** to control which cache you are inspecting.

Feature caching is great when you are authoring a workspace. It lets you use iterative and incremental development to add one transformer or feature type at a time, create a cache, and inspect it to confirm the data looks as you expect. It is particularly useful when working with web, database, or compressed data, allowing you to download, query, or extract the data once and work with a cache, saving time and effort when reading large datasets or making API calls. However, creating these caches takes time, so it's wise to disable this mode when you want FME to run at peak efficiency.

Feature caching can be disabled by clicking the drop-down triangle next to **Run** to access the **Run** menu, then deselecting **Enable Feature Caching**.

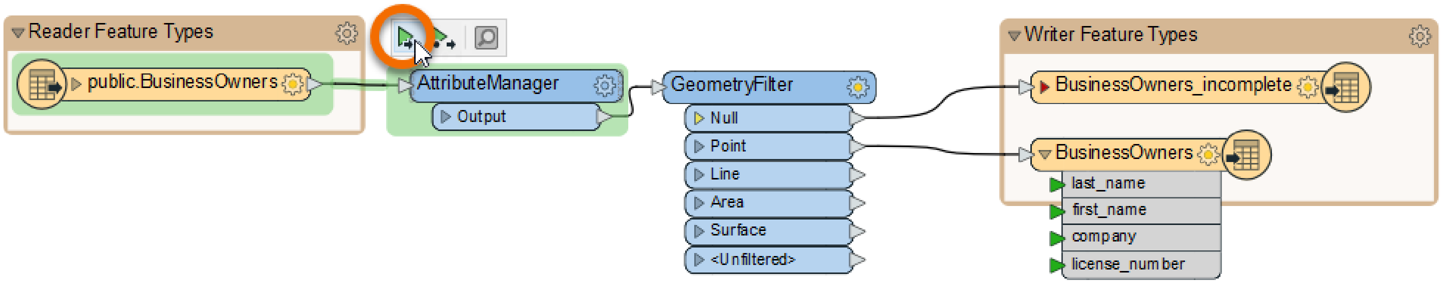

Run
Stop
Cut
Copy
Paste
Undo
Redo
Select
Pan
Run Workspace F5
Rerun Entire Workspace Shift+F5
Run Selected
Prompt for User Parameters Ctrl+Alt+F5
✓ Enable Feature Caching Ctrl+F5
Stop at Breakpoints Ctrl+Shift+F5

Debug Your Workspace with Partial Runs

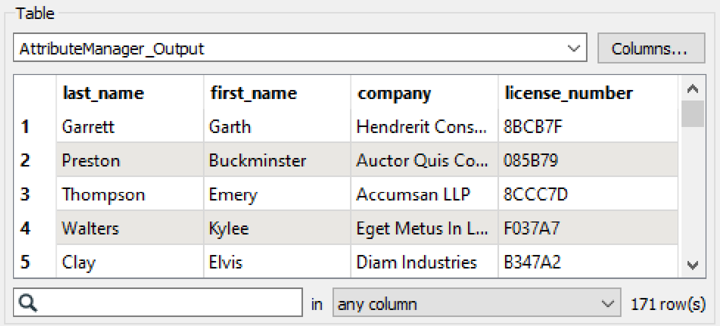
Jennifer will use another feature, partial runs, to speed her authoring of workspaces and debug them as she works.

Partial runs allow you to run specified sections of your workspace instead of the entire thing. This feature works in tandem with feature caches to enable incremental development. When you add a new transformer or feature type, you can run that new object independently and inspect its cache for any problems. Authoring workspaces this way saves time by allowing you to detect and fix problems early.

Jennifer wants to run her workspace to ensure the **AttributeManager** worked properly. She could just click **Run** > **Rerun Entire Workspace**. However, as workspaces grow in size, it's better to just run the section that changed. She does this by clicking the **AttributeManager** to select it, then clicking **Run To This** above the transformer. She can see what part of the workspace will run because it is highlighted in green.



The relevant section of the workspace runs. When it's finished, **Visual Preview** will show the features coming out of the **AttributeManager** output port. Jennifer looks at **Table View** to ensure that the schema appears as she intended. It does.



	last_name	first_name	company	license_number
1	Garrett	Garth	Hendrerit Cons...	8BCB7F
2	Preston	Buckminster	Auctor Quis Co...	085B79
3	Thompson	Emery	Accumsan LLP	8CCC7D
4	Walters	Kylee	Eget Metus In L...	F037A7
5	Clay	Elvis	Diam Industries	B347A2

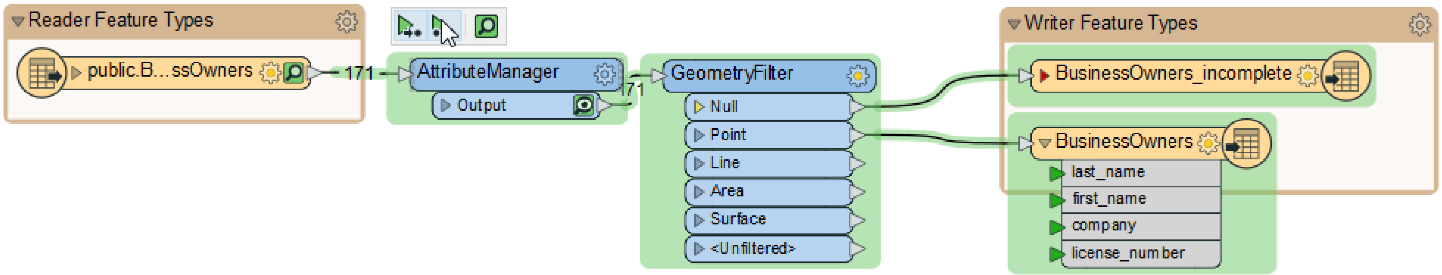
Partial runs will utilize any connected caches that exist "upstream" (i.e. earlier in the flow of data). As few transformers or feature types will run as are required to carry out the partial run, saving Jennifer time. For example, if Jennifer had already run the **AttributeManager**, using **Run To This** on the **GeometryFilter** would use the cache from the **AttributeManager**.

If any changes have been made to the workspace, any "downstream" (i.e. later in the flow of data) caches turn yellow, indicating they are invalid. These are known as invalid caches and can be rerun to make them valid again. For example, if a change was made to the **AttributeManager**, its cache and that of the **GeometryFilter** would become invalid, but the upstream reader feature type cache would remain valid.

Other options for partial runs exist depending on the object's location on the canvas:

- **Run From This** (or **Selected** if multiple objects are selected): all parts of the workspace downstream of a selected object (or objects) run.
- **Run Just This** (or **Selected**): only a selected object (or objects) runs.
- **Run To This** (or **Selected**): all parts of the workspace that are upstream of a selected object (or objects) run.
- **Run Between Selected**: all parts of the workspace between selected objects run.

Now that she's confirmed the schema is correct, Jennifer wants to write the data, so she hovers over the **Run** button. Notice how the reader feature type is not highlighted in green. FME will use its cache instead of reading the data again. When she clicks **Run**, the rest of the workspace runs, and FME writes her data.



If Jennifer made changes to her workspace and wanted to update all the caches, she could choose **Rerun Entire Workspace** from the **Run** menu.

If the source database changed and she wanted to update the data, she could find the **FME Training PostGIS Database [POSTGIS] Reader** in the **Navigator**, right-click it, and select **Update**. When the underlying data changes additional features matching the existing reader schema will be read in, but if the schema changes (e.g. new table, new attributes, renamed attributes, etc.), these changes will only be reflected if the reader is updated.

Learn more about updating readers and writers

Learn how to make dynamic workspaces that can automatically account for schema changes

Exercise

Make sure you have followed along with Jennifer's steps.

1To inspect a feature cache in Visual Preview, click the:

○ A.Green magnifying glass icon
○ B.Transformer cogwheel icon
○ C.Run button
○ D.Canvas

2Partial runs can help speed up the authoring of workspaces by:

○ A.Allowing you to only run part of your workspace.
○ B.Allowing you to update a single cache.
○ C.Allowing you to quickly inspect data you just added to the canvas.
○ D.Allowing you to create a cache of web, database, or compressed data.
○ E.All of the above.

3You just made a change to a transformer in the middle of your workspace and want to rewrite the output data to reflect the change. Which partial run button would you use after clicking on the transformer where you made changes?

○ A.Run To This
○ B.Run From This
○ C.Run Just This
○ D.Run Between Selected

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Find the Right Transformer

Learning Objectives

After completing this unit, you'll be able to:

- Understand the resources available to help you find transformers for your task.
- Use these resources to find a transformer to solve a data problem.

You don't have to follow along with Jennifer using FME in this unit. Instead, read her story and then do the exercise yourself.

Resources

- Starting workspace
- Complete workspace
- Optional example workspace that shows how Jennifer queries the MetaWeather API

Video

The Challenge

When Jennifer first began using FME, she found the list of almost 500 transformers a bit daunting. One of the most common challenges new FME users face is finding the right transformer for a given task. However, Jennifer learns that most users only focus on the subset of transformers that are relevant to their day-to-day workflow. Since Jennifer doesn't plan on working with raster data in the near future, there is a whole category of raster-dedicated transformers that she doesn't need to know. She knows she doesn't need to be familiar with every single transformer to use FME effectively.

Transformer Resources

As she learned FME, Jennifer found the following resources useful for learning about transformers:

Resource	Why use it?
Searching using Quick Add in FME Workbench	Quickly search transformers, see the Help, and add to the canvas to try them out.
Browsing or searching using the Transformer Gallery on safe.com	Filter transformers by category and sort by most-used, plus links to Help.
Browsing or searching using the Transformer Gallery in FME Workbench	Filter transformers by category and search directly in Workbench.
Using the FME Transformer Reference Guide	Contains snippets from Help to explain transformer use-cases.

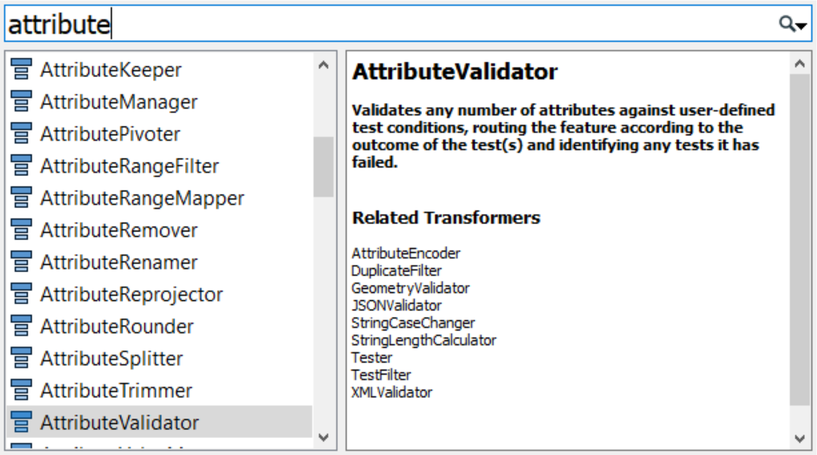
Finding a Transformer

Jennifer needs to find a new transformer for a different workspace that retrieves today's weather from the MetaWeather API. She finds that the results of the API call (a web connection request for data located on the web) contain attributes that she doesn't need. From her experience, she knows it would be tedious to manually remove them with an **AttributeManager** or **AttributeRemover**. She turns to her resources to find a transformer to help her.

You can view the optional example workspace linked in **Resources** above to see Jennifer's MetaWeather workspace, but this example is just for reference.

First, she tries searching for some keywords in **Quick Add**. She opens her workspace and starts typing keywords. First, she tries "deleter," but this search is too specific and doesn't return any results related to attributes. By default, **Quick Add** searches for transformer names only, not descriptions. She presses the **Tab** key to search in the **Help** text, but this results in transformers that don't meet her requirements.

She tries some more keywords in both search modes: "attribute," "remover," "eraser," "simplifier," etc. Some of these searches produce useful results, but they still contain too many transformers.


attribute

AttributeKeeper
AttributeManager
AttributePivoter
AttributeRangeFilter
AttributeRangeMapper
AttributeRemover
AttributeRenamer
AttributeReprojector
AttributeRounder
AttributeSplitter
AttributeTrimmer
AttributeValidator

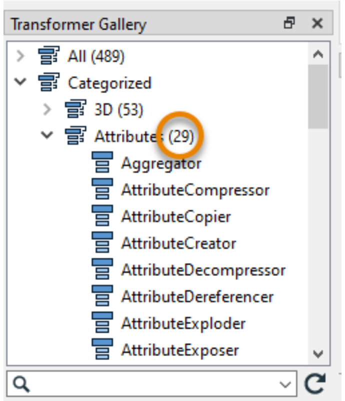
AttributeValidator

Validates any number of attributes against user-defined test conditions, routing the feature according to the outcome of the test(s) and identifying any tests it has failed.

Related Transformers

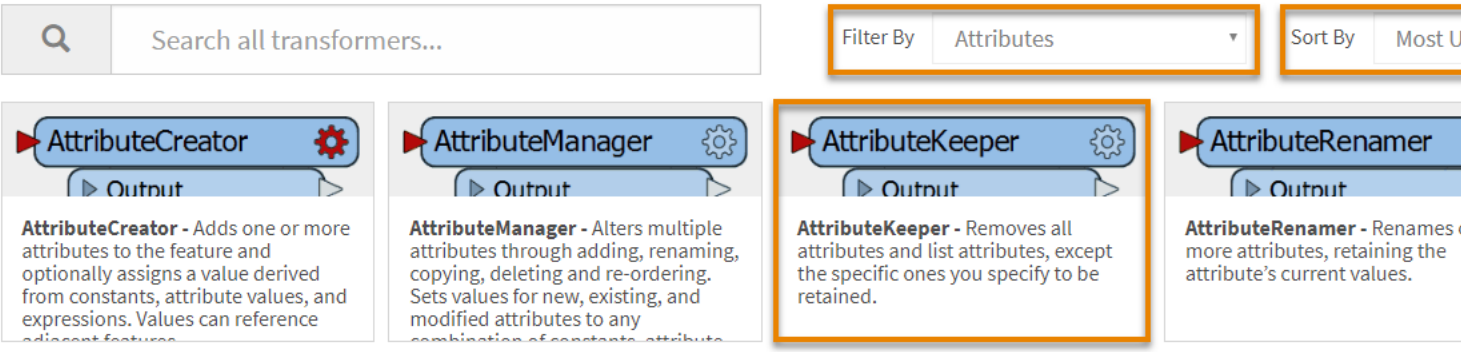
AttributeEncoder
DuplicateFilter
GeometryValidator
JSONValidator
StringCaseChanger
StringLengthCalculator
Tester
TestFilter
XMLValidator

Jennifer decides to look in the Workbench **Transformer Gallery** categories. She sees the **Attribute** category, but it still has a lot of transformers in it. Again, she could look through this list and read the **Help** text, but it could take a while to find what she is looking for.


Transformer Gallery

All (489)
Categorized
3D (53)
Attributes (29)
Aggregator
AttributeCompressor
AttributeCopier
AttributeCreator
AttributeDecompressor
AttributeDereferencer
AttributeExploder
AttributeExposer

Jennifer now turns to the online Transformer Gallery. Here she restricts the category to **Attributes**. With the default **Sort By** set to "Most Used", she scans the top attribute transformers and their descriptions. This method produces a promising result: the **AttributeKeeper**, one of the most-used attribute transformers! Its description reads, "Removes all attributes and list attributes, except the specific ones you specify to be retained." That sounds like it should work.


Search all transformers...

Filter By Attributes

Sort By Most U

AttributeCreator
Output

AttributeCreator - Adds one or more attributes to the feature and optionally assigns a value derived from constants, attribute values, and expressions. Values can reference adjacent features

AttributeManager
Output

AttributeManager - Alters multiple attributes through adding, renaming, copying, deleting and re-ordering. Sets values for new, existing, and modified attributes to any combination of constants, attribute

AttributeKeeper
Output

AttributeKeeper - Removes all attributes and list attributes, except the specific ones you specify to be retained.

AttributeRenamer
Output

AttributeRenamer - Renames more attributes, retaining the attribute's current values.

Jennifer goes back to FME Workbench and adds an **AttributeKeeper** using **Quick Add.** She connects it to her workflow, sets the parameters, and finds it does just what she wanted. She can choose a few attributes she wants to keep and remove the rest.

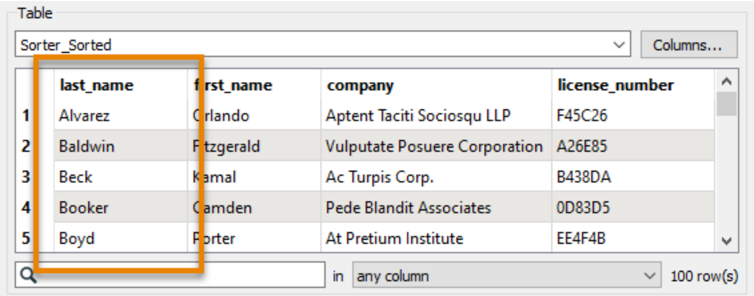
If Jennifer's use case was more complex, she might have considered using the Transformer Reference Guide or posting on the Community Forums for help.

The **AttributeKeeper** is a performance booster for your workspaces. Use it early in your workspace to remove any unnecessary attributes. Not having to process those will let your workspace run much faster.

Exercise

You've read about Jennifer's search for the **AttributeKeeper**. Now it's your turn!

Using the techniques above and the starting workspace (in FME Workbench 2022.0 or later), try to find a transformer that will order your features in ascending alphabetical order by "last_name". Once you find it, connect it between the **GeometryFilter** and the **BusinessOwners** feature type and use it to order your features. You can use **Visual Preview** with feature caching to confirm the features are in the right order. Your output should look like this:



Table

Sorter_Sorted

	last_name	first_name	company	license_number
1	Alvarez	Orlando	Aptent Taciti Sociosqu LLP	F45C26
2	Baldwin	Fitzgerald	Vulputate Posuere Corporation	A26E85
3	Beck	Kamal	Ac Turpis Corp.	B438DA
4	Booker	Camden	Pede Blandit Associates	0D83D5
5	Boyd	Porter	At Pretium Institute	EE4F4B

in any column 100 row(s)

1Which of the following would you use if you wanted to search for a transformer and order the results by how often they are used?

○ A.FME Workbench Transformer Gallery
○ B.FME Workbench Quick Add
○ C.safe.com Transformer Gallery
○ D.FME Transformer Reference Guide

2Which transformer is used to complete the task in the above exercise?

○ A.ListSorter
○ B.Orderer
○ C.PointCloudSorter
○ D.StreamOrderCalculator
○ E.SummaryReporter
○ F.Sorter

3After completing the exercise, what is the value of last_name for the last feature in the ordered data (the end of the alphabet)?

○ A.Zalesky
○ B.Zamboni
○ C.Zimmerman
○ D.Zwick

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Filter Data By Attribute Values

Learning Objectives

After completing this unit, you'll be able to:

- Understand how filtering lets you create multiple streams of data in your workspaces.
- Filter your data using a Tester transformer.
- Inspect multiple feature caches.

Resources

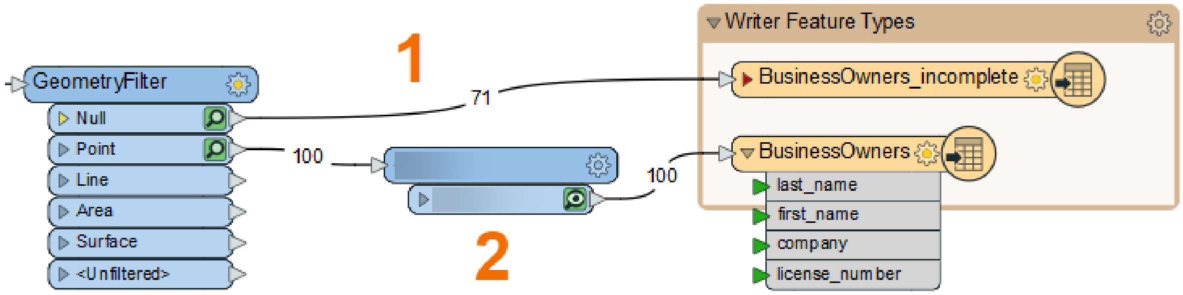
- Interactive tutorial to help you follow along
- Starting workspace
- Complete workspace

Video

Creating Multiple Streams of Data

FME workspaces send data from left to right across the canvas from reader feature types to writer feature types. The simplest workspace only has one "stream" of data: features are read in, all processed in the same way, and then written out. However, workspaces can have multiple streams of data, splitting and merging features as required.

In Jennifer's workspace, the **GeometryFilter** created multiple streams when it split the PostGIS features into point and null geometry. She now has two streams coming out of the **GeometryFilter**: #1 contains point geometry features going to the **BusinessOwners** feature type, and #2 contains features missing geometry going to the **BusinessOwners_incomplete** feature type.



The name of the transformer in stream #2 above is obscured because it is the answer to a previous quiz question.

Jennifer has been given a CSV file with a list of business license numbers that have been flagged to be revoked. She would like to use this list to filter out features with revoked licenses from her source data before writing to the geodatabase.

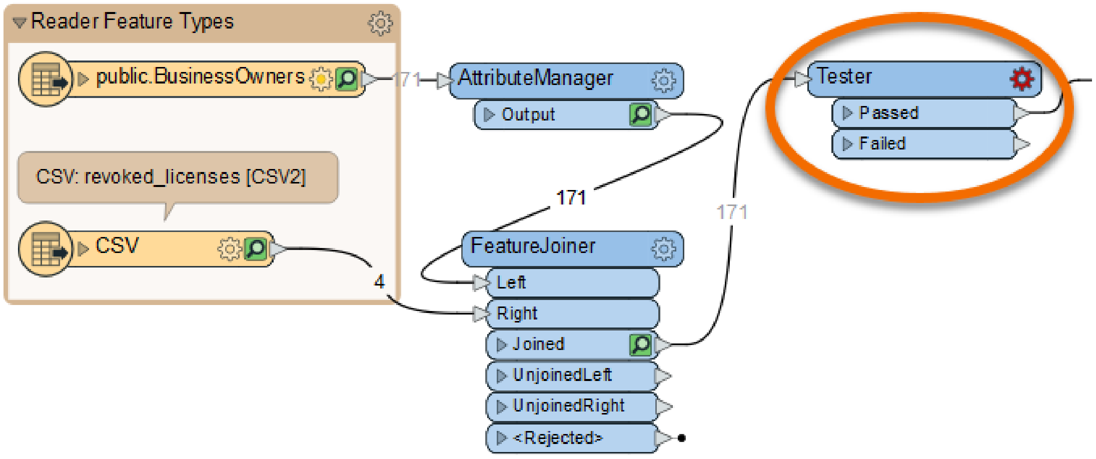
You don't have to follow along with Jennifer here. We provide a starting workspace below to let you skip this step. It's a bit more advanced. If you want to learn more about the FeatureJoiner, you can read this tutorial.

Jennifer adds a CSV reader and reads the revoked licenses file. She then uses a **FeatureJoiner** to join the revoked license data to her existing BusinessOwners data. Now her features have a "revoked_license" attribute that can be used to filter out the businesses with revoked licenses.

Filter Your Data Using a Tester

You can follow along with Jennifer now using this starting workspace in FME Workbench (2022.0 or later)

Jennifer adds a **Tester** between the **FeatureJoiner** and the **GeometryFilter** using **Quick Add** so she can filter out the stream of data to get all businesses with valid licenses only.



We are filtering our features into two streams with the **Tester**. Other transformers with "Filter" in their name are capable of routing features into more than three streams based on various operations, for example, the **TestFilter**.

To do this, she double-clicks the **Tester** to open its **Parameters** dialog. The table here allows her to enter a logical test or series of tests against incoming features. It works a bit like an "if-then-else" statement in programming languages. If the feature meets the criteria of the test(s), it comes out of the **Passed** port. If it does not, it comes out of the **Failed** port. The **Tester** is used to filter data and allows for simple branching of your data integration workflow.

Jennifer sets up her **Tester** as follows:

Logic	Left Value	Operator	Right Value
NOT	← revoked_license	Attribute Has a Value	<Unused>

These settings do the following: "For each feature being read by the **Tester**, if it does NOT have a value for the attribute "revoked_license", it passes. Otherwise, it fails." This test accomplishes our goal of sending any features with a revoked license to the **Failed** port. Jennifer uses **Run to This** on the **Tester** and sees 167 results come out of the **Passed** port and four come out of the **Failed** port.

Starting with FME 2020.1, you can now use cached values in the **Tester**. This feature makes it faster and easier to build tests, saving you the step from manually confirming attribute values. For example, Jennifer could use this feature if she wanted to filter out businesses owned by a particular person, searching the list of last names to find the correct one.

On a **Tester** with a cache, simply click the drop-down arrow on **Right Value** and choose **Cached Values**. A list of values is displayed to choose from, including a search filter.

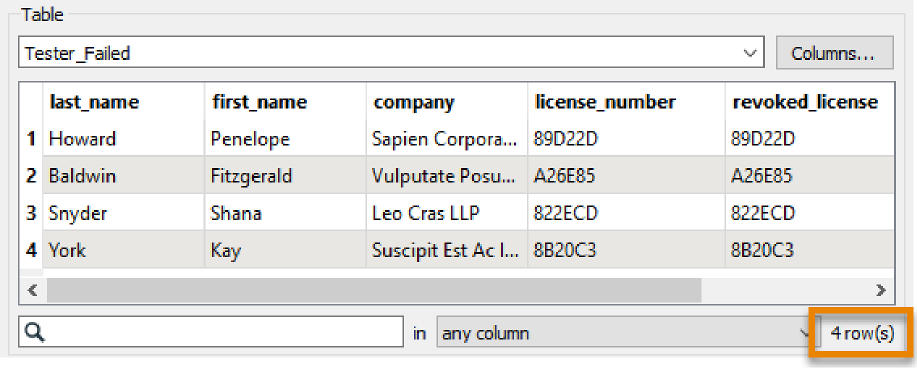
See the Tester documentation for more information.

This is a simplified example. It is more likely these revoked licenses would be removed from the PostGIS database or added to a license history table in the database. We are using a CSV file here to keep things simple.

We used the NOT Logic operator in this example. For more complex tests, you can combine different Logic operators like OR or AND.

Inspect Your Filtered Data

Jennifer inspects the **Tester** cache to make sure the correct features were filtered out. She clicks the **Tester:Failed** feature cache and sees **Visual Preview's Table View** reporting in the bottom-right corner that 4 rows are being displayed, meaning the four revoked licenses have been filtered out successfully via the **Failed** port.



Table

Tester_Failed

	last_name	first_name	company	license_number	revoked_license
1	Howard	Penelope	Sapien Corpora...	89D22D	89D22D
2	Baldwin	Fitzgerald	Vulputate Posu...	A26E85	A26E85
3	Snyder	Shana	Leo Cras LLP	822ECD	822ECD
4	York	Kay	Suscipit Est Ac I...	8B20C3	8B20C3

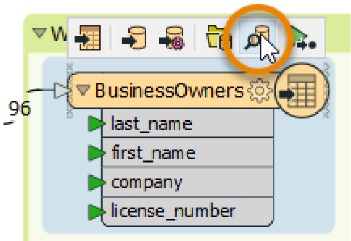
in any column 4 row(s)

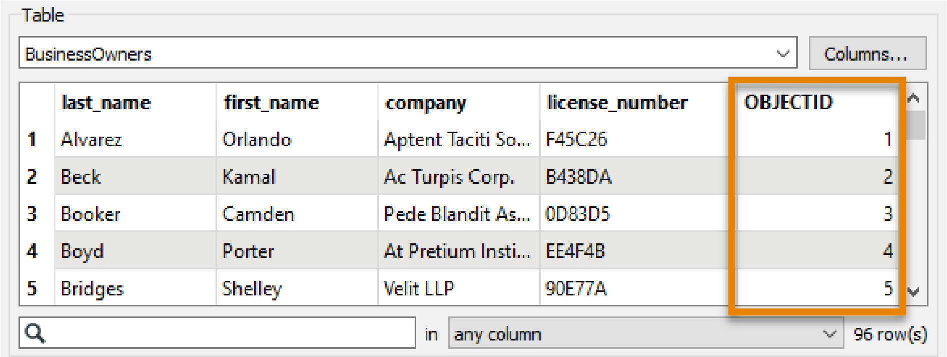
She can confirm the correct features failed because their "license_number" and "revoked_license" attributes match.

Inspect Final Results

Now that her workspace is complete, Jennifer turns off feature caching from the toolbar (**Run** > uncheck **Enable Feature Caching**) and then clicks **Run**. Her entire workspace runs successfully.

Then she clicks the **BusinessOwners** writer feature type to select it and clicks **View Written Data**. This shows the 96 valid records that were written to this feature class. Note that FME automatically added the required **OBJECTID** column to the data as required by the geodatabase format.





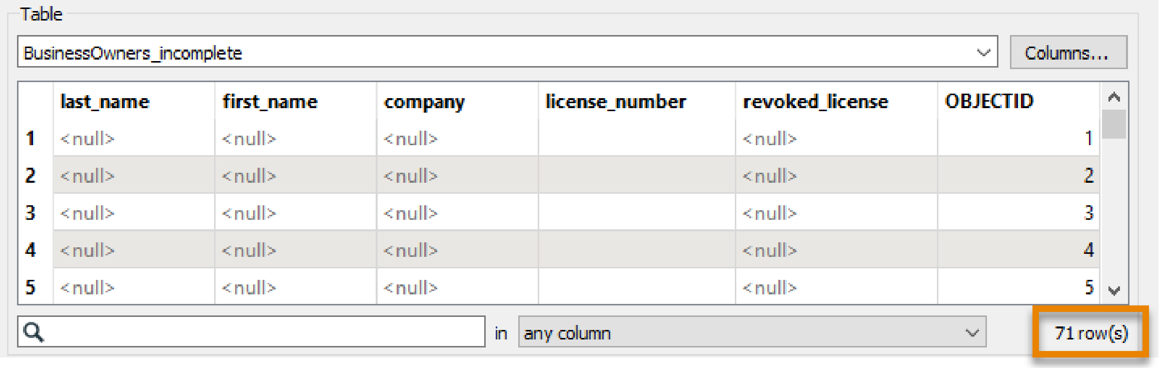
Table

BusinessOwners

	last_name	first_name	company	license_number	OBJECTID
1	Alvarez	Orlando	Aptent Taciti So...	F45C26	1
2	Beck	Kamal	Ac Turpis Corp.	B438DA	2
3	Booker	Camden	Pede Blandit As...	0D83D5	3
4	Boyd	Porter	At Pretium Insti...	EE4F4B	4
5	Bridges	Shelley	Velit LLP	90E77A	5

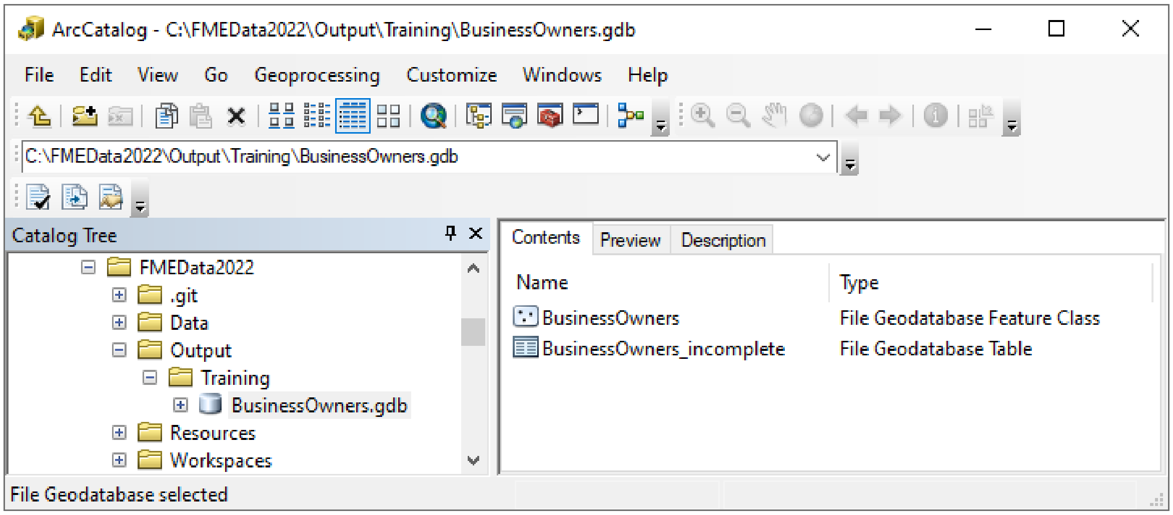
in any column 96 row(s)

Using the same method to inspect **BusinessOwners_incomplete** shows the 71 features with missing values.



	last_name	first_name	company	license_number	revoked_license	OBJECTID
1	<null>	<null>	<null>		<null>	1
2	<null>	<null>	<null>		<null>	2
3	<null>	<null>	<null>		<null>	3
4	<null>	<null>	<null>		<null>	4
5	<null>	<null>	<null>		<null>	5

Jennifer clicks one of the writer feature types and then **Open Containing Folder**, viewing the geodatabase in her file browser. From there she can open the geodatabase in ArcCatalog, ArcMap, or ArcGIS Pro.



Name	Type
BusinessOwners	File Geodatabase Feature Class
BusinessOwners_incomplete	File Geodatabase Table

Exercise

Make sure you have followed along with Jennifer's steps.

1All FME workspaces must have multiple streams of data.

○ A.True
○ B.False

2You have a Tester reading road features. There are two tests: the first tests if the road type is a highway, the second tests if the road type is an off-ramp. Which Logic option do you need for the Tester to pass both highways and off-ramps features?

○ A.AND
○ B.OR
○ C.AND NOT
○ D.OR NOT

3What happens when Automatic Inspect on Selection is enabled, and you click a transformer with multiple feature caches?

○ A.All feature caches are inspected
○ B.No feature caches are inspected
○ C.The top cache only is inspected
○ D.The bottom cache only is inspected

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Create a Self-Serve Workspace for the Web

Learning Objectives

After completing this unit, you'll be able to:

- Describe how FME Server can automate repetitive tasks.
- Design a workspace with user parameters to give the end-user control over how a workspace runs.
- Set user parameters.

Learning content in the FME Academy presents the story of a user addressing their data integration challenges with FME. **You should follow along with their actions using your own installation of FME (2022.0 or later)** or request an on-demand virtual machine in the footer link below. Some units will require you to follow their steps or to take additional steps on your own to answer a quiz question.

The **Resources** section will provide you with links to interactive tutorials and starting workspaces when necessary.

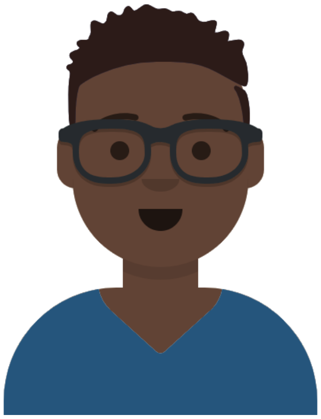
Resources

- Starting workspace
- Complete workspace

Video

Videos are provided in this module if you prefer to watch instead of reading the text below. Note that some Quiz answers might require you to read the text.

Automate Repetitive Data Integration Tasks with FME Server



Frank is a GIS Administrator working with Jennifer for a local government. He is also the FME Server Administrator for his organization. Because his town doesn't have an open data portal, he is constantly bombarded with emails from other departments and the public asking for GIS data. He has to read through the emails, find out what layer people are requesting, and manually send it to them. People most often request layers from the community resources geodatabase, which contains data about street food vendors, parks, community centers, etc.

His colleague Jennifer offers to help him use FME Server to let users access the data directly without his help, creating self-serve data delivery.

FME Server can automatically run workspaces on a schedule or in response to a trigger such as an email being received. It also lets you manage data securely and make data accessible to everyone in your organization.

FME Server has five default security roles. You will need at least the fmeauthor role to follow along in this module.

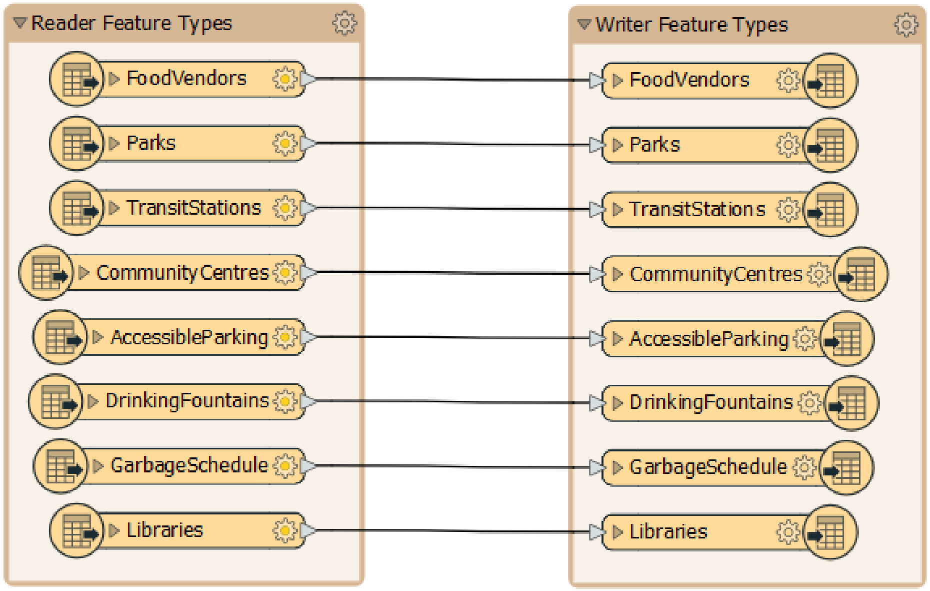
Learn More

Give Users Control Using Published Parameters

Jennifer sends Frank a workspace to help him get started. He downloads and opens it with FME Workbench.

Follow along with Frank's steps using your own version of FME.

It's a simple workspace that reads in all the feature classes of an Esri geodatabase and then writes them out to the format of a user's choice.



Jennifer tells him he can create a workspace that gives end-users control over what data they get. She suggests two steps to let users customize the data they receive:

1. Let them choose the **format**
2. Let them choose the **layers**

Jennifer's advice raises an important point. Remember to design your workspaces based on the FME platform user roles. Frank has to design his workspace so it works well for end-users. These end-users might be running the workspace as a logged-in user of FME Server or might be using a public-facing FME Server App. The choices he provides need to make sense for these situations.

The first step is to let users specify which data format they want by providing them with a fixed selection of format choices. Since the workspace uses a Generic writer, users are currently able to choose any format. This unlimited choice can lead to problems:

- Users may become overwhelmed with options.
- Users may choose a nonsensical format, e.g. trying to write text data to an image/raster format like JPEG, which can create errors that cause the workspace to fail.

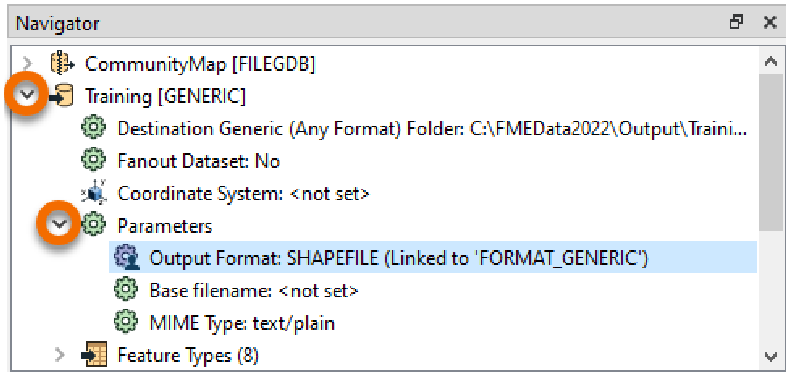
Jennifer tells Frank he can create user parameters to address this issue. User parameters give users control over how a workspace runs. You can use published parameters when running workspaces locally, but it is critical to use them when creating workspaces for FME Server, which are likely to be run by other people.

User parameters can be published or private.

- **Published parameters** are shown to the user when they run the workspace.
  Private parameters are used for setting a value once and using it in many places in a workspace. These are not shown to other users.

Define Which Formats Users Can Write To

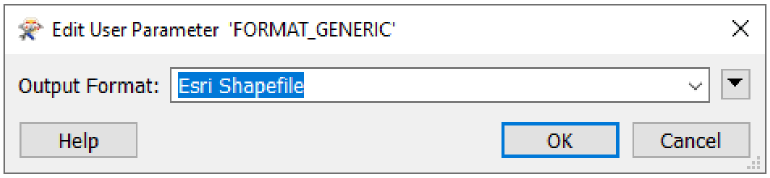
To restrict the formats available for writing, Frank finds the **Training [GENERIC]** writer in the Navigator. He clicks the drop-down arrow and expands the **Parameters** section.



Navigator

- CommunityMap [FILEGDB]
- Training [GENERIC]
  - Destination Generic (Any Format) Folder: C:\FMEData2022\Output\Traini...
  - Fanout Dataset: No
  - Coordinate System: <not set>
  - Parameters
    - Output Format: SHAPEFILE (Linked to 'FORMAT_GENERIC')
    - Base filename: <not set>
    - MIME Type: text/plain
  - Feature Types (8)

He double-clicks the **Output Format** parameter and sees it’s currently assigned to “Esri Shapefile”.

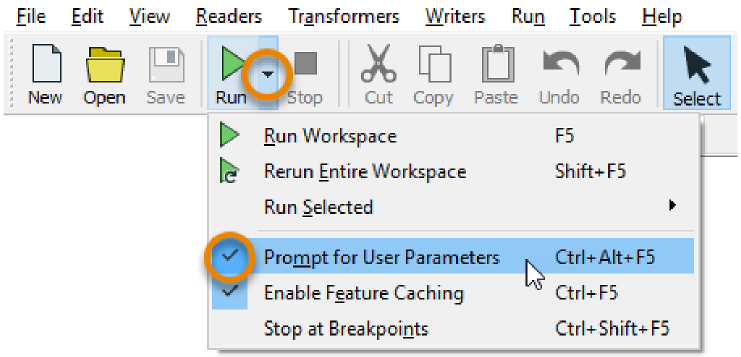


Edit User Parameter 'FORMAT_GENERIC'

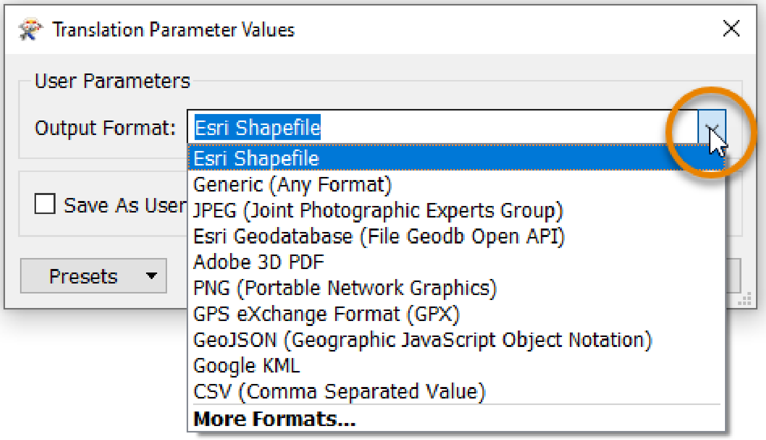
Output Format: Esri Shapefile

Back in the **Navigator** window, Frank sees that the output format is linked to a user parameter because it has a different icon and says “(Linked to ‘FORMAT_GENERIC’)”. Most parameters in a workspace – reader, writer, and transformer – can be linked to published parameters to let users set their value when they run the workspace.

Frank wants to see how the parameter works, so he clicks the **Run menu** and ensures **Prompt for User Parameters** is checked, and then clicks **Run**.


File
Edit
View
Readers
Transformers
Writers
Run
Tools
Help
New
Open
Save
Run
Stop
Cut
Copy
Paste
Undo
Redo
Select
Run Workspace F5
Rerun Entire Workspace Shift+F5
Run Selected
Prompt for User Parameters Ctrl+Alt+F5
Enable Feature Caching Ctrl+F5
Stop at Breakpoints Ctrl+Shift+F5

The **Translation Parameter Values** dialog pops up and asks the user to select a value for **Output Format**, defaulting to "Esri Shapefile". Frank clicks the drop-down menu for **Output Format** and sees he can pick any format supported by FME. That's too many options.

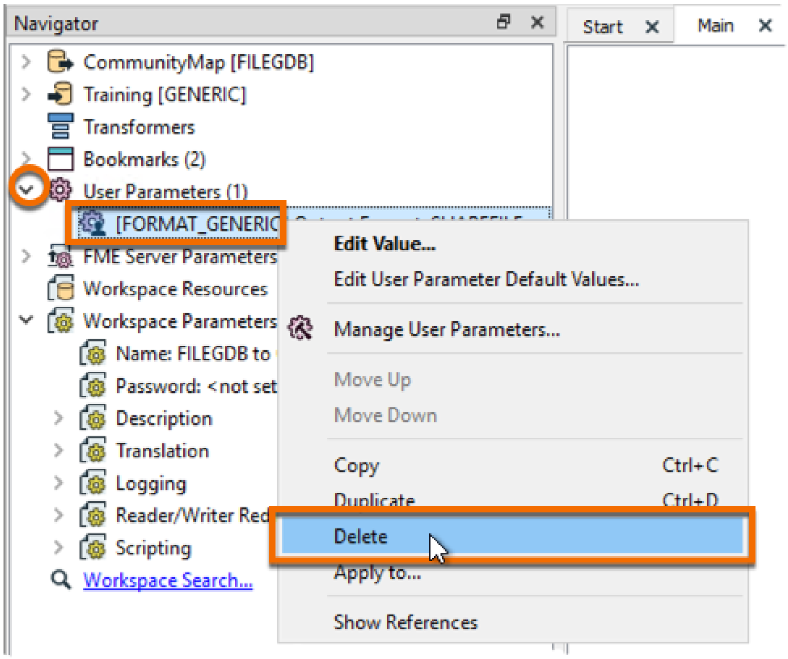

Translation Parameter Values

User Parameters

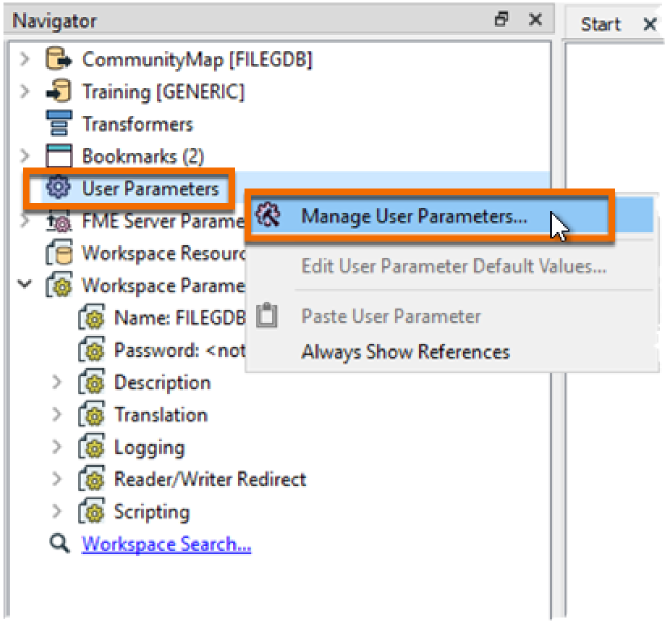
Output Format: Esri Shapefile

Esri Shapefile
Generic (Any Format)
JPEG (Joint Photographic Experts Group)
Esri Geodatabase (File Geodb Open API)
Adobe 3D PDF
PNG (Portable Network Graphics)
GPS eXchange Format (GPX)
GeoJSON (Geographic JavaScript Object Notation)
Google KML
CSV (Comma Separated Value)
More Formats...
□ Save As User
Presets

To fix this problem, Frank closes the dialog and looks at the **User Parameters section of the Navigator.** He finds the "[FORMAT_GENERIC]" published parameter. He will replace this to restrict the user's options. He right-clicks it and selects **Delete**.

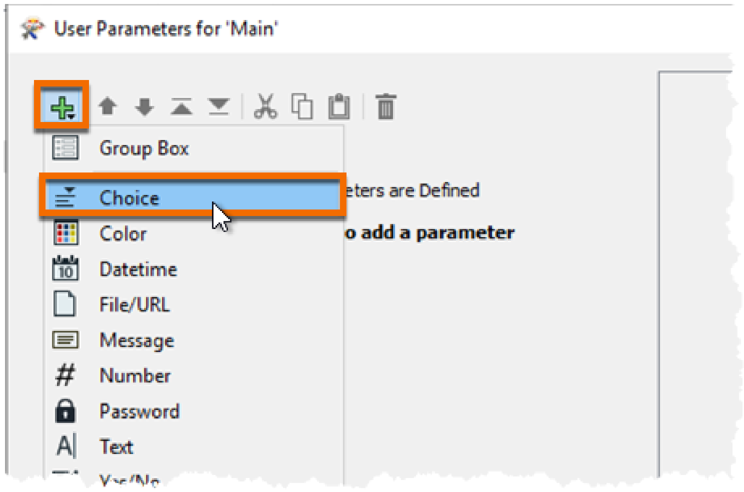


Then he right-clicks **User Parameters** and clicks **Manage User Parameters...**



Frank is deleting the existing user parameter and adding a new one because some user parameters that are created automatically cannot be edited fully. For those, it's better to start fresh.

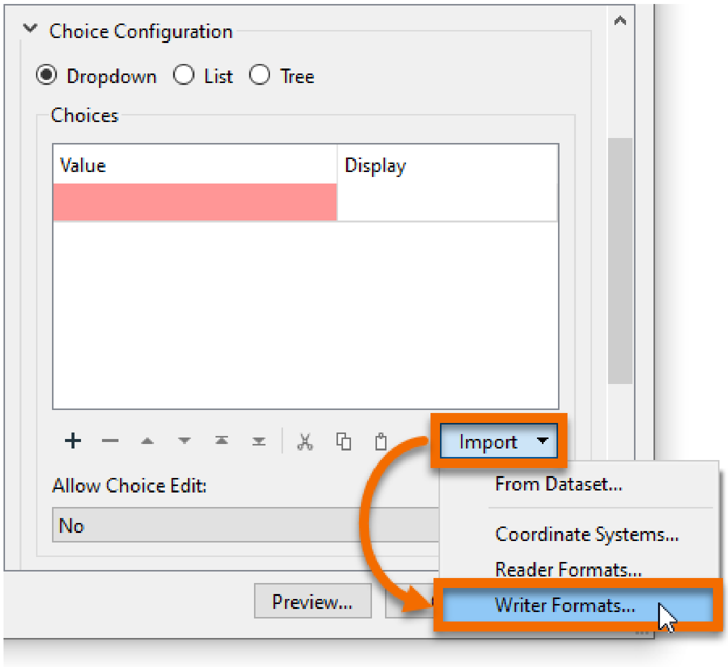
The **Parameter Manager** dialog opens. There are many types of user parameters available. In this case, Frank chooses **Choice**, which lets him provide a list of options to the user that map onto a different value provided to the workspace. This user parameter is useful for letting users choose formats or coordinate systems because it hides the more complex name FME needs and instead shows a simple version to the user.



On the right-hand side Frank fills out the dialog with the following parameter properties:

Parameter Identifier	OutputFormat
Prompt	Enter an output format
Published	Enabled
Required	Disabled
Disable Attribute Assignment	Enabled
Choice Configuration	Drop-down

To fill in the **Choices** table, Frank wants to import the file formats the user can choose from. He clicks **Import** > **Writer Formats**.



Choice Configuration

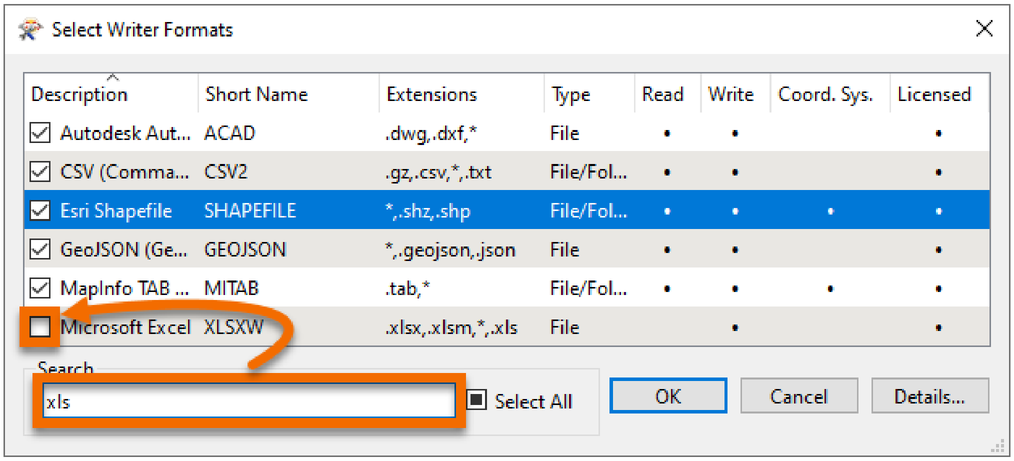
● Dropdown ○ List ○ Tree

Choices

Value	Display

Allow Choice Edit: No

In the **Select Writer Formats** dialog, use the Search bar to search for the following formats. Click the check to add them to the list.



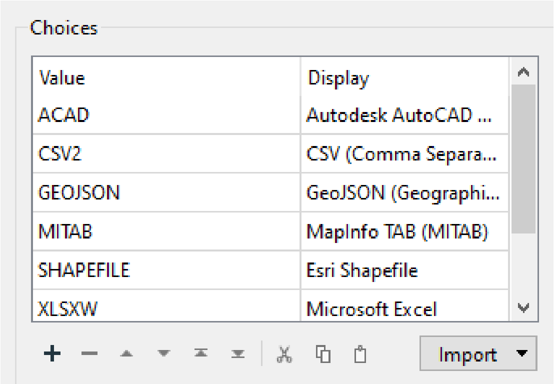
Select Writer Formats

Description	Short Name	Extensions	Type	Read	Write	Coord. Sys.	Licensed
☑ Autodesk Aut...	ACAD	.dwg,.dxf,*	File	•	•		•
☑ CSV (Comma...	CSV2	.gz,.csv,*,.txt	File/Fol...	•	•		•
☑ Esri Shapefile	SHAPEFILE	*,.shz,.shp	File/Fol...	•	•	•	•
☑ GeoJSON (Ge...	GEOJSON	*,.geojson,.json	File	•	•		•
☑ MapInfo TAB ...	MITAB	.tab,*	File/Fol...	•	•	•	•
□ Microsoft Excel	XLSXW	.xlsx,.xlsm,*,.xls	File		•		•

Search: xls

Click **OK**.

The selected formats appear in the **Choices** table:

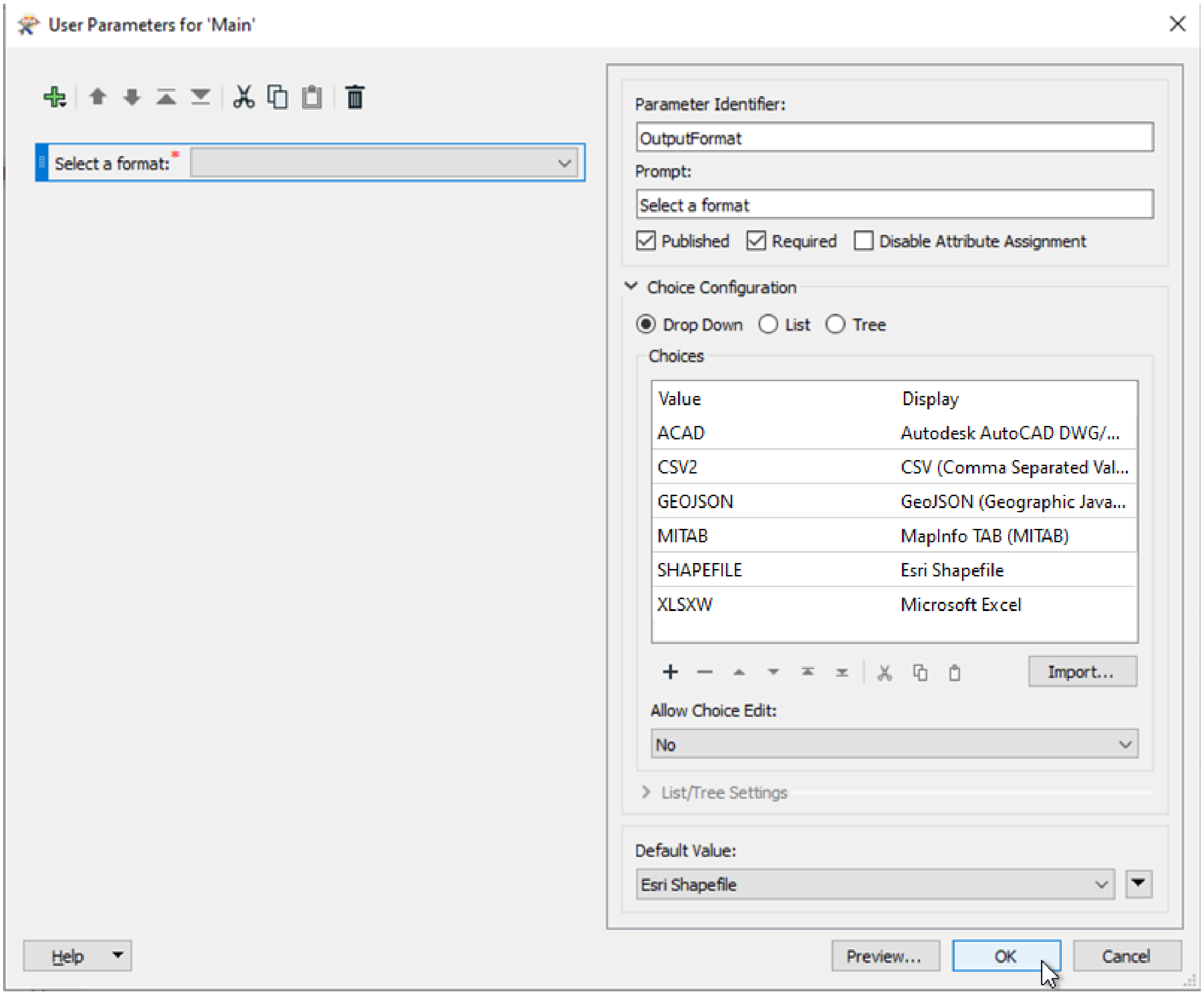


Choices

Value	Display
ACAD	Autodesk AutoCAD ...
CSV2	CSV (Comma Separa...
GEOJSON	GeoJSON (Geographi...
MITAB	MapInfo TAB (MITAB)
SHAPEFILE	Esri Shapefile
XLSXW	Microsoft Excel

Frank uses a table here because he wants to show the user a more readable name for the format (**Description**) instead of the value that FME needs to choose the format (**Short Name**).

Finally, Frank selects Esri Shapefile as the default value.



User Parameters for 'Main'

Select a format:*

Parameter Identifier: OutputFormat

Prompt: Select a format

☑ Published ☑ Required □ Disable Attribute Assignment

Choice Configuration

● Drop Down ○ List ○ Tree

Choices

Value	Display
ACAD	Autodesk AutoCAD DWG/...
CSV2	CSV (Comma Separated Val...
GEOJSON	GeoJSON (Geographic Java...
MITAB	MapInfo TAB (MITAB)
SHAPEFILE	Esri Shapefile
XLSXW	Microsoft Excel

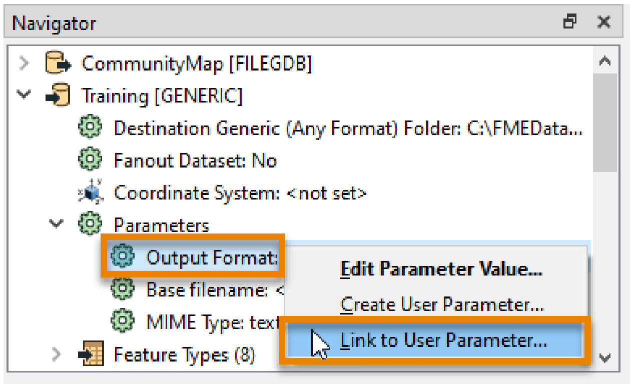
Allow Choice Edit: No

List/Tree Settings

Default Value: Esri Shapefile

He clicks **OK** to create the published parameter.

Now that he has created a published parameter, he needs to link it to the correct writer parameter for it to work. He goes back to the writer's **Output Format** parameter, right-clicks it, and clicks **Link to User Parameter**.



Navigator

CommunityMap [FILEGDB]

Training [GENERIC]

Destination Generic (Any Format) Folder: C:\FMEData...

Fanout Dataset: No

Coordinate System: <not set>

Parameters

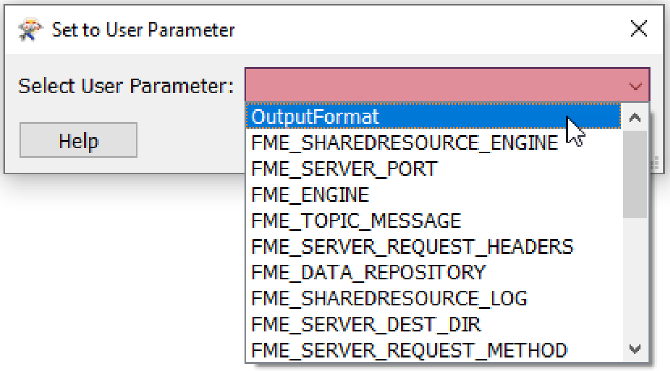
Output Format:

Base filename: <

MIME Type: text

Feature Types (8)

In the **Set to User Parameter** dialog, he picks **OutputFormat** from the list and clicks OK.

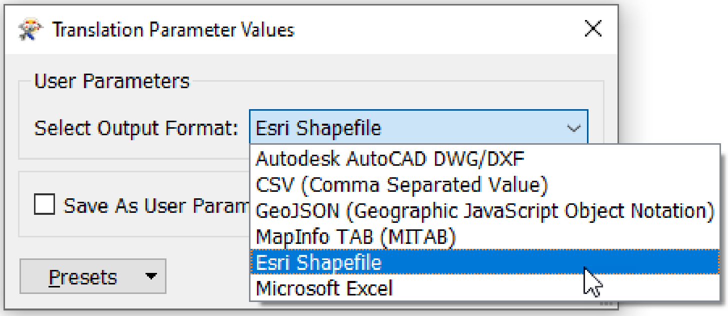


Set to User Parameter

Select User Parameter:

OutputFormat
FME_SHAREDRESOURCE_ENGINE
FME_SERVER_PORT
FME_ENGINE
FME_TOPIC_MESSAGE
FME_SERVER_REQUEST_HEADERS
FME_DATA_REPOSITORY
FME_SHAREDRESOURCE_LOG
FME_SERVER_DEST_DIR
FME_SERVER_REQUEST_METHOD

Now when he clicks **Run**, he is prompted to choose an output format from the restricted list. The workspace runs and he receives the data in the format he chose.



Translation Parameter Values

User Parameters

Select Output Format: Esri Shapefile

Autodesk AutoCAD DWG/DXF
CSV (Comma Separated Value)
GeoJSON (Geographic JavaScript Object Notation)
MapInfo TAB (MITAB)
Esri Shapefile
Microsoft Excel

□ Save As User Param

Exercise

Frank is happy with the format selection, but now he'd like to let the user pick the layers from the geodatabase they receive. He knows he can control this with the user parameter **Feature Types to Read**, found in the **Navigator** under **CommunityMap [FILEGDB]** reader > **Parameters** > **Features to Read**.

For the exercise, create a new published parameter that lets the user pick which feature types to read and write. **Hint**: in some cases, creating a published parameter is as easy as right-clicking the parameter you want to link and selecting **Create User Parameter**.

1FME Server can run workspaces on a schedule.

○ A.True
○ B.False

2User parameters let FME users:

○ A.Control how workspaces run
○ B.Log in so the run event is tied to their user profile
○ C.Control how fast their workspace runs
○ D.Control who can run their workspace through Windows Active Directory

3User parameters that are set by the user at run-time are called:

○ A.Global parameters
○ B.Local parameters
○ C.Published parameters
○ D.Private parameters

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Publish a Self-Serve Workspace to the Web

Learning Objectives

After completing this unit, you'll be able to:

- Connect FME Desktop to FME Server using a web connection.
- Publish a workspace to the web.
- View your published workspace in FME Server.

Resources

- Starting workspace

Video

Publish a Workspace to the Web

Now that Frank has created a self-serve workspace, he needs to publish it to FME Server.

The rest of this module assumes you have access to an FME Server. If you need help installing, licensing, and creating an account for FME Server, please follow these instructions. Options for accessing FME Server include:

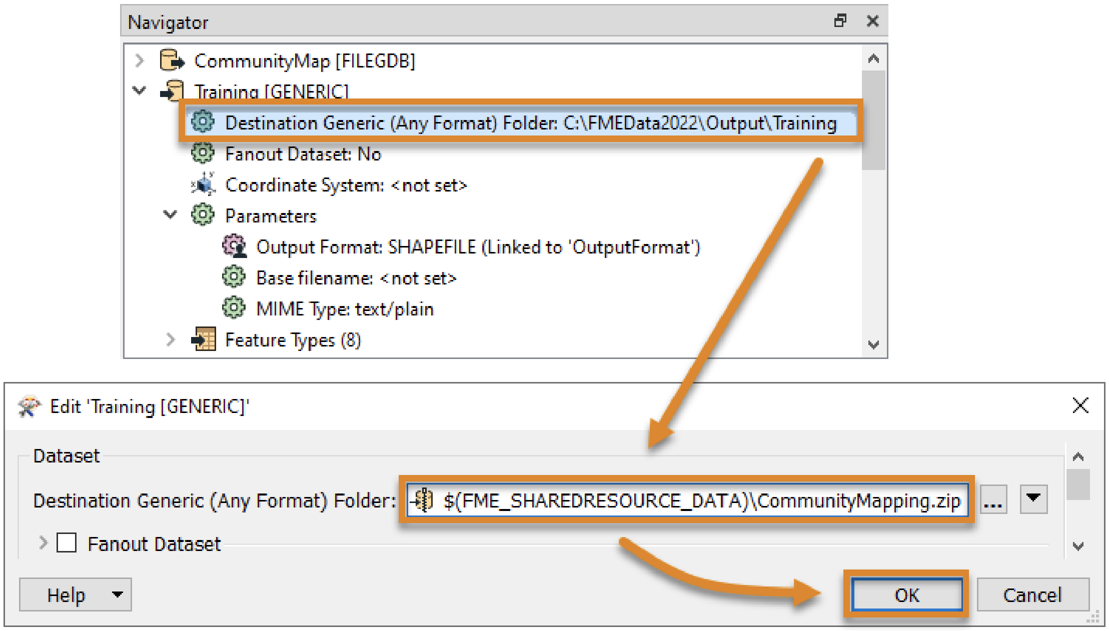
- Install FME Server on your own machine (a free trial is available)
- Access your organization's Server
- Use an instance provided by requesting an on-demand virtual machine in the footer link below

Preparing a Workspace to Publish to FME Server

Frank already has a workspace that is working well. However, Jennifer notices a problem that he needs to address before publishing to FME Server. She explains that creating the correct user parameters before publishing makes a much better experience for users running the workspace using FME Server.

The path for the output data in the workspace is currently hardcoded to "C:\FMEData2022\Output\Training\Training.xlsx". Frank confirms this in the **Navigator** > **Training [Generic]** writer > **Destination Generic (Any Format) Folder** parameter. Having this hard coded is fine when he just runs it on his own machine. When he publishes a workspace to FME Server, the workspace will try to write data to the C:\ drive of the machine that is hosting FME Server. However, he can't assume access to a C:\ drive: it could be restricted for security purposes, or if it's a Linux machine it won't even exist! Jennifer informs him it's best practice to instead read and write data that exists in FME Server's shared **Resources** folder.

To fix this, Frank double-clicks on the **Destination Generic (Any Format) Folder** parameter and enters "$(FME_SHAREDRESOURCE_DATA)\CommunityMapping.zip". This path includes the "FME_SHAREDRESOURCE_DATA" user parameter (viewable under **User Parameters** > **FME Server Parameters**) and sets the workspace to write out its results to a ZIP file called CommunityMapping. When the workspace is published to FME Server, the data will be written out to the shared **Data** folder for all users to access. We'll take advantage of that feature in a later unit.



Navigator

- CommunityMap [FILEGDB]
- Training [GENERIC]
  - Destination Generic (Any Format) Folder: C:\FMEData2022\Output\Training
  - Fanout Dataset: No
  - Coordinate System: <not set>
  - Parameters
    - Output Format: SHAPEFILE (Linked to 'OutputFormat')
    - Base filename: <not set>
    - MIME Type: text/plain
  - Feature Types (8)

Edit 'Training [GENERIC]'

Dataset

Destination Generic (Any Format) Folder: $(FME_SHAREDRESOURCE_DATA)\CommunityMapping.zip

□ Fanout Dataset

FME has many useful built-in parameters similar to **FME_SHAREDRESOURCE_DATA** that let you access information about FME itself, such as the FME build number used to run the workspace or the location of the temporary data folder on the machine running the workspace. These parameters can be used for customizing how FME runs in different environments or for creating customized reporting on workspace runs.

Learn More About FME Parameters

Learn More About FME Server Parameters

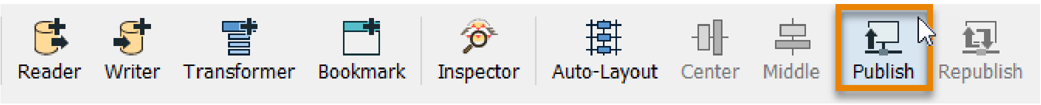
All file- and folder-based FME formats can be read as or written to ZIP files. Simply provide a ZIP file for the **Dataset** parameter of your reader or add ".zip" into the **Dataset** file path of your writer.

Connect to FME Server with a Web Connection

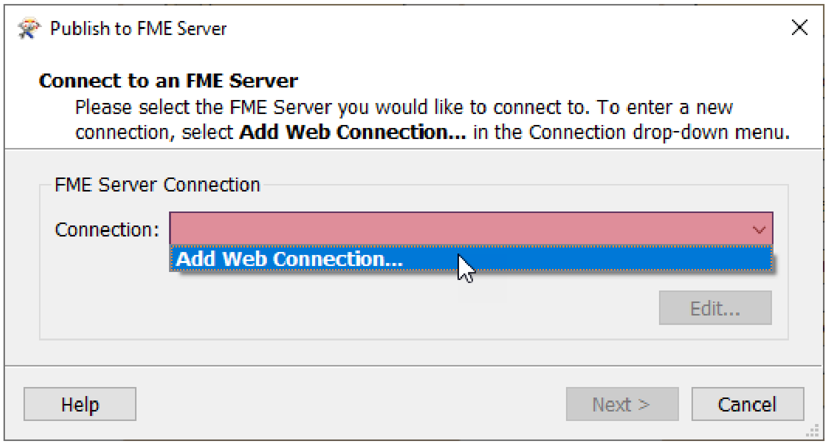
Now that Frank's prepared the workspace by creating the correct user parameters, he needs to create an FME Server web connection before he can publish. Jennifer walks him through the steps.

Remember that database and web connections save authentication information to connect to databases, web services, and APIs. They are stored on the user's operating system profile, so authentication information can be stored separately from the workspace. They can also be published to FME Server to allow multiple users to share them without exposing any passwords. The FME Server web connection works the same way as other web connections.

To create an FME Server web connection, Frank clicks the **Publish to FME Server** button in the Workbench toolbar.



Because this is his first time connecting to FME Server, the **Connection** drop-down is red. He clicks the drop-down and chooses **Add Web Connection...**



Publish to FME Server

Connect to an FME Server

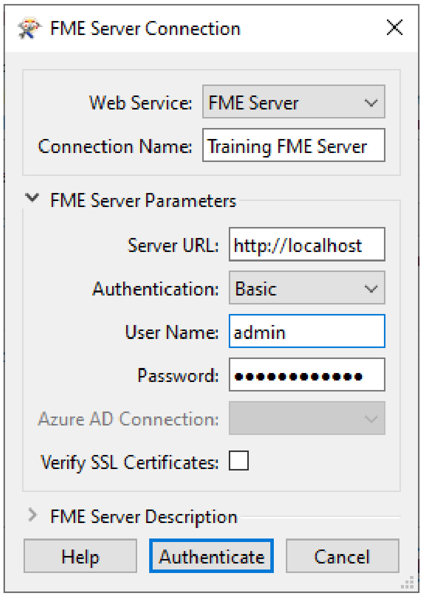
Please select the FME Server you would like to connect to. To enter a new connection, select **Add Web Connection...** in the Connection drop-down menu.

FME Server Connection

Connection:

You can manage Database and Web Connections under Tools > FME Options from the Workbench menu bar if you are running a Windows machine, or FME Workbench > Preferences if you are running a Mac. You can also create them where they are needed, such as in a reader or transformer dialog.

The **FME Server Connection** dialog opens and he fills it out to connect to FME Server running on his own machine:



FME Server Connection

Web Service: FME Server

Connection Name: Training FME Server

FME Server Parameters

Server URL: http://localhost

Authentication: Basic

User Name: admin

Password: ●●●●●●●●●●●●

Azure AD Connection:

Verify SSL Certificates: □

FME Server Description

If you are attending a Safe Software training course, the password is FMElearnings.

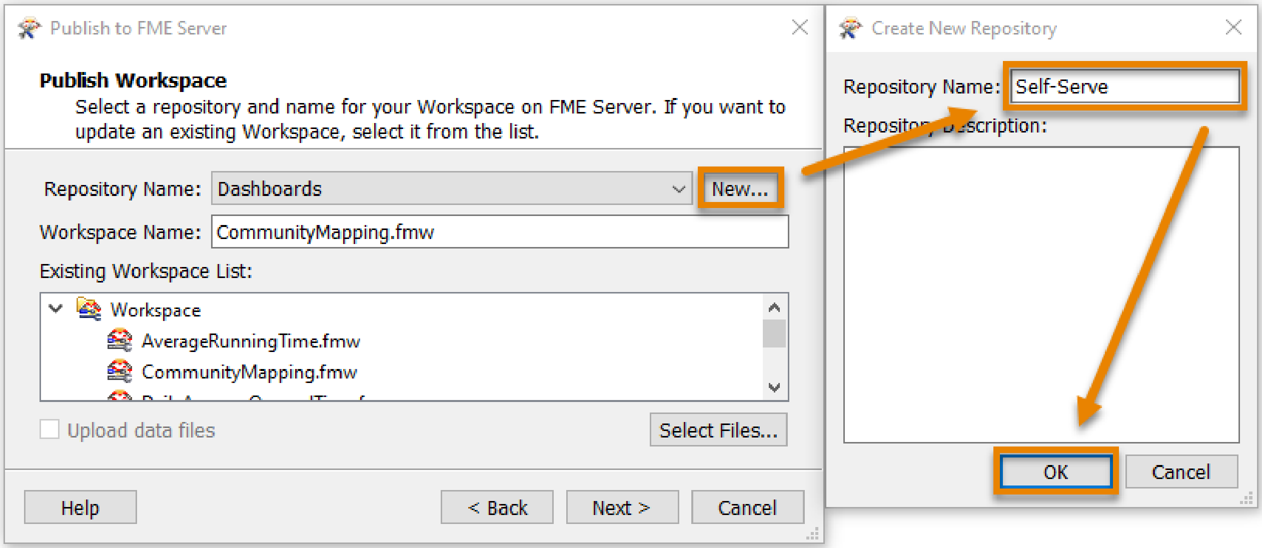
You can connect to your own organization’s FME Server if you have permission. Get the connection details from your FME Server Administrator. If you are using FME Server on your own machine, use http://localhost and your own username and password set during installation.

After filling out the dialog, Frank clicks **Authenticate**. FME Workbench tests the connection and then returns him to the **Publish to FME Server** dialog. He clicks **Next**.

Publish a Workspace to FME Server

Now that he’s set up a web connection, Frank can finish publishing his workspace. He renames it **CommunityMapping.fmw** by typing in the **Workspace Name** field.

Workspaces in FME Server are stored in repositories. To create a new repository for his workspace, Frank clicks **New...** and makes a new repository called “Self-Serve”. He clicks **OK** twice to finish creating his new repository.



Publish to FME Server

Publish Workspace

Select a repository and name for your Workspace on FME Server. If you want to update an existing Workspace, select it from the list.

Repository Name: Dashboards

Workspace Name: CommunityMapping.fmw

Existing Workspace List:

- Workspace
  - AverageRunningTime.fmw
  - CommunityMapping.fmw

□ Upload data files

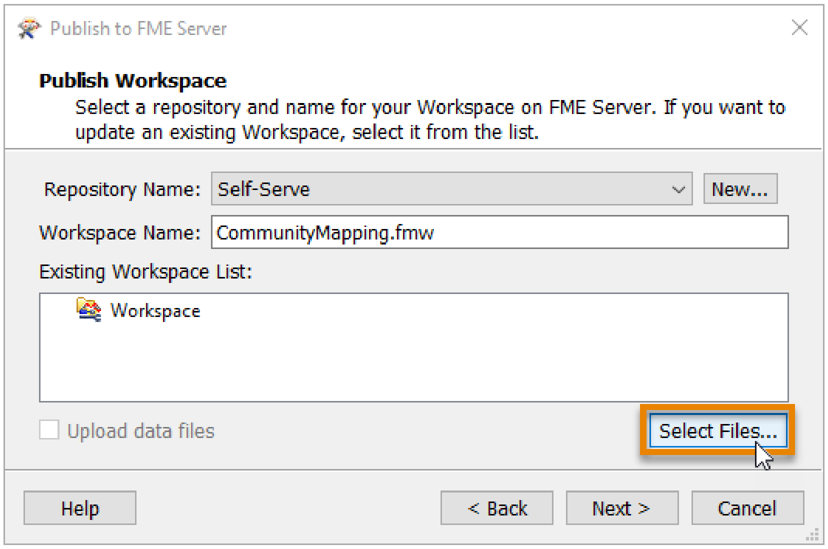
Create New Repository

Repository Name: Self-Serve

Repository Description:

Jennifer reminds Frank to consider how his workspace will access data after he publishes it to FME Server. Many workspaces use data stored locally on the author’s computer, such as C:\FMEData2022\Data. However, FME Server is usually run on a different machine that doesn’t have access to that directory. Therefore, data used in workspaces published to FME Server should also be added to FME Server.

To accomplish this, Frank clicks the **Select Files...** button.



Publish to FME Server

Publish Workspace

Select a repository and name for your Workspace on FME Server. If you want to update an existing Workspace, select it from the list.

Repository Name: Self-Serve

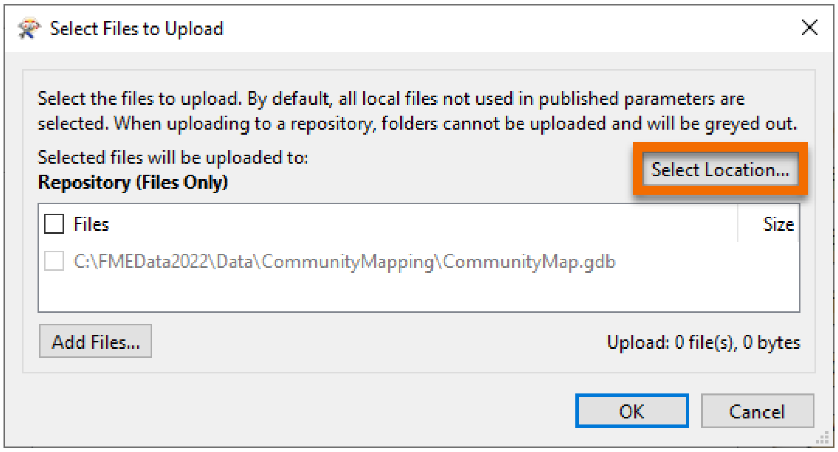
Workspace Name: CommunityMapping.fmw

Existing Workspace List:

- Workspace

□ Upload data files

This action opens the **Select Files to Upload** dialog, which Frank sees he can use to add the geodatabase to the shared **Data** folder in FME Server. Frank does this by clicking the **Select Location...** button, then choosing the **Upload to a shared resource folder** option. Finally, he clicks the **Data** folder to select it and then presses **OK**.



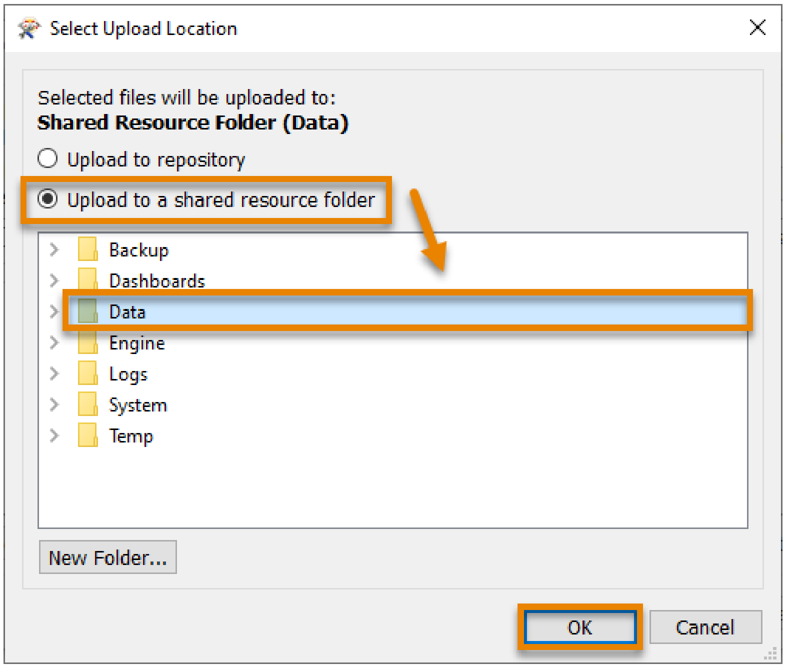
Select Files to Upload

Select the files to upload. By default, all local files not used in published parameters are selected. When uploading to a repository, folders cannot be uploaded and will be greyed out.

Selected files will be uploaded to:
Repository (Files Only)

☐ Files	Size
☐ C:\FMEData2022\Data\CommunityMapping\CommunityMap.gdb	

Upload: 0 file(s), 0 bytes



Select Upload Location

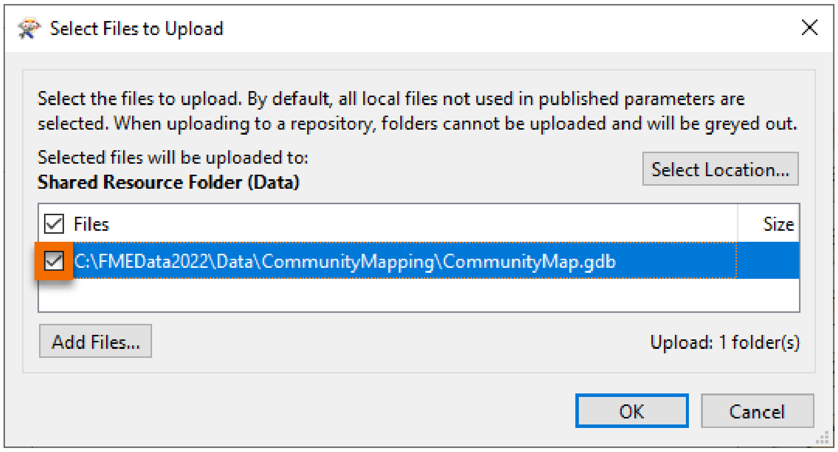
Selected files will be uploaded to:
Shared Resource Folder (Data)

○ Upload to repository

● Upload to a shared resource folder

- Backup
- Dashboards
- Data
- Engine
- Logs
- System
- Temp

He's indicated where he wants to upload the data but now has to confirm which datasets to upload. He does this by checking the geodatabase.



Select Files to Upload

Select the files to upload. By default, all local files not used in published parameters are selected. When uploading to a repository, folders cannot be uploaded and will be greyed out.

Selected files will be uploaded to:
Shared Resource Folder (Data)

☑ Files	Size
☑ C:\FMEData2022\Data\CommunityMapping\CommunityMap.gdb	

Upload: 1 folder(s)

With a destination for the geodatabase on FME Server selected, he clicks **OK** and then **Next**.

Frank could have published his data directly to the Self-Serve repository, but that comes with a number of challenges. See the Manage FME Server Data and Connections module for more information.

Before he can publish, Frank has to select service(s) for his workspace to use.

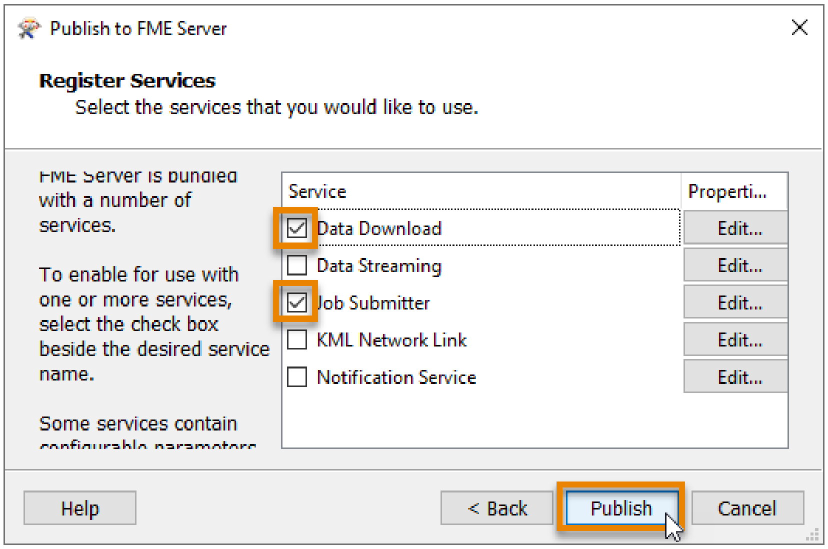
An **FME Server service** is an interface for interacting with FME Server.

Learn More

Frank has to choose services to control how users receive the data when they run the workspace on FME Server. These services each provide a different way to interact with the workspace:

- The **Data Download service** provides a link to the output as a downloadable ZIP file.
- The **Data Streaming service** either immediately downloads the output, or displays it in the browser if possible (e.g. HTML).
- The **Job Submitter service** accepts and runs workspace job requests. Data is written, but not presented to the user.
- The **Notification service** lets the workspace interact with FME Server Automations.

Frank makes sure both the **Data Download** and **Job Submitter** services are checked. With these checked, anyone can run the workspace and receive a ZIP file of the results. Then he clicks **Publish**.



Publish to FME Server

Register Services

Select the services that you would like to use.

FME Server is bundled with a number of services.

To enable for use with one or more services, select the check box beside the desired service name.

Some services contain configurable parameters

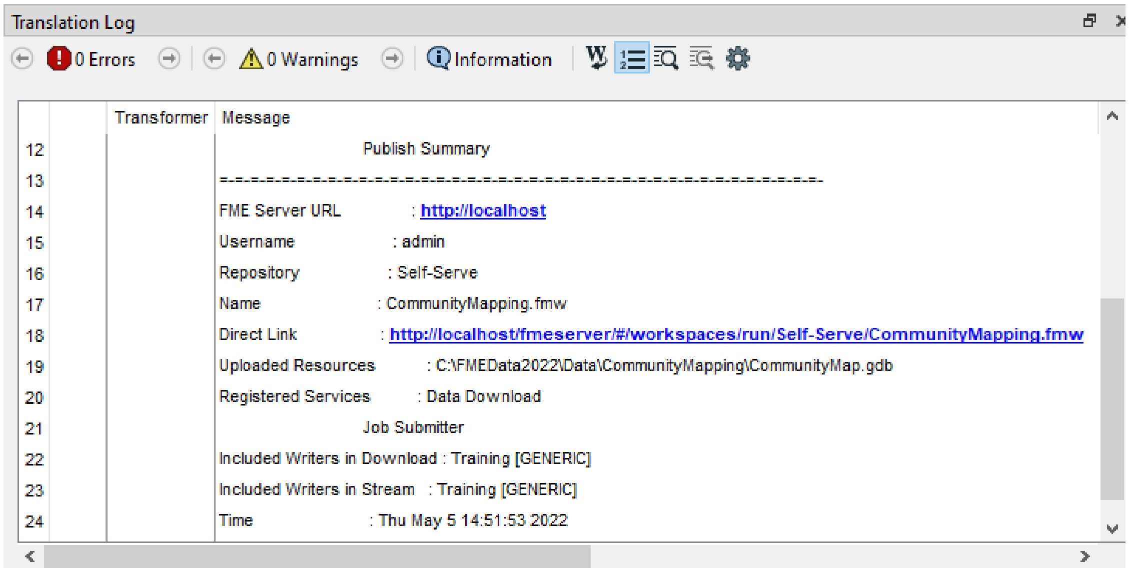
Service	Properti...
☑ Data Download	Edit...
□ Data Streaming	Edit...
☑ Job Submitter	Edit...
□ KML Network Link	Edit...
□ Notification Service	Edit...

Frank's workspace and data are now available in FME Server.

Most of these actions (uploading data, controlling services, and running workspaces) can also be controlled using the FME Server REST API. The one exception is publishing workspaces, which must be done through FME Workbench.

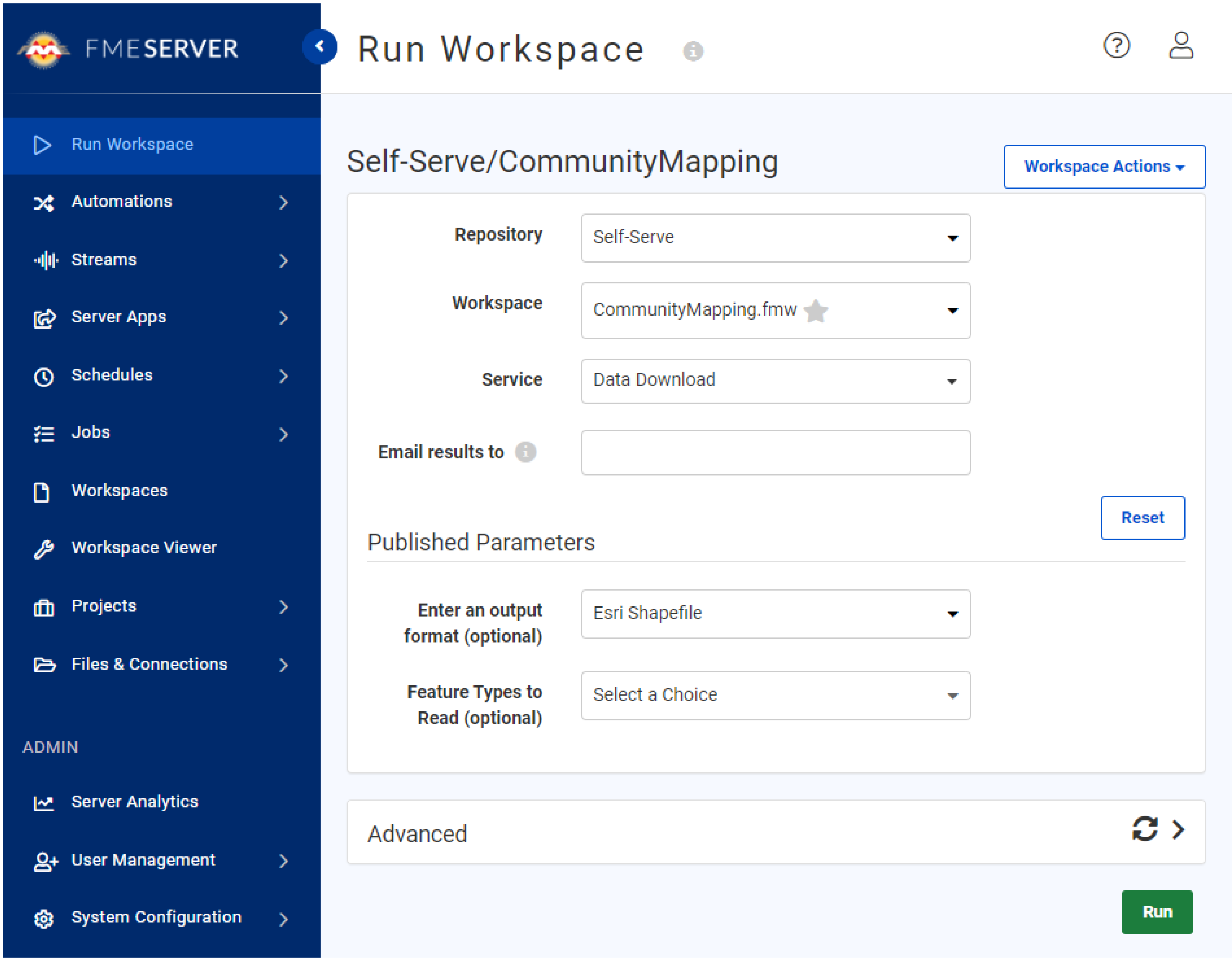
View a Published Workspace

Frank can access his workspace in a web browser using the **Direct Link** provided in the **Translation Log**: http://localhost/fmeserver/#/workspaces/run/Self-Serve/CommunityMapping.fmw.



```
                         Publish Summary
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
FME Server URL          : http://localhost
Username                : admin
Repository              : Self-Serve
Name                    : CommunityMapping.fmw
Direct Link             : http://localhost/fmeserver/#/workspaces/run/Self-Serve/CommunityMapping.fmw
Uploaded Resources      : C:\FMEData2022\Data\CommunityMapping\CommunityMap.gdb
Registered Services     : Data Download
                          Job Submitter
Included Writers in Download : Training [GENERIC]
Included Writers in Stream   : Training [GENERIC]
Time                    : Thu May 5 14:51:53 2022
```

He clicks the link to open FME Server and logs in, if required. This link directs Frank to the **Run Workspace** page. Here he can choose which workspace to run, which service to use, optionally provide an email to send the results to, and set the published parameters. Here he sees why published parameters are important - creating them in FME Workbench allows him to customize the experience for users running his workspace on the web.



Run Workspace

Self-Serve/CommunityMapping

Repository: Self-Serve

Workspace: CommunityMapping.fmw

Service: Data Download

Email results to:

Published Parameters

Enter an output format (optional): Esri Shapefile

Feature Types to Read (optional): Select a Choice

Exercise

Make sure you have followed along with Frank’s steps.

1Web connections let you share authentication information (logins, passwords, API keys, etc.) with other users of FME Server without exposing your plaintext password.

○ A.True
○ B.False

2Which of the following is not a valid way to access your workspace in FME Server after publishing it?

○ A.From FME Workbench, click Run > Run on FME Server.
○ B.Login to FME Server and click the link to the Run workspace page on the left sidebar.
○ C. Click the Direct Link URL in the Translation Log after publishing the workspace from FME Workbench.

3The Run Workspace page lets users set Published Parameters that control how the workspace will run.

○ A.True
○ B.False

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Run a Workspace on the Web

Learning Objectives

After completing this unit, you'll be able to:

- Run a workspace from FME Server.
- Describe how you can run workspaces on the go with the FME Data Express mobile app.

Resources

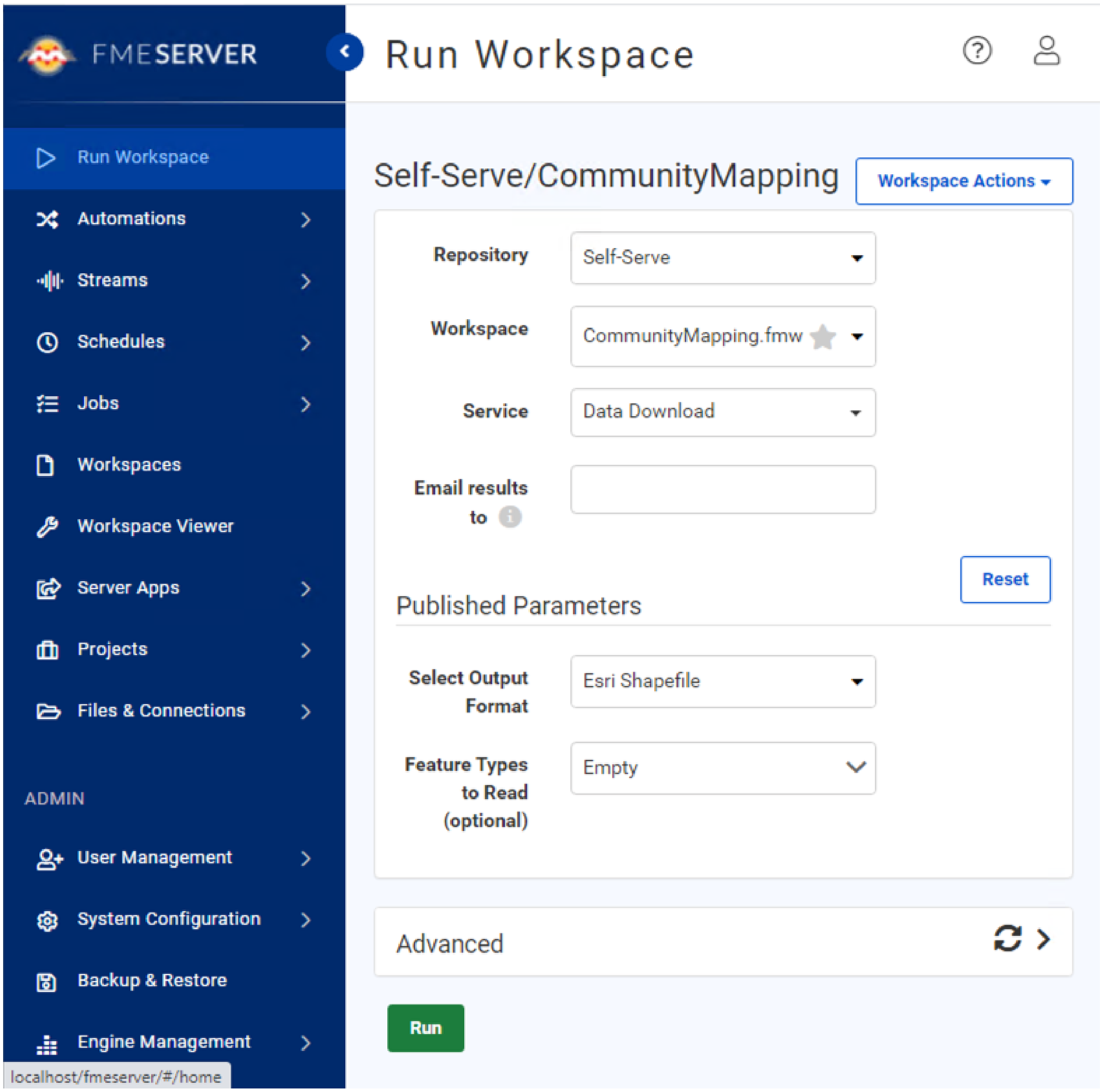
- Interactive tutorial to help you follow along
- Starting workspace (should already be published to FME Server as per the previous unit's instructions)

Video

Running a Workspace on the Web

Frank is ready to test his self-serve workspace. He clicks the link from the **Translation Log** (http://localhost/fmeserver/#/workspaces/run/Self-Serve/CommunityMapping.fmw/) to open the workspace in FME Server (2022.0 or later).

He logs into FME Server and sees the **Run Workspace** page.



Run Workspace

Self-Serve/CommunityMapping

Repository: Self-Serve

Workspace: CommunityMapping.fmw

Service: Data Download

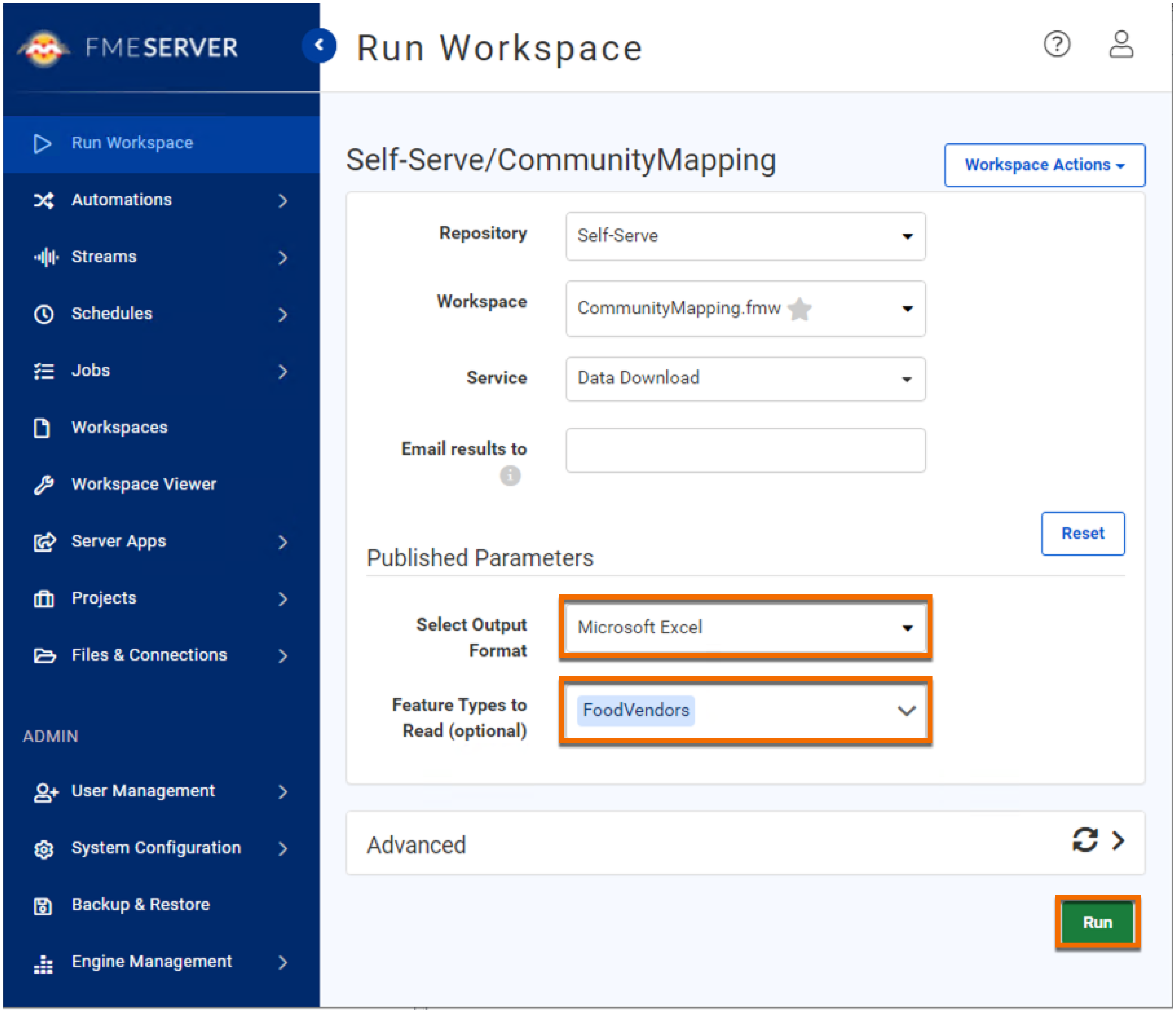
Email results to:

Published Parameters

Select Output Format: Esri Shapefile

Feature Types to Read (optional): Empty

He leaves the default settings at the top part of the interface. He selects "Excel" from the **Output Format** drop-down. He wants a current extract of the **FoodVendors** table, so he looks at the **Feature Types to Read** Published Parameter. He clicks the drop-down and selects “FoodVendors” from the list. Then he clicks **Run** to request his data.



Self-Serve/CommunityMapping

Repository: Self-Serve

Workspace: CommunityMapping.fmw

Service: Data Download

Email results to:

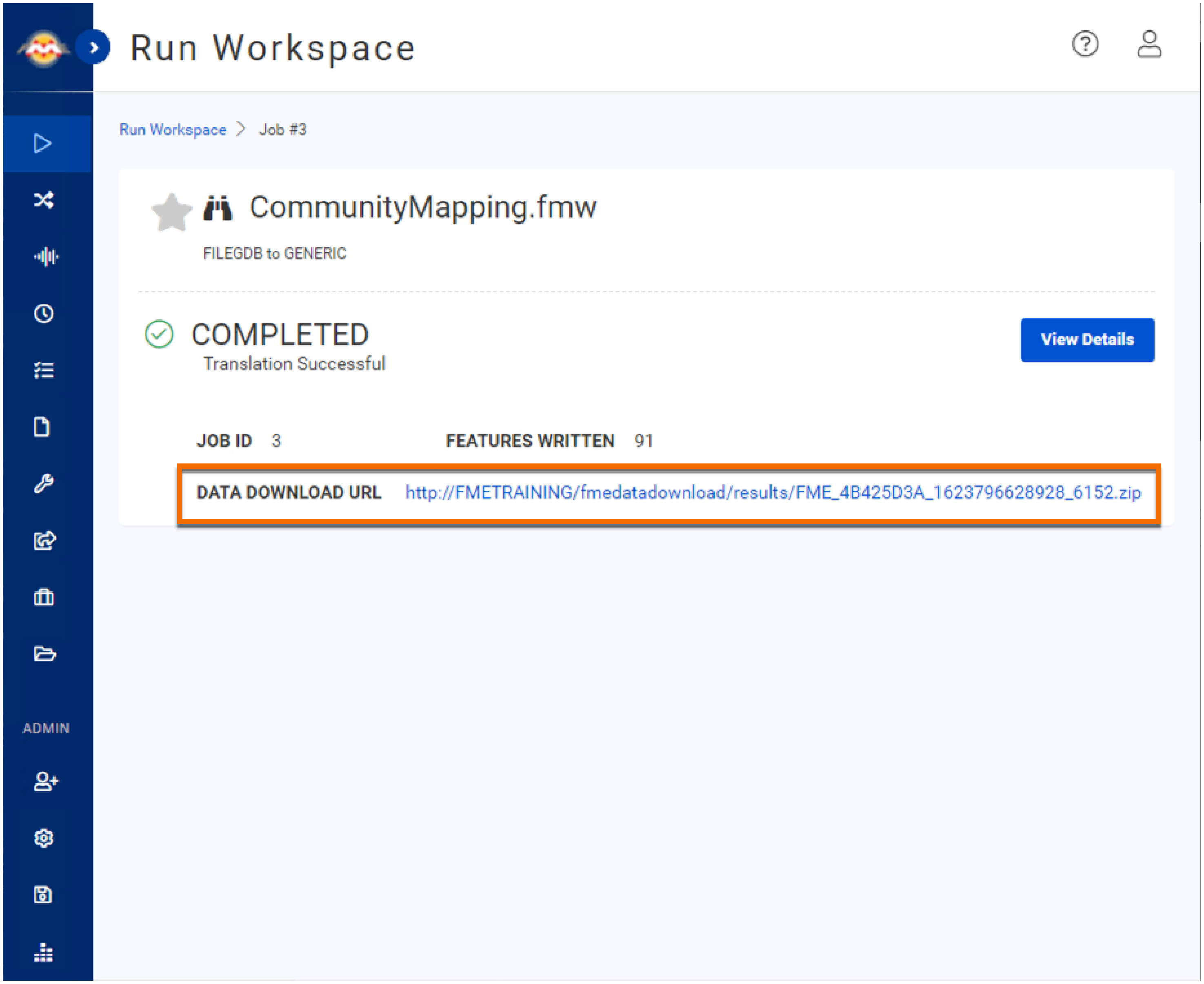
Published Parameters

Select Output Format: Microsoft Excel

Feature Types to Read (optional): FoodVendors

Receiving the Results of a Data Download

Because he used the **Data Download** service, when the workspace finishes running, he is taken to a page with a link to a ZIP file of the results of the workspace. He clicks it to download.



CommunityMapping.fmw

FILEGDB to GENERIC

COMPLETED
Translation Successful

JOB ID 3 FEATURES WRITTEN 91

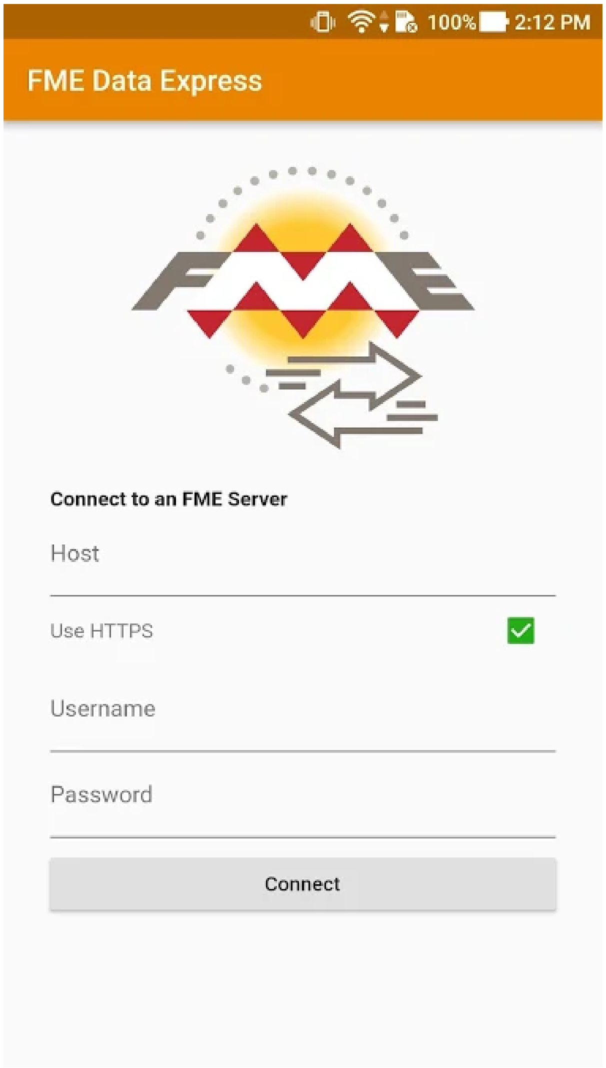
DATA DOWNLOAD URL http://FMETRAINING/fmedatadownload/results/FME_4B425D3A_1623796628928_6152.zip

He finds the ZIP file in his local **Downloads** folder and extracts it. He has received the **FoodVendors** data as an Excel file called **Training.xlsx**. Now he can send the self-serve link to any end-user and they can download their desired data.

Mobile

Fatima is one of Jennifer and Frank’s colleagues. She works as a Business Data Analyst at the city’s Business License Office. She often needs to add data to the **FoodVendors** feature class when she is out of the office doing business license inspections. Frank sets up FME Data Express on her tablet (available on Android or iOS) so she is able to access FME Server on-the-go. Using FME Data Express, she can run a different workspace designed for data submission, which will automatically read her GPS coordinates and use them to create a point feature.

FME Data Express lets mobile users run any workspace published to FME Server, letting them get access or create data on the go. It can also allow users to share their location or upload images directly from their device camera. This app unlocks powerful mobile data integration workflows and saves vital time for users back at the office.



Exercise

Try running Frank’s original workspace, but this time, request an extract of the **Libraries** table in **GeoJSON**. Open the results in your preferred application and examine them. Use the results to answer the quiz question below.

Optionally, try running your workspace using FME Data Express.

1After publishing a workspace to FME Server, only the original author can run the workspace via the web interface.

○ A.True
○ B.False

2If I want to view applicable results directly in my browser, which FME Server service should I use when I run a workspace?

○ A.Data Download
○ B.Data Streaming
○ C.Job Submitter
○ D.Notification

3Which product in the FME platform is designed to run workspaces from a mobile device?

○ A.FME Cloud
○ B.FME Data Express
○ C.FME Desktop
○ D.FME Server

4From the exercise results, which Vancouver library branch has the fewest number of books available for borrowing (BookCount)?

○ A.Accessible Services
○ B. Central Branch
○ C.Firehall
○ D.Strathcona

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Automate Workflows on Schedule

Learning Objectives

After completing this unit, you'll be able to:

- Explain how FME Server Automations can save time.
- Create an Automation to run a workspace on a schedule.
- See the other Automation options available.

Resources

- Interactive tutorial to help you follow along
- Starting workspace (should already be published to FME Server as per the previous unit's instructions)

Video

Automate Your Use of FME Server

Fatima tells Frank that she would like to receive an email with an Excel report of all the food vendors once a month. Frank offers to use FME Server Automations to create that workflow for her.

Automations takes data integration to the next level. Designing your FME workspaces with this in mind lets you create reusable, modular, and integrated applications for your organization, rather than one-off, single-use workspaces for every workflow. Automations support automated schedules and event-driven workflows that can:

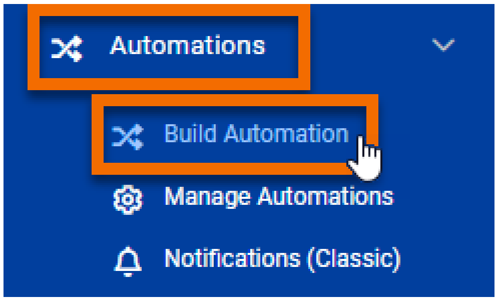
- Eliminate manual intervention.
- Remove schedule-based delays.
- Ensure data is always current and available.

Automations let you implement the **enterprise integration patterns** popularized by Gregor Hohpe and Bobby Woolf: repeatable solutions to commonly occurring problems encountered when integrating applications or systems.

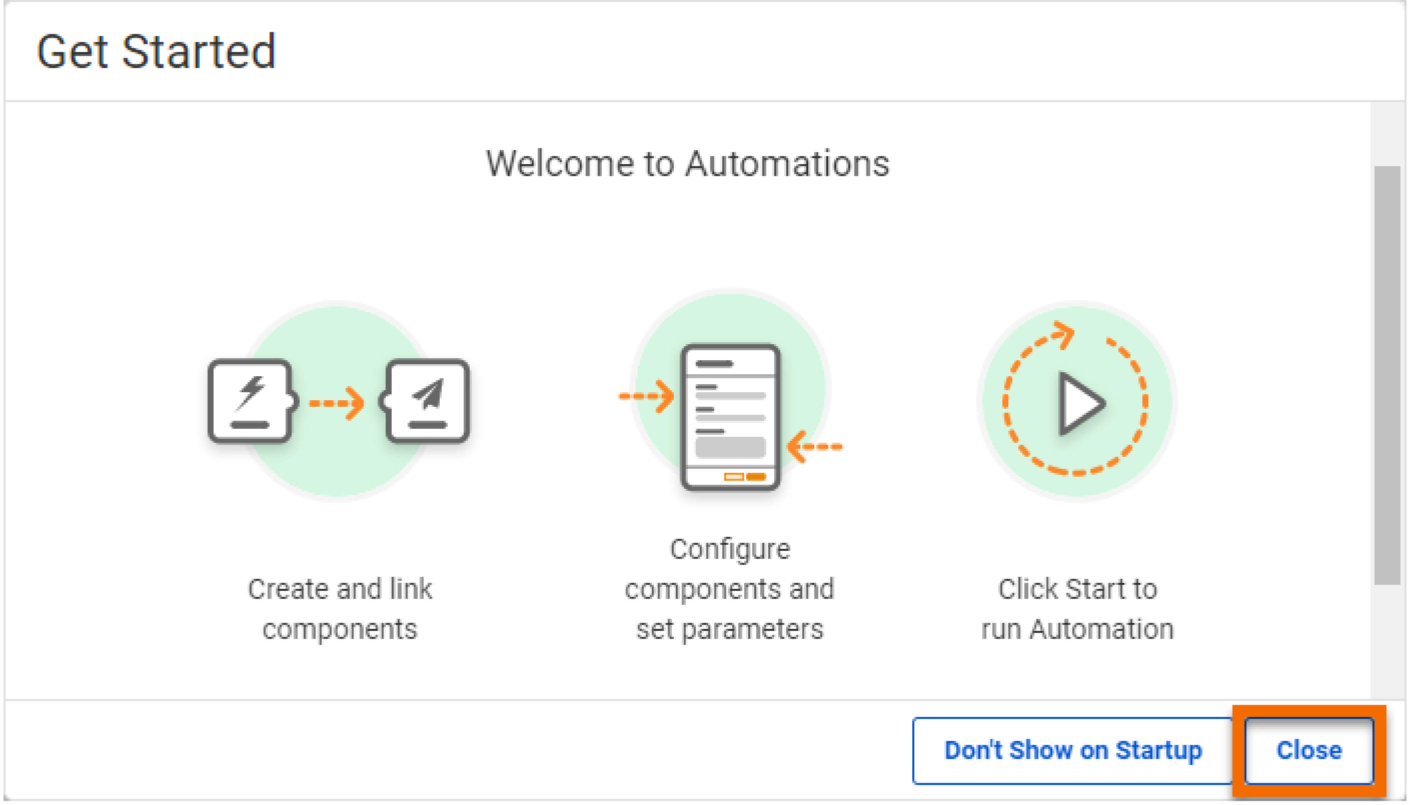
Create an Automation Triggered on a Schedule

Frank needs to create a new Automation to run his self-serve workspace once a month and send the results as an Excel file to Fatima.

He logs into FME Server (2022.0 or later) and clicks **Automations > Build Automation** in the web interface menu on the left side of the window.



Because he has built Automations before, he clicks **Close** when the **Get Started** window appears.



Get Started

Welcome to Automations

Create and link components

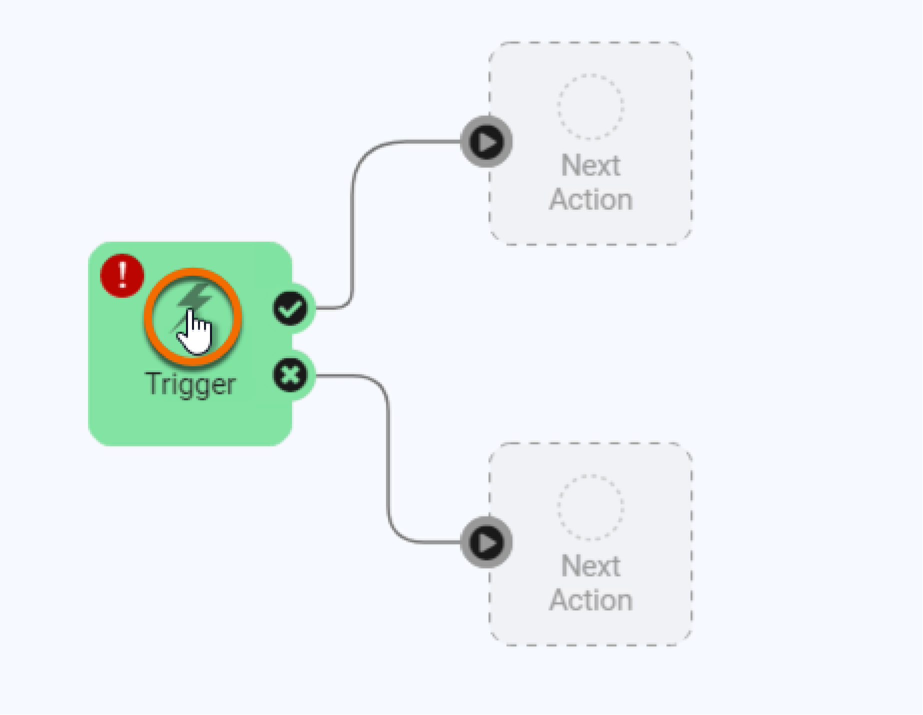
Configure components and set parameters

Click Start to run Automation

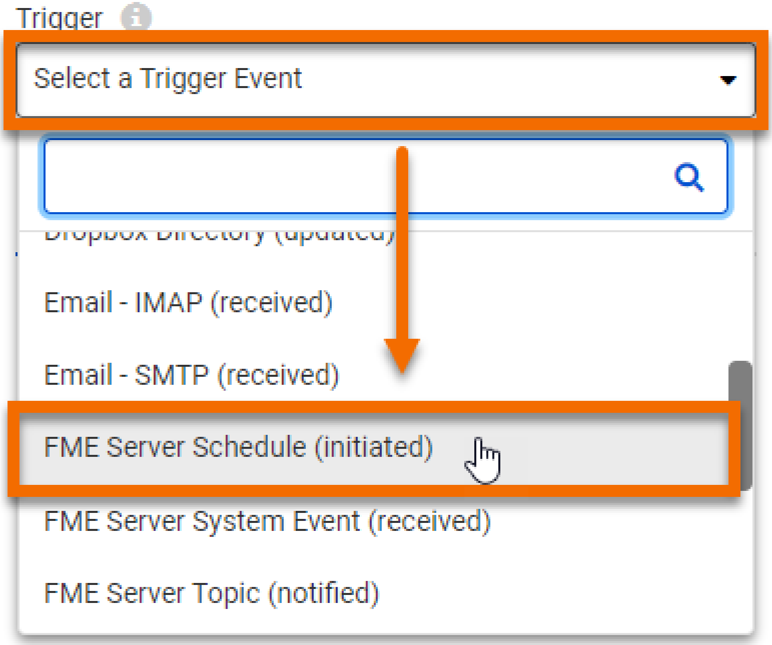
Frank knows all Automations are composed of three objects:

- **Triggers** listen for and receive messages from an external client or from within FME Server. Every Automation begins with a Trigger.
- **Actions** process messages between triggers and external actions, either in an FME workspace or through another tool.
- **External Actions** send a message to an external client or another process in FME Server. What happens afterward is no longer part of this Automation.

A green **Trigger** is already on the canvas. He double-clicks the **Trigger** to open the **Trigger Details** pane.



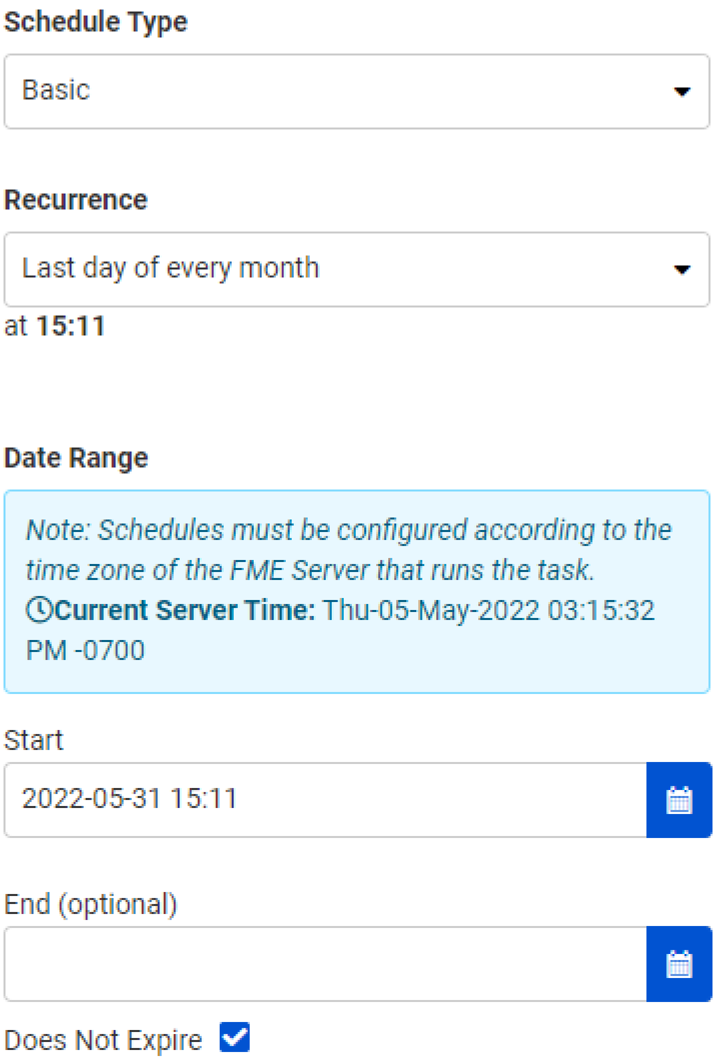
He clicks the **Trigger** drop-down and scrolls down to select **FME Server Schedule (initiated)**.


Trigger
Select a Trigger Event
Dropbox Directory (updated)
Email - IMAP (received)
Email - SMTP (received)
FME Server Schedule (initiated)
FME Server System Event (received)
FME Server Topic (notified)

You can trigger Automations from a variety of sources, including cloud services, email, file directories, or webhooks.

Learn More

Frank notes that FME Server Schedules are representative of the time zone where FME Server was installed. He sets up a **Basic Schedule** with the **Start** date as the last day of the current month and **Recurrence** as **Last day of every month**.



Schedule Type

Basic

Recurrence

Last day of every month

at **15:11**

Date Range

Note: Schedules must be configured according to the time zone of the FME Server that runs the task.
Current Server Time: Thu-05-May-2022 03:15:32 PM -0700

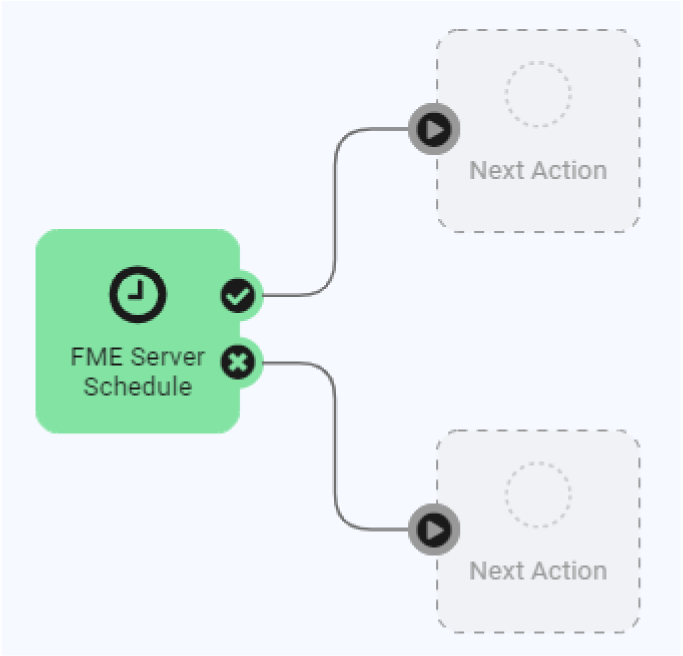
Start

2022-05-31 15:11

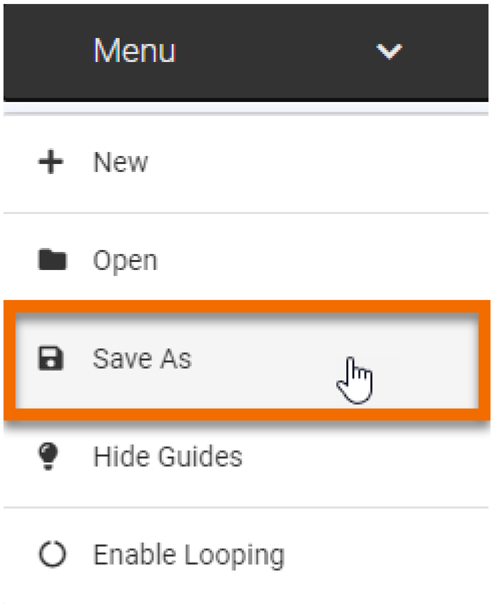
End (optional)

Does Not Expire ☑

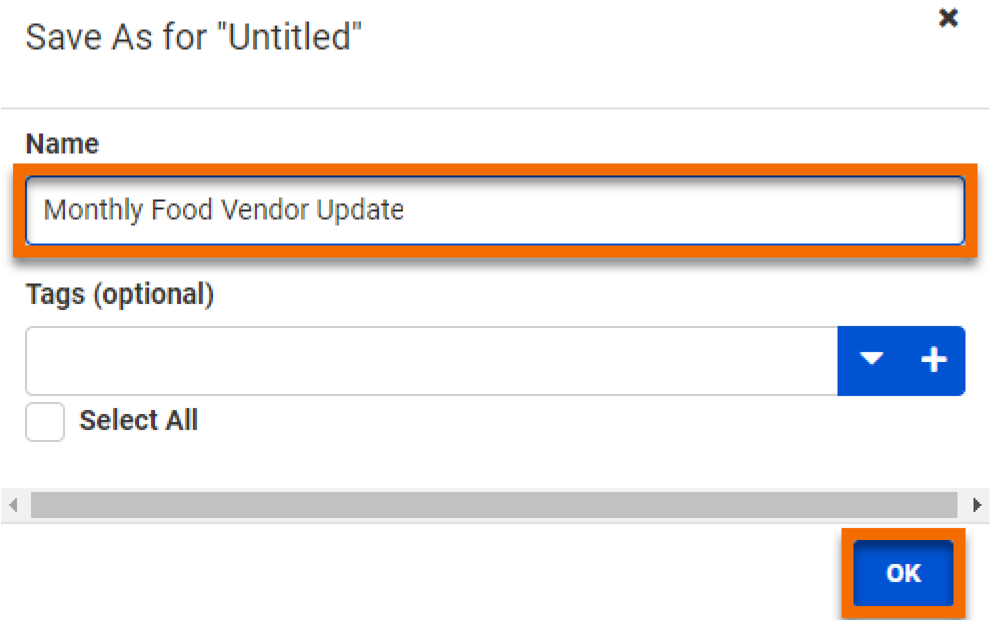
He clicks **Apply** and the **Trigger** changes to a **Schedule** to show it’s been set.



Then he clicks **Menu > Save As**.



He names the Automation “Monthly Food Vendor Update” and clicks **OK**.


Save As for "Untitled"

Name

Monthly Food Vendor Update

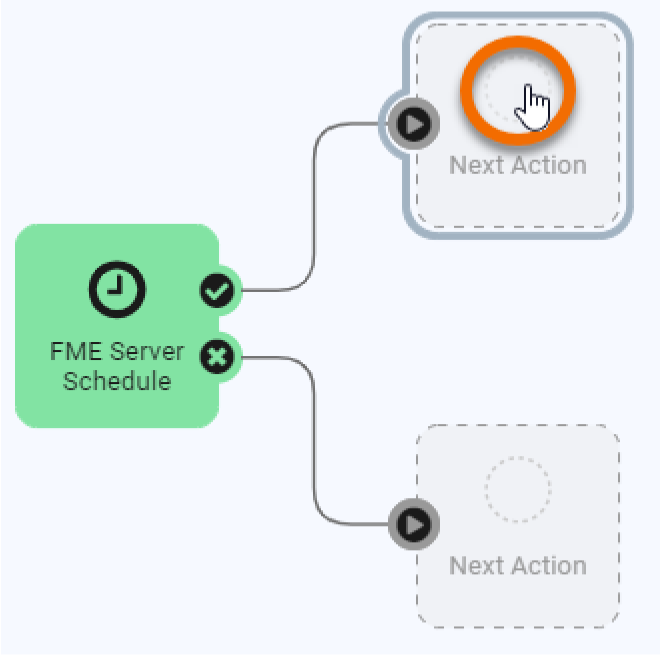
Tags (optional)

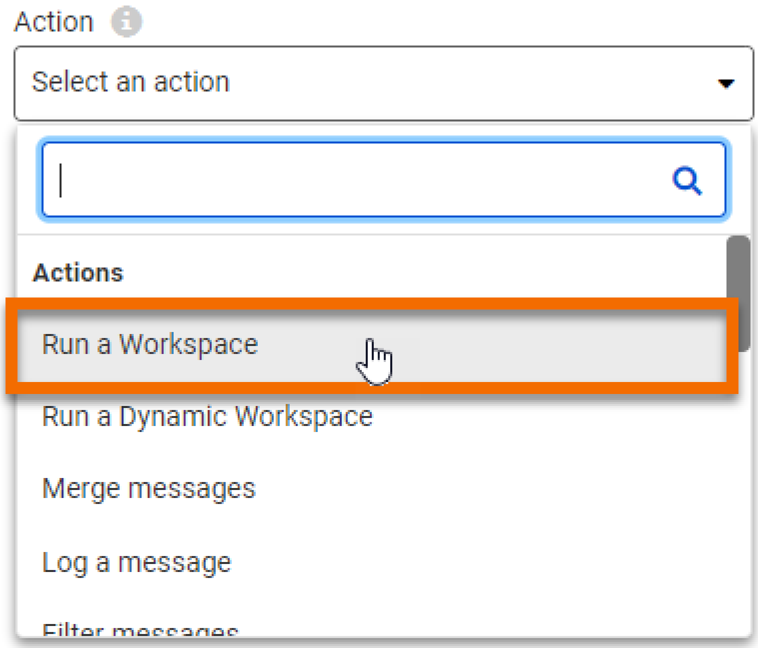
□ Select All

Run a Workspace in Response to a Trigger

Now that he’s set a **Trigger**, he needs to add an **Action** that will occur when the trigger is activated.

He clicks on the empty circle icon above **Next Action** to open the **Next Action Details** pane. Then he selects **Run a Workspace** from the drop-down options.





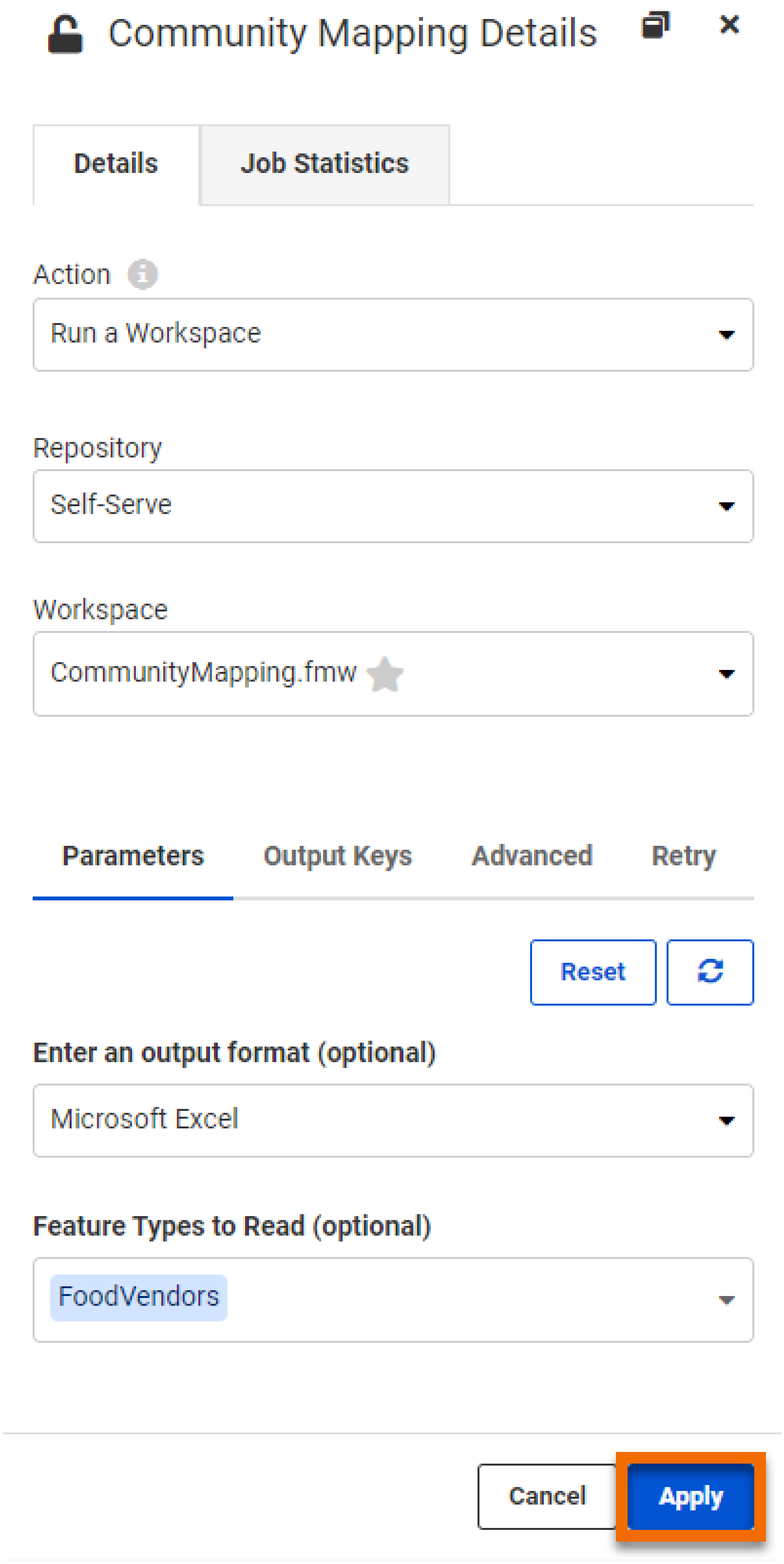
Other **Actions** include:

- **Merge Messages** lets you combine the results of previous actions, such as the results from two separate workspaces.
    - **Filter** lets you conduct tests from previous triggers or actions to control how your Automation runs. It’s like a [Tester transformer]() for Automations.
    - **Log** lets you add to the Automations logs for workflow reporting and debugging.

[Learn More]()

He will set the Automation to run the self-serve workspace. He clicks **Repository** and chooses **Self-Serve**, then for **Workspace** chooses **CommunityMapping.fmw**.

Next, he fills in the **Parameters**, choosing "Excel" for the **Output Format** and selecting "FoodVendors" from **Feature Types to Read**. With the **Parameters** filled in, he clicks **Apply**.



Community Mapping Details

Action: Run a Workspace

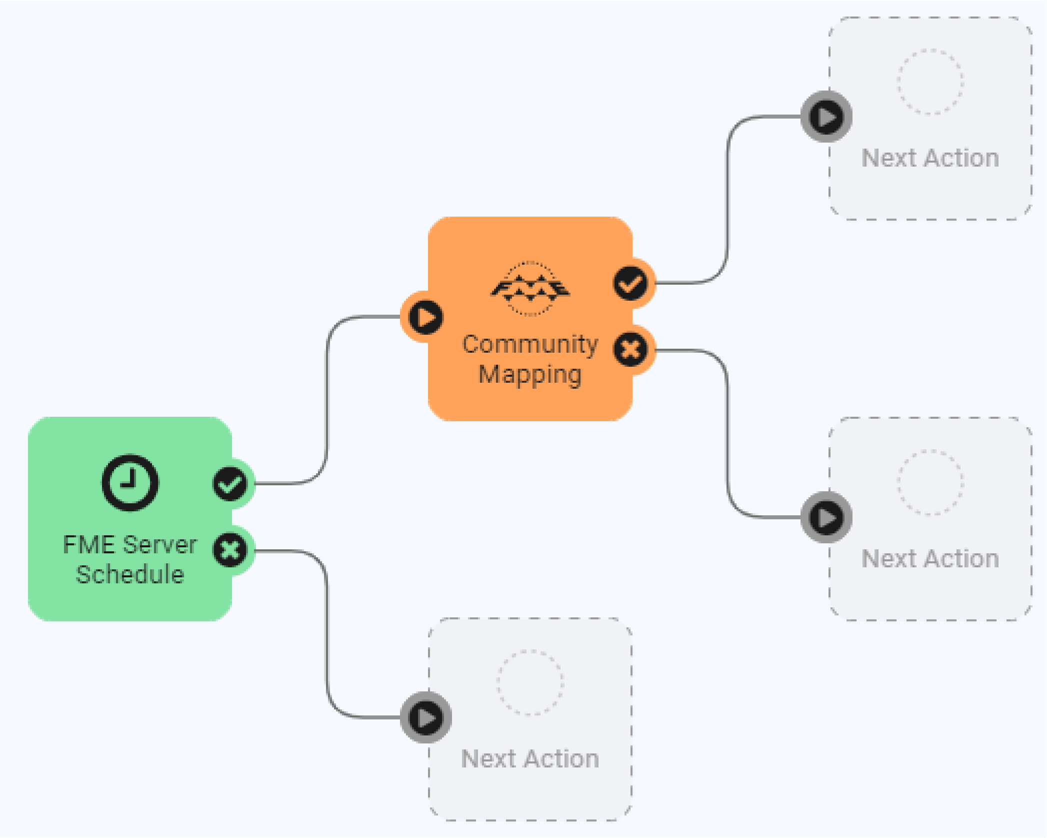
Repository: Self-Serve

Workspace: CommunityMapping.fmw

Enter an output format (optional): Microsoft Excel

Feature Types to Read (optional): FoodVendors

After clicking **Apply**, two new options appear on the canvas: a **Next Action** option for each port of the **CommunityMapping.fmw** workspace. The **✓ (check) port** action will occur if the workspace runs successfully, and the **X port** action will occur if the workspace fails. These branching options let you create robust Automations that account for failure.



Send an Email with the Results of a Workspace

Frank clicks the **Next Action** option coming out of the success port at the top and chooses **Email (send)** for the **Action**. He fills in the email account information for his own company Gmail address, using the Gmail template provided via **Load Template**.

If you are using Gmail or other email services requiring two-factor authentication, you will have to create an app password to allow FME Server access to your account. Use this instead of your normal password. For Gmail, see this tutorial. If you don't want to set that up, you can just read along.

A large number of External Actions are also available, including integration with APIs, Amazon Web Services, Azure, Dropbox, FTP, JMS, and Websockets.

Learn More

For the **Email To** parameter, he types in Fatima's email address, and for **Email From**, he enters his. For **Email Subject**, she enters "Monthly Food Vendor Report". For **Email Body** he enters:

“Hello Fatima,

Please find the most up-to-date Food Vendor data attached.

Best, Frank”

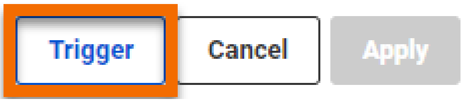
Finally, he needs to attach the data created by the workspace. For **Attachment** he enters the same path used for **Output Location** when the workspace runs, “$(FME_SHAREDRESOURCE_DATA)/CommunityMapping.zip”.

To test the email server connection, he clicks **Validate**. After a moment, an alert reporting the connection is **Valid** appears at the top of the pane. Seeing that it works, he clicks **Apply**.

His Automation is complete! He clicks the icon in the toolbar to save his Automation.

Then he clicks **Start Automation** in the top-right corner of the window.

The Automation starts. To test it (as it won’t run for a while), he clicks the clock icon on the **Schedule** trigger to open its parameters. There is a new button: **Trigger**. This button is available once an Automation is running and lets you test it by immediately running it once. He clicks **Trigger** to run the Automation once right now.



Fatima tells Frank she received an email: it worked. One less manual task for Fatima to do thanks to FME!

This Automation is very tailored to Fatima’s workflow. Frank could improve it if Fatima’s needs change. For example, he could create a parameter that controls if all the data is written (the current behavior), or just the data updated in the last month. He could also create a User Key to pass the output location path to the Email action instead of manually entering it in two places. He could even start building the FME Automations writer directly into his workspaces, creating modular pieces that fit into enterprise integration patterns.

Learn More

Exercise

Create an Automation like Frank’s.

If you wish, you can use your own email account for the **Email (send) External Action**, using one of the provided templates if appropriate. Instead of sending the report to Marcie, send an email to an account you monitor.

If you don't want to use email, you can simply use the **Log a message Action** instead. Use **Menu > View Log File** from the page with your running Automation to test if it's working.

1Which of the following scenarios would benefit from the use of an Automation running on a schedule?

○ A.Automatically updating a database every time a file is added to an S3 bucket.
○ B.Automatically providing a dataset whenever an email is received with a particular subject line.
○ C.Automatically making an API call to an asset management system whenever a GIS layer is updated.
○ D.Automatically providing delivery drivers with a list and map of their daily deliveries every morning.

2If Fatima changes her mind and wants a weekly update instead of a monthly one, where does Frank need to go in the Automation to make the change?

○ A.Email External Action > Email To parameter
○ B.Run a workspace Action > Output Keys
○ C.Schedule Trigger > Start parameter
○ D.Schedule Trigger > Recurrence parameter
○ E.None of the above

3Which of the following workflows could be created using Automations?

○ A.When an AutoCAD DWG file is uploaded to an Amazon S3 bucket, run it through a quality assurance testing workspace. If it passes, convert it to a GIS format and upload that to S3. If it fails, email the uploader.
○ B.Automatically run a nightly database backup.
○ C.When a field technician emails in a daily summary spreadsheet via email, email them back a report with a route map and weather forecast for their sites the next day.
○ D.When a user adds an Excel file of addresses to a Dropbox folder, automatically add another file with all the addresses geocoded.
○ E.All of the above

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.

Create a Self-Serve Web App

Learning Objectives

After completing this unit, you'll be able to:

- Explain how FME Server Apps let you share FME Server workflows with external users.
- Turn a workspace into an FME Server App.
- Use an FME Server App.

Resources

- Interactive tutorial to help you follow along
- Starting workspace (should already be published to FME Server as per the previous unit's instructions)

Video

Turn Workspaces into Apps

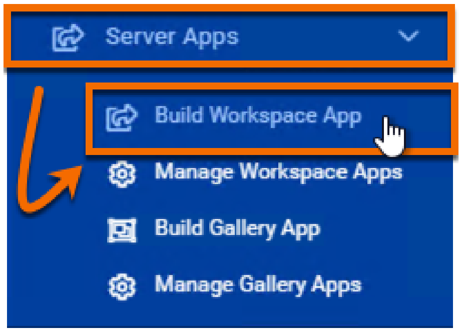
The word about self-serve data access is spreading around Fatima, Frank, and Jennifer's municipality. Frank is getting more and more users interested in accessing his self-serve workflows. However, not every member of the municipality — and certainly not the public — have FME Server logins. As the FME Server administrator, Frank's faced with the overhead of managing increasing numbers of user accounts or finding a way to provide users access to FME Server without creating accounts.

Thankfully he knows he can use FME Server Apps. Apps let you create custom web pages to provide a self-serve portal, allowing the user to upload data to be transformed by your workflow or to download data created as the output of a workflow. An FME Server login is not required to access an FME Server App. Sharing data and increasing accessibility is easy with FME Server Apps.

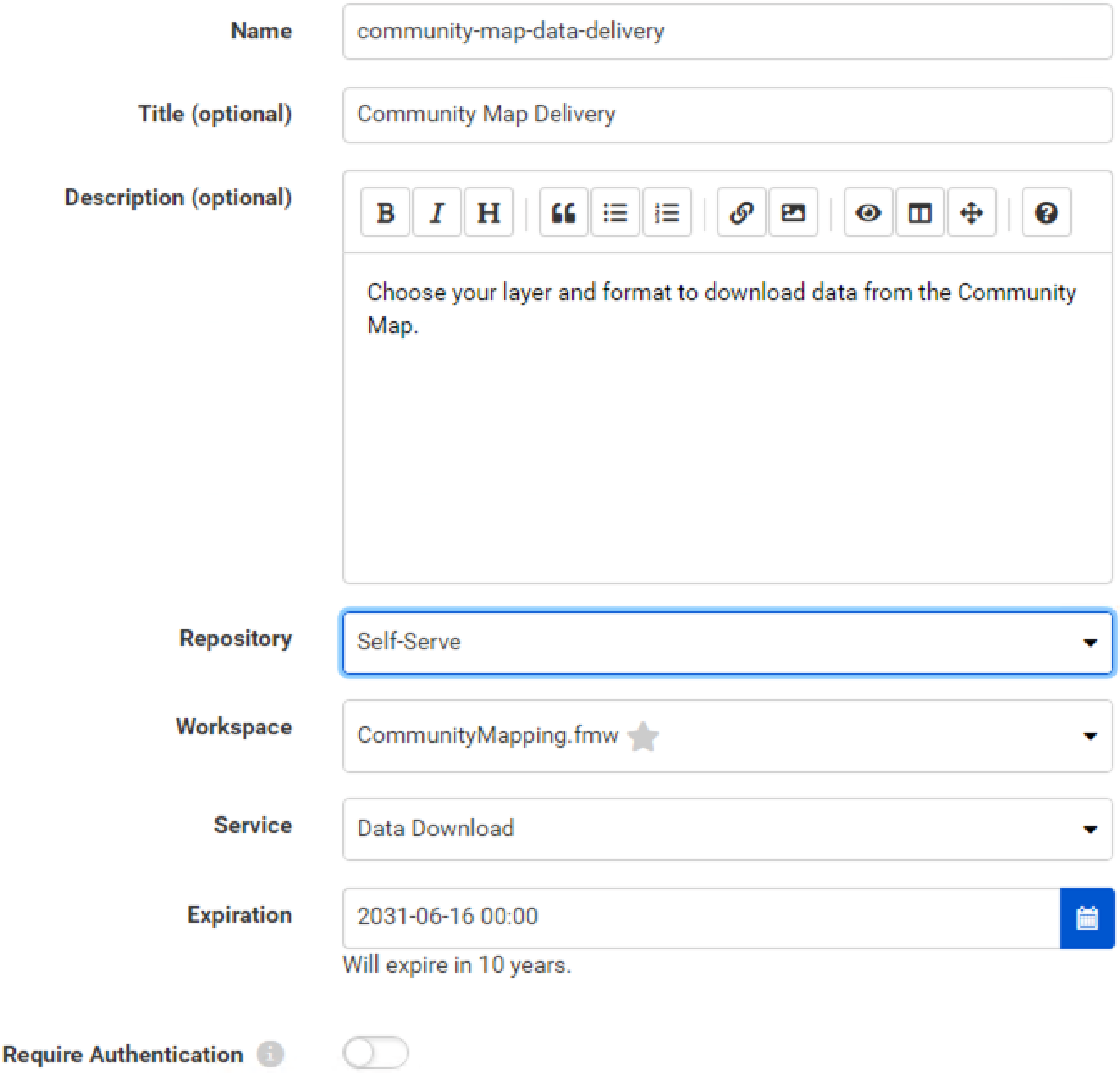
View an example FME Server App for self-serve data distribution.

Create a Self-Serve App

Frank wants to convert his self-serve workspace into a proper FME Server App. To do this, he logs into FME Server (2022.0 or later) and clicks **Server Apps > Build App**.



He fills out the form like this:



Name: community-map-data-delivery

Title (optional): Community Map Delivery

Description (optional): Choose your layer and format to download data from the Community Map.

Repository: Self-Serve

Workspace: CommunityMapping.fmw

Service: Data Download

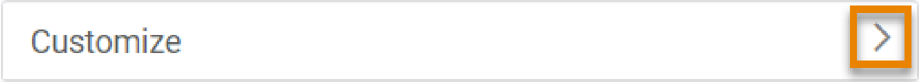
Expiration: 2031-06-16 00:00

Will expire in 10 years.

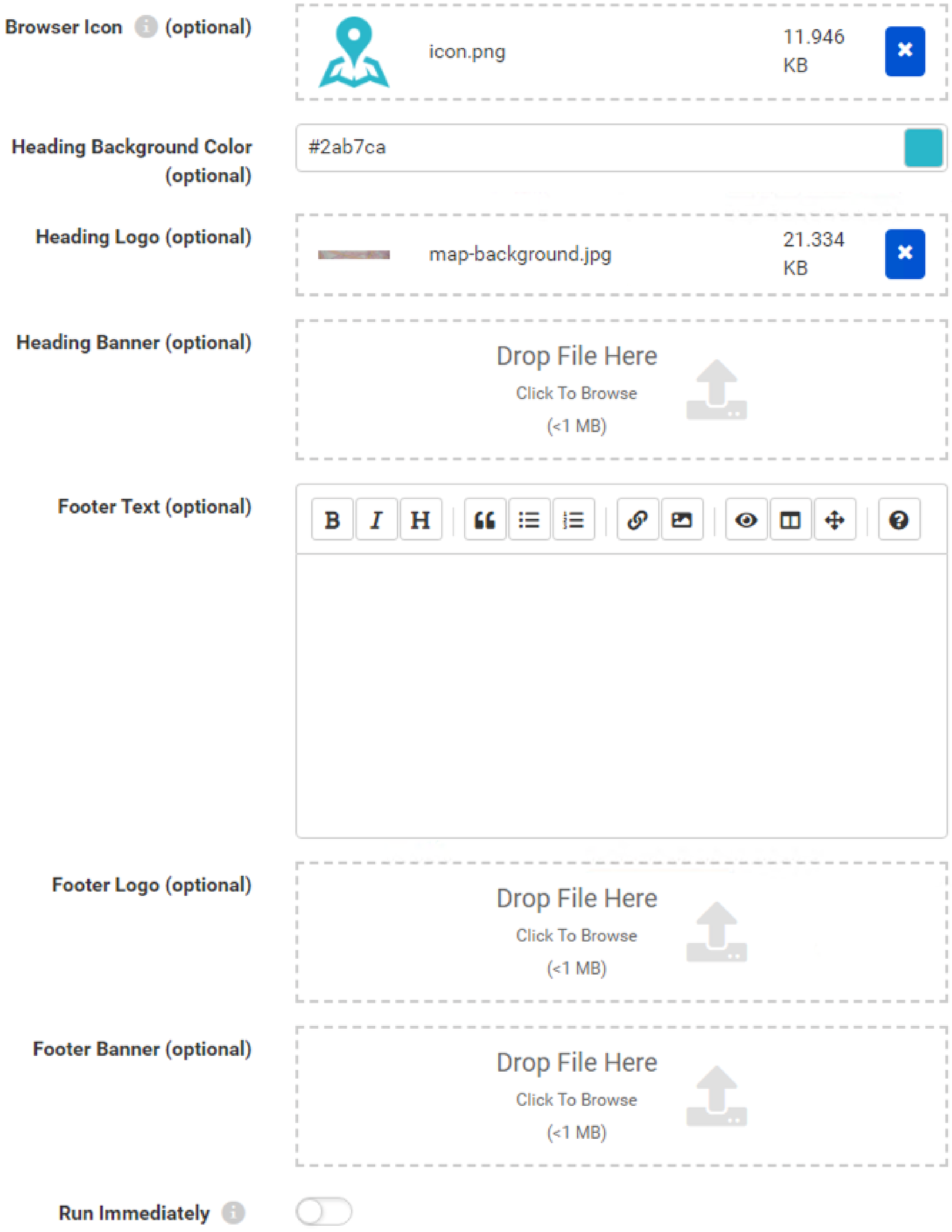
Require Authentication: ○

After selecting a workspace, the **Parameters** section appears below. This section lets you pick which published parameters you want users to fill out. Frank wants to have users fill them all out, so he skips this section.

He also chooses to customize his app to fit his organization’s branding. He clicks the right-pointing arrow to the right of **Customize** to expand that section.



He fills this section out like this:



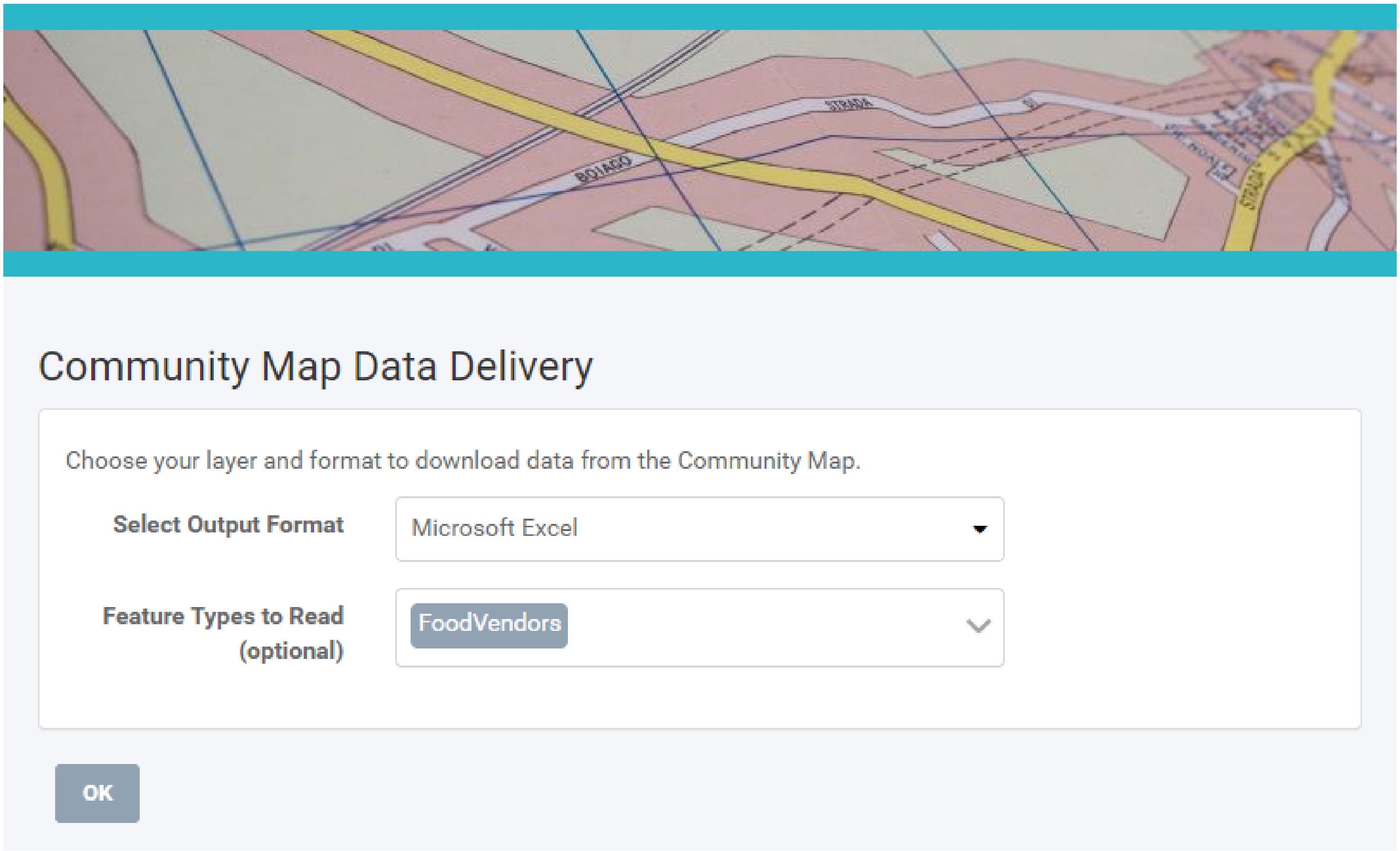
Browser Icon (optional)	icon.png 11.946 KB ✖
Heading Background Color (optional)	#2ab7ca 🟦
Heading Logo (optional)	map-background.jpg 21.334 KB ✖
Heading Banner (optional)	Drop File Here Click To Browse (<1 MB)
Footer Text (optional)	
Footer Logo (optional)	Drop File Here Click To Browse (<1 MB)
Footer Banner (optional)	Drop File Here Click To Browse (<1 MB)
Run Immediately	○

If you want to use the same **Heading Logo** and **Icon** as Frank, please use these links: map-background.jpg and icon.png. These images help customize your app.

With the page filled out, Frank clicks **OK**. He is taken to a new page confirming his app has been saved. He is given a URL that he can now share with anyone wishing to access the community map data. With a customized header and icon, users won’t even need to know they are interacting with FME Server! All they have to do is go to the link, fill out the form, and receive their data.

Use a Self-Serve App

Frank sends the URL to Fatima to test out. She clicks the link and fills the form out to download the FoodVendors data in an Excel spreadsheet.



Community Map Data Delivery

Choose your layer and format to download data from the Community Map.

Select Output Format: Microsoft Excel

Feature Types to Read (optional): FoodVendors

It works great! She can even send this link to her colleagues in the Business License Office, who can now get on-demand up-to-date versions of the data without having to bother Fatima or log into FME Server. Everyone saves time using the FME platform.

This example is just a small slice of what FME Server can do. The combination of Automations and Apps means you can create complex and tailored integrations for your organization. Because most modern web platforms provide webhooks or API endpoints, you can connect to almost anything using FME Server, all without writing any code.

Learn More

Exercise

Share your App with a colleague. Note that if you are using a locally-hosted FME Server, your App will be running on the IP of your machine and might not be accessible for someone else if you decide to share the URL, depending on your network setup.

1FME Server Apps let you turn a workspace into an app, while Automations let you connect workspaces to each other and triggers.

○ A.True
○ B.False

2By default, only people with FME Server usernames and passwords can access and use FME Server Apps.

○ A.True
○ B.False

3The user of an FME Server App will always know they are interacting with an FME Server.

○ A.True
○ B.False

Check the Quiz to Earn 100 Points

Second attempt earns 50 points. Three or more earns 25 points.