# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1.1</td>
</tr>
<tr>
<td>Getting Started (20 minutes)</td>
<td>1.2</td>
</tr>
<tr>
<td>What is FME?</td>
<td>1.2.1</td>
</tr>
<tr>
<td>FME Components</td>
<td>1.2.2</td>
</tr>
<tr>
<td>FME Workbench</td>
<td>1.2.3</td>
</tr>
<tr>
<td>Exercise 1.1</td>
<td>1.2.4</td>
</tr>
<tr>
<td>Quiz</td>
<td>1.2.5</td>
</tr>
<tr>
<td>Translations (30 minutes)</td>
<td>1.3</td>
</tr>
<tr>
<td>Generate Workspace</td>
<td>1.3.1</td>
</tr>
<tr>
<td>Exercise 2.1</td>
<td>1.3.2</td>
</tr>
<tr>
<td>Inspecting Data</td>
<td>1.3.3</td>
</tr>
<tr>
<td>Exercise 2.2</td>
<td>1.3.4</td>
</tr>
<tr>
<td>Feature Caching</td>
<td>1.3.5</td>
</tr>
<tr>
<td>Exercise 2.3</td>
<td>1.3.6</td>
</tr>
<tr>
<td>Schema and Data Model</td>
<td>1.3.7</td>
</tr>
<tr>
<td>Exercise 2.4</td>
<td>1.3.8</td>
</tr>
<tr>
<td>Quiz</td>
<td>1.3.9</td>
</tr>
<tr>
<td>Transformations (45 minutes)</td>
<td>1.4</td>
</tr>
<tr>
<td>Transformers</td>
<td>1.4.1</td>
</tr>
<tr>
<td>Exercise 3.1</td>
<td>1.4.2</td>
</tr>
<tr>
<td>Locating Transformers</td>
<td>1.4.3</td>
</tr>
<tr>
<td>Exercise 3.2</td>
<td>1.4.4</td>
</tr>
<tr>
<td>Common Transformations</td>
<td>1.4.5</td>
</tr>
<tr>
<td>Exercise 3.3a</td>
<td>1.4.6</td>
</tr>
<tr>
<td>Exercise 3.3b</td>
<td>1.4.7</td>
</tr>
<tr>
<td>Quiz</td>
<td>1.4.8</td>
</tr>
<tr>
<td>Workflows (30 minutes)</td>
<td>1.5</td>
</tr>
<tr>
<td>The Workflow</td>
<td>1.5.1</td>
</tr>
<tr>
<td>Multiple Readers, Writers, and Feature Types</td>
<td>1.5.2</td>
</tr>
<tr>
<td>Exercise 4.1</td>
<td>1.5.3</td>
</tr>
<tr>
<td>Best Practice</td>
<td>1.5.4</td>
</tr>
<tr>
<td>Exercise 4.2a</td>
<td>1.5.5</td>
</tr>
<tr>
<td>Exercise 4.2b</td>
<td>1.5.6</td>
</tr>
<tr>
<td>Quiz</td>
<td>1.5.7</td>
</tr>
<tr>
<td>Wrap-up</td>
<td>1.5.8</td>
</tr>
</tbody>
</table>
Introduction to FME Desktop

Overview
This course introduces the essential components and capabilities of FME through hands-on problem-solving exercises. It is a condensed version of the more in-depth FME Desktop Basic training and is a great start to become an efficient user of FME.

The course covers:

- Format Translations
- Transformation Tools
- Common Workflows

Learning Objectives

- Learn what FME Desktop is and what it can do for you
- Convert data from one format to another using FME Workbench
- View and inspect data using the FME Data Inspector
- Manipulate data structure and content with transformers
- Work with multiple datasets in a single workspace
- Apply best practices to workspaces

Prerequisites

- None

Training Resources
To complete this training, you need a licensed version of FME Desktop installed. You can request a free trial or may be eligible for a grant license. The training material is produced using FME Desktop 2018.1, but many of the steps are applicable to older versions.

You can download the data [here](#). After downloading, extract the data to C:/FMEData2018.

Overlap with FME Desktop Basic Training

<table>
<thead>
<tr>
<th>Introduction</th>
<th>Basic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>-</td>
</tr>
<tr>
<td>Getting Started (20 minutes)</td>
<td>Similar skills as first half of Data Translation Basics</td>
</tr>
<tr>
<td>What is FME?</td>
<td>What is FME?</td>
</tr>
<tr>
<td>FME Components</td>
<td>FME Desktop Components</td>
</tr>
<tr>
<td>FME Workbench</td>
<td>Introduction to FME Workbench</td>
</tr>
<tr>
<td>Exercise 1.1</td>
<td>Similar skills as Exercise 1: Exploring FME</td>
</tr>
<tr>
<td>Quiz</td>
<td>-</td>
</tr>
<tr>
<td>Translations (30 minutes)</td>
<td>Similar skills as second half of Data Translation Basics and first half of Data Transformation</td>
</tr>
<tr>
<td>Generate Workspace</td>
<td>Creating a Translation</td>
</tr>
<tr>
<td>Exercise 2.1</td>
<td>Similar skills as Exercise 2: Basic Workspace Creation</td>
</tr>
<tr>
<td>Inspecting Data</td>
<td>Introduction to the FME Data Inspector</td>
</tr>
<tr>
<td>Exercise 2.2</td>
<td>Similar skills as Exercise 3: Basic Data Inspection</td>
</tr>
<tr>
<td>Feature Caching</td>
<td>Integrated Inspection</td>
</tr>
<tr>
<td>Exercise 2.3</td>
<td>-</td>
</tr>
<tr>
<td>Schema and Data Model</td>
<td>Structural Transformation</td>
</tr>
<tr>
<td>Exercise 2.4</td>
<td>Similar skills as Ex 1: Schema Editing and Exercise 2: Structural Transformation</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Quiz</td>
<td>-</td>
</tr>
<tr>
<td><strong>Transformations (45 minutes)</strong></td>
<td>Similar skills as first third of Practical Transformer Use</td>
</tr>
<tr>
<td>Transformers</td>
<td>Transformation with Transformers</td>
</tr>
<tr>
<td>Exercise 3.1</td>
<td>-</td>
</tr>
<tr>
<td>Locating Transformers</td>
<td>Locating Transformers and children</td>
</tr>
<tr>
<td>Exercise 3.2</td>
<td>-</td>
</tr>
<tr>
<td>Common Transformations</td>
<td>Most Valuable Transformers</td>
</tr>
<tr>
<td>Exercise 3.3a</td>
<td>-</td>
</tr>
<tr>
<td>Exercise 3.3b</td>
<td>-</td>
</tr>
<tr>
<td>Quiz</td>
<td>-</td>
</tr>
<tr>
<td><strong>Workflows (30 minutes)</strong></td>
<td></td>
</tr>
<tr>
<td>The Workflow</td>
<td>Transformers in Series and Transformers in Parallel</td>
</tr>
<tr>
<td>Multiple Readers, Writers, and Feature Types</td>
<td>Multiple Readers and Writers</td>
</tr>
<tr>
<td>Exercise 4.1</td>
<td>-</td>
</tr>
<tr>
<td>Best Practice</td>
<td>Annotating Workspaces and Bookmarks</td>
</tr>
<tr>
<td>Exercise 4.2a</td>
<td>-</td>
</tr>
<tr>
<td>Exercise 4.2b</td>
<td>-</td>
</tr>
<tr>
<td>Quiz</td>
<td>-</td>
</tr>
<tr>
<td>Wrap-up</td>
<td>-</td>
</tr>
</tbody>
</table>
Introduction to FME Desktop

20 minutes

Learning Objectives

After completing this unit, you will be able to:

- Describe FME and what it does.
- Distinguish between FME Desktop, FME Server, and FME Cloud.
- Open a workspace.
- Run a workspace.

Module Overview

This module will teach you the basics of working with FME Desktop. You will learn what FME is, how it works, how to accomplish basic data translation and transformation, and how to set up your FME workflows following best practices. By the end of the module, you will be ready to create workspaces or to take further training specific to your needs. Because a wide variety of industries use FME, this module is a general introduction to FME that will show how FME works with data regardless of format or structure.

This first unit will introduce you to FME and the fundamental concepts necessary to use the software. After completing it, you will understand what FME can do and the differences between FME Desktop, Server, and Cloud. You will also open and run a workspace in FME Desktop.
What is FME?

FME (the Feature Manipulation Engine) is a data integration tool used for transforming data.

History

Safe Software began in a basement in 1993, helping forestry companies exchange maps with the provincial government. Sure, it was technically possible to share the maps back then, but only after hours fighting with the data. Often, an incredible amount of information was lost in the process.

In short, nobody was happy. We built FME to change that.

We didn’t start with a business plan, but we did have a desire to help. Our passion for freeing data was ignited, and we’ve been on a mission to help people experience that freedom ever since. We’re continuing to expand what’s possible with FME, the data integration software with the best spatial support in the world.

Data Integration

“Getting information off the Internet is like taking a drink from a firehose.”

- Mitchell Kapor, co-founder of Lotus and co-founder of the Electronic Frontier Foundation.

“You can have data without information, but you cannot have information without data.”

- Daniel Keys Moran, American computer programmer and science fiction writer.

"The world’s most valuable resource is no longer oil, but data.”

- The Economist, 2017

As the quotations above illustrate, the creation, manipulation, and analysis of data represent a significant challenge for contemporary organizations. Never before has so much machine-readable data existed, but organizations still struggle to find ways to use this mass of information to aid in decision-making.

What is Data Integration?

Data integration means combining information from various sources into something useful. It’s about efficiently managing data and making it available to those who need it. Both a technical and a business process, IBM defines it as “discovery, cleansing, monitoring, transforming and delivery of data from a variety of sources.” Data integration allows the combination and analysis of data across isolated “silos” where it would normally be difficult to collaborate. It allows organizations with multiple departments, facilities, software, and workflows to bring all of their data together.

Data Integration with FME

FME accomplishes data integration by reading data from multiple sources (here A and B), using transformer tools to change or restructure the data to fit the users’ needs, and writing it into a destination (C):
FME's data manipulation capabilities can be used to transform data, convert file formats, or do both simultaneously. While many data integration tools process only spreadsheet (i.e., tabular) data, FME can handle spatial data. It also uses a graphical interface, so no coding is required.

**How FME Works**

At the heart of FME is an engine that supports an array of data types and formats: Excel, CSV, XML, and databases, as well as various types of mapping formats including GIS, CAD, BIM, and many more.

The capability to support so many data types is made possible by a rich data model that handles all possible geometry and attribute types.

**Who Uses FME?**
FME has helped thousands of customers worldwide leverage their data so it can be used exactly where, when, and how it’s needed. Many of our customers are in the following industries:

- Architecture & Engineering
- Federal Government
- Local Government
- Oil & Gas
- Telecommunications
- Utilities

Here are a few examples of how people use FME, with links to more details.

**Vancouver International Airport**

The Vancouver International Airport (YVR) wanted to provide passengers with indoor mapping data via their mobile app. However, their indoor mapping data was all in CAD drawings. YVR became one of the first airports to provide indoor Apple Maps by using FME to convert their CAD drawings and business information into Indoor Mapping Data Format.

*Watch a video about this example.* *Watch a presentation about this example.*

**The Weather Network or Pelmorex Corp.**

The Weather Network or Pelmorex Corp. used FME to create the Pelmorex Lightning Detection Network (PLDN). Using FME Server, they provide clients with lightning strike data from their sensors all over Canada.

*Read a blog post about this example.*

**Tetrad Sitewise**
Tetrad’s Sitewise provides market analysis solutions to help their clients select sites for new business locations software using FME. Sitewise can analyze competition, parking, transit accessibility, combining multiple datasets to make an educated assessment. For more information, see this recorded webinar.

Watch a webinar about this example.

Visit our website for more customer stories.

**Data Integration Platform**

This module covers using FME Desktop for data translations and transformations at the desktop level. FME Desktop is one piece of software in the FME data integration platform:

- **FME Desktop** lets you connect and transform data.
  - For example, taking an Excel spreadsheet of business information and addresses and adding it to a MySQL database that is the backend to a citizen data access portal that allows searching for business license information.
- **FME Server** provides enterprise-level automation.
  - For example, allowing business licensing officers working for a city government to add new business licenses to the database in real time by sending an email or filling out a web form.
- **FME Cloud** is the hosted version of FME Server in the cloud.
  - For example, the city government above could use an FME Cloud account instead of hosting their own FME Server.
FME Components

FME users need to know several concepts. These concepts are necessary to create a common language among the multitude of formats FME supports. Additionally, they are required to describe the different components of FME translations.

Workspace Components

A workspace is the primary element in an FME translation and is responsible for storing a translation definition. The workspace is the container for all the functionality of a translation. It contains the following components:

Readers and Writers

A reader is the FME term for the component in a translation that reads a source dataset. Likewise, a writer is the component that writes to a destination dataset.

Readers and writers are represented by entries in the Navigator window.

Feature Types

Feature type is the FME term that describes a subset of records. Common alternatives for this term are layer, table, sheet, feature class, and object class. For example, each sheet in an Excel workbook, table in a database, or layer in a spatial data file is defined by a feature type in FME.

Feature types are represented by objects that appear on the Workbench canvas.

Features

Features are the smallest single components of an FME translation.

They aren’t individually represented within a workspace, except by the feature counts on a completed translation.

Relationships

Each workspace can contain multiple readers and writers, each of which can have multiple feature types, with multiple features. They exist in a hierarchy that looks like this:
Examples

Here is a table showing some examples of how common formats are treated by default in FME.

<table>
<thead>
<tr>
<th>Format</th>
<th>Dataset</th>
<th>Reader/Writer</th>
<th>Feature Type</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excel</td>
<td>The .xls or .xlsx file, also known as a workbook</td>
<td>The definition in the workspace of how to read/write the dataset, the path to read or write</td>
<td>A worksheet (single table) within the workbook</td>
<td>Row in a worksheet</td>
</tr>
<tr>
<td>CSV</td>
<td>The .csv file (a single table)</td>
<td>The definition in the workspace of how to read/write the dataset, the path to read or write</td>
<td>A single table*</td>
<td>Row in a table</td>
</tr>
<tr>
<td>Esri Geodatabase</td>
<td>A .gdb file (either a system folder or a database)</td>
<td>The definition in the workspace of how to read/write the dataset, the path to read or write</td>
<td>A single ArcGIS feature class or table within the geodatabase</td>
<td>A single geometric feature plus attributes (e.g. a point, line, or polygon), or a row in a table</td>
</tr>
</tbody>
</table>

*Note: CSV data is stored in a plain text file, with no method of subdivision. Therefore each feature type in FME represents a different CSV file, instead of different divisions in a single file. For more info on each format see the R+W documentation.

Excel Breakdown

Here is a visual example using Excel. In the image below:

1. Dataset = the XLS or XLSX file (a.k.a. the workbook)
2. Feature types = the worksheets (a.k.a. the sheets or tables)
3. Feature = the row (with columns as attributes)
In Workbench

A workspace with multiple readers, writers, and transformers might look like this:

This workspace has two readers (each with three feature types), and three writers (with one, two, and one feature types). Each reader and writer is a different format, and each has a different name for its feature types. Note that feature types and transformers appear on the canvas and the Navigator window, while readers and writers appear only on in the Navigator.
FME Workbench

Let's take a closer look at FME Workbench. You can locate Workbench in the Windows start menu:

![Workbench in Windows Start Menu]

TIP

Feel free to follow along in Workbench during these informational sections if you wish. Alternatively, you can just complete the hands-on exercises in each unit, which will illustrate the concepts shown between exercises.

If you are using a virtual machine provided for training, we recommend opening Workbench now, as the first time opening on a new machine takes a moment.

Major Components of FME Workbench

The FME Workbench user interface has many major components:

- **Canvas**
- **Menu/Toolbar**
- **Navigator**
- **Parameters**
- ** Transformers**
- **Log**

Canvas
The FME Workbench canvas is where to define a translation. It is the primary window within Workbench:

By default the workspace reads from left to right; data source on the left, transformation tools in the center, and data destination on the right. Connections between each item represent the flow of data and may branch in different directions, merge together, or both.

Menu/Toolbar

The menu bar and toolbar contain many tools; for example, tools for navigating around the Workbench canvas, controlling administrative tasks, and adding or removing readers/writers:

Navigator

The Navigator window is a structured list of parameters that represent and control all of the components of a translation:

Transformer Gallery

The transformer gallery is a tool for the location and selection of FME transformation tools. The number of transformers (below, 497) will vary depending on the version of FME and any optional custom transformers installed:
Translation Log

The translation log reports on translations and other actions. Information includes any warning or error messages, translation status, length of translation, and the number of features processed:

Parameter Editor Window

The Parameter Editor window is for editing parameters for objects on the canvas window:
Exercise 1.1 | Opening and Running a Workspace

| Data | 3-1-1 case location details (XLS hosted on FTP) |
| Overall Goal | To open and run an FME workspace to explore what it can do with data |
| Demonstrates | Opening and running a workspace |
| Start Workspace | C:\FME\Data\IntroToDesktop\Ex1.1-Begin.fmw |
| End Workspace | None |

Rather than trying to explain what FME is and does, let’s try it for ourselves! In this exercise you will open the completed version of the FME workspace you will create in this module's exercises. We will explore it, run it, and look at the output.

The workspace provides an example of how a municipal government could use FME to summarize 3-1-1 call records. 3-1-1 is a special phone number used by municipalities in Canada and the United States to manage non-emergency municipal service calls.

1) Locate Workspace File

Translations and transformations in FME are defined in an FMW file.

Using a file explorer, browse to the file C:\FME\Data\IntroToDesktop\Ex1.1-Begin.fmw:

Double-click on the file. It will open an application called FME Workbench. You might be prompted to choose an application to use; if so, select FME Workbench 2018.1 win64.

2) Explore FME Workspace

When FME Workbench opens you will see the option of completing Getting Started exercises. You can complete these now if you wish, or come back to them later. For now, click the ‘X’ to close the window.

The central part of the application will look like this:

This part we call the canvas. It is where the translation and transformation of data are defined graphically. Although it might appear complicated, it does not take much practice with FME to create workflows of this type.

Examine the left-hand side of the canvas:
This area is where we read data. This object is a spreadsheet of 3-1-1 call records stored in an Excel workbook (XLSX).

Now look at the right-hand side:
This area is where we write data. The objects here represent a report of the call records as a simple webpage (HTML), an Excel file (311-requests-summary), and a CSV file with cleaned and edited data (311-requests).

In between the reader and writer are objects that transform data. They alter the structure (e.g., attribute names) and content (e.g., values of attributes) of the data as it moves from left to right.

Labels and other annotations show us what the workspace does:

- Reads 3-1-1 data from CSV
- Cleans the data by removing missing values and fixing inconsistent values
- Removes attributes that aren't needed in the reports
- Writes the edited original data to CSV
- Summarizes the number of cases by department for each local planning area, writing the results to Excel
- Generates an HTML report with a table and graph for each local area
- Writes the report to HTML

### 3) Run FME Workspace

Let's run this workspace.

Before doing so, we want to control how to run the workspace. By default, a feature called Run with Prompt is turned on. We don't need this on for this course, so let's turn it off by clicking the Run with Prompt button on the toolbar:
OK, now we are ready to run the workspace. Click on the Green run button on the Workbench toolbar:

The workspace will now run. As it does so, you will see messages pass by in a log window. You may also see numbers appear on the canvas connections and green annotated icons on each object. We will discuss these later!

4) Locate and Examine Output

Once the translation is complete, click on the HTML writer feature type. Choose the option to Open Containing Folder:

In the Explorer dialog that opens you will find the new HTML, Excel, and CSV datasets:

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>311-requests</td>
<td>12/6/2018 10:36 AM</td>
<td>OpenOffice.org XML ...</td>
<td>11,506 KB</td>
</tr>
<tr>
<td>311-requests-summary</td>
<td>12/6/2018 10:36 AM</td>
<td>Microsoft Excel Work...</td>
<td>33 KB</td>
</tr>
<tr>
<td>report</td>
<td>12/6/2018 10:36 AM</td>
<td>Chrome HTML Docu...</td>
<td>103 KB</td>
</tr>
</tbody>
</table>

Open a web browser such as Firefox or Chrome. Open the report.html file created by FME (usually Ctrl+O or File > Open is the easiest way). You will see a table and a graph of the total number of cases by department for each local planning area. FME generated all of this from the original Excel spreadsheets:
This small demonstration illustrates the power of FME. This workspace read data from multiple datasets and wrote it out to datasets in both spatial and spreadsheet (i.e., tabular) formats. In between it carried out a series of transformations, creating added value and information.

Keep this report open - you will need it for the next section, a quiz on this section's content.

**CONGRATULATIONS**

By completing this exercise, you have learned how to:
- Open an FME workspace
- Run an FME workspace
- Locate the output from an FME workspace
FME Concepts Quiz

Each section ends with a quiz to test your new knowledge. Make your selection and click "Check my answers" to check each individual question. If you want an explanation for the answer, click "Explain".

Note: your score won't be tallied; this is just for your own review purposes.

Which FME product lets you host your own server to automate workflows and run them on a schedule or in response to triggers?

- FME Desktop
- FME Server
- FME Cloud

Workspaces are built in FME Desktop, hosted by FME Server, and can be hosted in managed cloud instances using FME Cloud.

Which FME product offers a pay-as-you-go model to let us take care of hosting and managing your data integration workflows?

- FME Desktop
- FME Server
- FME Cloud

Workspaces are built in FME Desktop, hosted by FME Server, and can be hosted in managed cloud instances using FME Cloud.

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

- FME Desktop
- FME Server
- FME Cloud

Workspaces are built in FME Desktop, hosted by FME Server, and can be hosted in managed cloud instances using FME Cloud.

Most data is divided or categorized into tables, layers, sheets, or classes. What is the FME umbrella term for these subdivisions?

- Data Types
- Feature Types
- Object Types
- Record Types

The basic unit of translation in FME is the feature. Groups of these are known as feature types.

What is the hierarchy of FME components, going from largest to smallest?

- Workspaces define translations. Workspaces contain reader(s) and writer(s), which control how data is read and written. Each reader and writer has feature type(s), which are logical groupings of features, the atom of FME data. For spreadsheet data, feature types are tables and features are rows; for spatial data, feature types are layers and features are single geometric features like points, lines, or polygons.

Examine the output from Exercise 1.1. How many ENG - Streets cases were filed in the Downtown local planning area?

- 3,351
- 1,569
- 2,541
- 1,431

If you inspect C:\FMEData2018\Output\Training\Report.html by opening it in a browser, you should be able to find the table and chart for the Downtown local planning area. If you look at the table or chart, you can see there are 2,541 cases listed under ENG - Streets.
FME Translations

30 minutes

Learning Objectives

After completing this unit, you will be able to:

- Define what a workspace is and generate one in FME Workbench.
- Inspect data using FME Data Inspector.
- Explain the difference between FME Workbench and FME Data Inspector.
- Explain what a schema is and edit it in FME Workbench.
- Explain what schema mapping is and do it using transformers.
- Turn feature caching on and off and use it when building a workspace.
- Use partial runs with feature caching.

Translations

At its heart, FME is a data translation tool, and this is usually the first aspect users wish to learn.

Data translation is the term we at Safe Software use to refer to the conversion of data from one format to another. Have you ever had trouble opening data in an unusual format in your go-to application? Have you ever needed to load spreadsheet data into a database in a systematic way? Have you ever wanted to extract and convert web data in formats like HTML, JSON, and XML with no coding? Then you have come to the right place! FME makes data conversion easy, allowing you to translate between over 400 different formats.

We refer to translation rather than conversion to emphasize the goal of seamlessly letting data speak in the language of a different format. FME is designed to let you not just convert data from one format to another, but to create output data to your exact specifications.

In this unit, you will learn how to conduct basic data translations yourself using FME.
Generate Workspace

Workbench’s intuitive interface makes it easy to set up and run a simple format-to-format (‘quick’) translation.

The Start Tab

The Start Tab in FME Workbench includes different ways to create or open a workspace. The simplest method is Generate Workspace:

Create Workspace

You can also generate a workspace through File > Generate Workspace.

TIP

Generate Workspace is used to quickly set up a workflow to convert files from a single format. If you have multiple input formats, you can build on a generated workspace or start with a blank canvas. For this lesson, we will start out using one file; we will cover multiple formats later in the module.

Generate Workspace Dialog

The Generate Workspace dialog condenses all the choices into a single dialog box. It has fields for defining the format and location of both the data to be read, and the data to be written.
Red coloring in an FME dialog indicates mandatory fields. Users must enter data in these fields to continue. In most dialogs, the OK button is not activated until the mandatory fields are complete.

**Format and Dataset Selection**

A key requirement is the format of the source data. All format selection fields let you type to search formats, or use a pull-down menu to browse.

You can type format names to search available formats. Additionally, the drop-down list shows some of the most commonly used formats, so many favorite formats are instantly available:

Click 'More Formats' and a table opens showing all of the formats supported by FME.

The source dataset is another key requirement. Dataset selection fields are a text entry field, but with a browse button to open an explorer-like dialog for file selection.

Similarly, this dialog defines the Writer format and dataset:
Feature Types Dialog

Clicking OK on the Generate Workspace dialog causes FME to generate the defined workspace. However, whenever a source dataset contains multiple feature types, the user is first prompted to select which data to translate.

You choose the feature types to include through the Select Feature Types dialog. As mentioned earlier, feature type is another term for layer, table, sheet, feature class, and object class. For example, each layer in a DWG file, or each table in an Oracle database, is defined as a feature type in FME. Only selected feature types appear in the workspace. This dialog gives you control over which groups of data you want to use from your source.

Here, for example, is a Select Feature Types dialog where the user has chosen to include all available layers in the workspace:
Exercise 2.1  Basic Workspace Creation

<table>
<thead>
<tr>
<th>Data</th>
<th>3-1-1 case location details (XLS hosted on FTP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Goal</td>
<td>Create a workspace to translate XLS to CSV</td>
</tr>
<tr>
<td>Demonstrates</td>
<td>Basic workspace creation with FME Workbench</td>
</tr>
<tr>
<td>Start Workspace</td>
<td>None</td>
</tr>
<tr>
<td>End Workspace</td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex2.1-Complete.fmw</td>
</tr>
</tbody>
</table>

Congratulations! You have just landed a job as a technical analyst in the Digital and Contact Centre Services department of your local city. You will be working to manage and analyze the data they collect from their 3-1-1 Contact Centre. 3-1-1 is a local phone number used widely across cities in the United States and Canada for citizens to call in and request services. Most cities also collect this data online now as well.

On your first day, you've been asked to do a simple file format translation. The 3–1–1 case data is currently stored as XLS files, one for each past month. These files are stored on the city's FTP server. To host this public data in a format anyone can access, your manager wants you to translate the 12 XLS files from 2017 into a single CSV file covering the entire year.

We’ve outlined all of the actions you need to take, though FME's interface is so intuitive you should be able to carry out the exercise without the need for these step-by-step instructions.

1) Start FME Workbench

If it is not open already, start FME Workbench by selecting it from the Windows start menu. You'll find it under Start > FME Desktop 2018.1 > FME Workbench 2018.1.

If Workbench is already open, click on the Start tab above the main canvas.

2) Select Generate Workspace

In the Create Workspace section of the Start page, select the option to Generate (Workspace). Alternatively, you can use the shortcut Ctrl+G.

3) Define Translation

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

<table>
<thead>
<tr>
<th>Reader Format</th>
<th>Microsoft Excel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reader Dataset</td>
<td>ftp://webftp.vancouver.ca/opendata/xls/CaseLocationsDetails_2017_XLS.zip</td>
</tr>
<tr>
<td>Note:</td>
<td>You can access the data locally at C:\FMEData2018\Resources\311\CaseLocationsDetails_2017_XLS.zip instead</td>
</tr>
<tr>
<td>Writer Format</td>
<td>CSV (Comma Separated Value)</td>
</tr>
<tr>
<td>Writer Dataset</td>
<td>C:\FMEData2018\Output\Training</td>
</tr>
</tbody>
</table>
The dialog will look like this:

Remember, you can set a format by typing its name, by selecting it from the drop-down list, or by choosing “More Formats” and selecting the format from the full table of formats. Note that the writer needs only a folder and not a specific file name.

You may click the Parameters buttons in the Generate Workspace dialog to check the reader/writer parameters, but none of them need changing in this exercise. For now, you can also ignore the Coordinate System setting and the Workflow Options.

4) Generate and Examine Workspace

Click OK to close the Generate Workspace dialog. FME Workbench generates a new workspace on the canvas:

You can expose the list of attributes by clicking the arrow icon on each object:
You can also expand the feature type to fit the entire name by hovering over its side until you get an expander arrow and then double-clicking:

5) Examine Bookmarks

Now that you've expanded your feature type, you might notice it no longer fits in the brown rectangle it was formerly contained in, labelled Reader Feature Types. This is a bookmark. Two were automatically created with Generate Workspace, but to follow FME best practice, we recommend using these to enclose sections of your workspace. Objects contained within them are grouped in the Navigator window and can be moved together on the canvas.

Bookmarks act like paragraphs for your workspaces, separating them into sections to keep them organized and help other users understand what each section does.

We'll use bookmarks later in the module, but for now let's just resize this one to contain the entire feature type. You can simply mouse over the bottom right corner until you see an expander arrow, and then click and drag to change the bookmark size to encompass the feature type:
Try moving the bookmark by clicking where it says "Reader Feature Types" and dragging. You'll notice the objects move with the bookmark.
Now your reader feature type should look like this:
6) Save Workspace

Save the workspace where ever you want. We will be using it in a later exercise. Remember there is a toolbar save button, and on the menu, there is File > Save As.

TIP

When a translation is run immediately without adjustment it’s known as a “Quick Translation.” Because FME is a ‘semantic’ translator; with an enhanced data model, the output from a quick translation is as close to the source data in structure and meaning as possible, given the capabilities of the destination format.

7) Run Workspace

Run the workspace by clicking the run button on the toolbar, or by using Run > Run Translation on the menu bar. The workspace runs and the Translation Log window reports a successful translation:
8) Locate Output

Locate the destination data in Windows Explorer to prove that it’s been written as expected (do not forget the Open Containing Folder button from Exercise 1):

For now, let’s use Notepad to ensure the data looks as we would expect. Right-click on CA18COV1 - Service Request Loca.csv and choose Open With > Notepad. If you look at the top row, you should see the attribute names we saw under the writer feature type:

- Year, Month, Day, Hour, Minute, Department, Division, Case_Type, Hundred_Block, Street_Name, Local_Area

In the next exercise, we’ll cover how to inspect the data using FME Data Inspector.

CONGRATULATIONS

By completing this exercise, you have learned how to:
- Generate an FME workspace
- Run an FME workspace
Inspecting Data

Introduction to Data Inspection

To ensure that you're dealing with the right information you need a clear view of your data at every stage of the transformation process. Data Inspection meets this need: it is the act of viewing data for verification and debugging purposes, before, during, or after a translation.

Introduction to the FME Data Inspector

The best place to start inspecting data in FME is in a complementary application called the FME Data Inspector.

What is the FME Data Inspector?

The FME Data Inspector is a utility that allows viewing of data in any of the FME supported formats. It is used primarily to preview data before translation or to verify it after translation.

The FME Data Inspector is closely tied to FME Workbench so that Workbench can send data directly to the Inspector. It's also closely connected to FME Workbench, to help set up and debug workspaces by inspecting data during the translation.

What the FME Data Inspector Is Not!

The FME Data Inspector isn’t designed to be a full-featured spreadsheet, database, or mapping application. It has no analysis or editing functionality, and the tools for symbology modification or printing are intended for data validation rather than producing output.

Starting the FME Data Inspector

To start the Data Inspector locate it in the Windows start menu:

![Image showing the Windows start menu with FME Desktop applications]

Major Components of the FME Data Inspector

When FME Data Inspector opens a dataset, it looks something like this:
View Window

The View window is the spatial display area of the FME Data Inspector. Multiple views of different datasets may be open at any one time.

Menu bar and Toolbar

The menu bar and toolbar contain many tools. Some are for navigating around the View window, some control administrative tasks such as opening or saving a dataset, and others are for particular functions such as selective filtering of data or the creation of dynamic attributes.
The Display Control window shows a list of the open datasets and their feature types. Tools here let users turn these on or off in the display, alter their symbology, and adjust the display order.

**Feature Information Window**

When you query a feature in the View window, the Information window shows information about that feature. This information includes the feature's feature type, attributes (both user and format attributes), coordinate system and details about its geometry.

**Table View Window**

The Table View window is a spreadsheet-like view of a dataset and includes all of the features and all of the attributes, with a separate tab for each feature type (layer).
<table>
<thead>
<tr>
<th>LibraryName</th>
<th>LibraryAddress</th>
<th>LibraryURL</th>
<th>BookCount</th>
<th>UserVisits</th>
<th>Circulation</th>
<th>OBJECTID</th>
<th>LibraryID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible Services</td>
<td>302-345 Robson St</td>
<td><a href="http://www.vpl">http://www.vpl</a>...</td>
<td>18322</td>
<td>2517</td>
<td>58288</td>
<td>8</td>
<td>BVA017</td>
</tr>
<tr>
<td>Carnegie</td>
<td>401 Main St</td>
<td><a href="http://www.vpl">http://www.vpl</a>...</td>
<td>26689</td>
<td>41089</td>
<td>184465</td>
<td>2</td>
<td>BVA002</td>
</tr>
<tr>
<td>Central Branch</td>
<td>350 W Georgia St</td>
<td><a href="http://www.vpl">http://www.vpl</a>...</td>
<td>1188300</td>
<td>1958528</td>
<td>1751704</td>
<td>3</td>
<td>BVA003</td>
</tr>
<tr>
<td>Firehall</td>
<td>1455 W 10th Av</td>
<td><a href="http://www.vpl">http://www.vpl</a>...</td>
<td>30413</td>
<td>157007</td>
<td>240695</td>
<td>4</td>
<td>BVA007</td>
</tr>
<tr>
<td>Joe Fortes</td>
<td>870 Denman St</td>
<td><a href="http://www.vpl">http://www.vpl</a>...</td>
<td>41718</td>
<td>324515</td>
<td>370113</td>
<td>5</td>
<td>BVA010</td>
</tr>
<tr>
<td>Kitsilano</td>
<td>2423 Macdonald St</td>
<td><a href="http://www.vpl">http://www.vpl</a>...</td>
<td>61697</td>
<td>316304</td>
<td>531395</td>
<td>6</td>
<td>BVA013</td>
</tr>
<tr>
<td>Mount Pleasant</td>
<td>1 Kingway</td>
<td><a href="http://www.vpl">http://www.vpl</a>...</td>
<td>67402</td>
<td>431618</td>
<td>413059</td>
<td>7</td>
<td>BVA015</td>
</tr>
</tbody>
</table>
Let’s see how the FME Data Inspector interface works by inspecting the results of the last exercise.

1) Start FME Data Inspector

Start the FME Data Inspector by selecting it from the Windows start menu. You’ll find it under Start > FME Desktop 2018.1 > FME Data Inspector 2018.1.

2) Open Dataset in FME Data Inspector

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

To open a dataset, select File > Open Dataset from the menu bar, click the Open Dataset icon in the toolbar, or use Ctrl + O:

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

<table>
<thead>
<tr>
<th>Reader Format</th>
<th>CSV (Comma Separated Value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reader Dataset</td>
<td>C:/FMEData2018/Output/Training/CA18COV1 - Service Request Loca.csv</td>
</tr>
</tbody>
</table>

Click OK, and you will see a table appear in the Table View window. The main View will be blank and will say “There is no geometry in this View.” The View displays this message because we have loaded data with no geometry. As you will see later, if we load spatial data like a map layer, CAD drawing, or 3D model, it will be displayed here. If you scroll through the Table View, you can see the different values of the 3-1-1 request attributes:
3) Add Dataset in FME Data Inspector

Using Open Dataset always displays the data in a new view in Data Inspector, shown in a tabbed interface above the View window:

You can also add another dataset to the open view. Let’s try out this feature by adding a dataset we’ll be using in a later unit: local planning areas. This dataset is a KML file of Vancouver’s local planning areas, which are designated by the city for purposes of planning and development. It contains their geometry, i.e., the areas they cover, and attributes about them, e.g., their names.

To add a dataset to an open view, select File > Add Dataset from the menu bar, click the Add Dataset button on the toolbar, or use Ctrl + D:

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

<table>
<thead>
<tr>
<th>Reader Format</th>
<th>Google KML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reader Dataset</td>
<td>C:\FMEData2018\Data\Boundaries\LocalAreas.kml</td>
</tr>
</tbody>
</table>

Click OK, and you will see a new feature type appear in the Display Control window. Additionally, you will see the local areas appear in the View window:
4) Add a Background Map

The local areas currently appear without any context. Let’s improve the display by adding a background map. Select Tools > FME Options > Background Map (or on Mac, FME Data Inspector > Preferences > Background Map). Click the drop-down next to Background Format:

You will notice that FME supports a variety of background map services. Most require an existing account or server, but for this example let’s select Stamen Maps, which we can use without an account.

After selecting Stamen Maps, click Parameters. Click the ellipsis button to retrieve the list of background map styles available. Select terrain and click OK:

Click OK twice to close the FME Options windows. You should see the background map appear shortly:
You can use the pan and zoom tools in the Toolbar to move the view around the map:

![Toolbar tools]

**CONGRATULATIONS**

By completing this exercise, you have learned how to:
- Open datasets in a new view in the FME Data Inspector
- Add datasets in the open view in the FME Data Inspector
- Add a background map in the FME Data Inspector
Feature Caching

Sometimes it's important to be able to inspect data at any step of the translation. FME has an option to cache data automatically: feature caching. You can turn it on by clicking the Run with Feature Caching button on the toolbar:

When caching is turned on, running a translation causes data to cache at every part of the workspace. In subsequent runs, those caches can be used instead of having to re-run entire sections of the workspace.

Here, for example, a workspace has been run with caching turned on:

You can click the green magnifying glass icon to inspect the features at that point of the translation.

Partial Runs

A partial run is when only one section of a workspace executes.

Consider if the author above makes a change to the AreaCalculator parameters:

Notice that the caches change color (to yellow) on the AreaCalculator transformer and subsequent transformers. This color denotes that caches are stale; their data contents no longer match what the workspace would produce.

To get the new results, the author must re-run the workspace. However, they do not have to re-run the entire workspace; they can start the workspace at the point of change - the AreaCalculator:
**Run From This** causes the workspace to run from that point only, using data cached up until that point. Notice how hovering over the option highlights all “downstream” transformers. They are the only ones that will run. That makes the translation quicker.

The other option is Run To This. The author could use that option on the writer feature type and get much the same effect:

...but notice how the second branch from the StatisticsCalculator does not get highlighted. It will not run. That shows how you can avoid running a particular section of workspace.

---

**TIP**

A partial run is particularly useful in avoiding re-reading data from its source; especially when the data comes from a slow, remote location such as a web service.

Also, caches can be saved with the workspace when saved as a template. That means the workspace can be re-run using the caches from a previous session or even from another author!

---

**WARNING**

Be aware that while using feature caching and partial runs can speed up your development of a workspace, caching data obviously causes the translation to be slower and to use system resources such as disk space. Caching is very useful while developing a workspace, but should be turned off before putting a workspace into production.
Exercise 2.3  Feature Caching

<table>
<thead>
<tr>
<th>Data</th>
<th>3-1-1 case location details (XLS hosted on FTP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Goal</td>
<td>Inspect data</td>
</tr>
<tr>
<td>Demonstrates</td>
<td>Inspect data using feature caches</td>
</tr>
<tr>
<td>Start Workspace</td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex2.3-Begin.fmw</td>
</tr>
<tr>
<td>End Workspace</td>
<td>None</td>
</tr>
</tbody>
</table>

Let's use feature caching to inspect our data.

1) Start Workbench

Start Workbench (if necessary) and open the workspace from Exercise 2.1. Alternatively you can open C:\FMEData2018\Workspaces\IntroToDesktop\Ex2.3-Begin.fmw.

2) Use Feature Caching

Feature caching is turned off by default; you can see the magnifying glass and small play button icon in the top left of your toolbar:

Click the button to turn feature caching on, or go to Run > Run with Feature Caching. Then rerun your workspace. You should notice a little green square with a magnifying glass filling up on your reader feature type:

This icon represents a feature cache. The cache stores features at that point of the translation, and you can click it to inspect the data. Click the green icon to inspect the data at that point of the translation; your original Excel data will open in Data Inspector.

3) Use Partial Runs with Feature Caching
Feature caching also allows us to selectively run sections of our workspace.

If you select your reader feature type by clicking it, you will see some tool-tip icons appear above it. Here you have two partial runs options. Because this feature type is the first object in the workspace, you can choose to Run Just This, which in this case reads the data and caches it. You can also choose to Run From This, which will run from this object to the end of the workspace, running only what it is connected to.

If you mouse over each option, you will see the objects that will run if you click are highlighted. For Run From This, just the reader feature type is highlighted:

For Run From This, the writer feature type will also run:

Finally, you can select the writer feature type and see the option Run To This, which will run all connected objects preceding this one:
In this specific scenario these tools are not that useful, since we only have two objects in our workspace. However, as our workspace grows, partial runs help one test and develop small sections of your workspace at a time. For now, use **Run Just This** on your reader feature type. Then, click the green icon to inspect your read data.

**4) Sort Table View to Check for Missing and Inconsistent Values**

Now that Data Inspector is open, let’s use its Table View to identify problems in our source data.

The column headers in Table View contain the names of each attribute. If you right-click them you get a context menu with many options: to sort the column alphabetically (ascending or descending) or numerically (ascending or descending), or to clear all sorting.

Since ultimately we want to report on 3-1-1 calls by local planning area, let’s see if any of our features are missing values for the **Local_Area** attribute. To do so, scroll over in your Table View until you see the **Local_Area** attribute. Then, right-click and select **Sort Alphabetically Ascending**.
If you scroll down, you will find there are 4,122 features with missing values for **Local Area**!

If you take the time to look through, it turns out we have two kinds of missing data:

1. Some records do not have street addresses, leading them to not be associated with a local area.
2. There are also two rows of missing data in between each month of records. You can verify this by scrolling down to find the gaps between each month (where the **Month** attribute changes from 1 to 2, 2 to 3, etc.).

If you have a keen eye for data problems, you might also spot inconsistent naming for the values in the **Local Area** attribute. Two of the area names have some values with a hyphen in some of the values, but not all: Dunbar Southlands and Dunbar-Southlands and Arbutus Ridge and Arbutus-Ridge:

You can verify this problem by using the search bar in the bottom-left of Table View. Click the drop-down and select **Local Area**:
Compare the results (visible in the bottom right) of typing in "Arbutus" and "Arbutus-" in the text box with the magnifying glass. You'll find that of the 2,236 features in the Arbutus-Ridge local area, 116 have a hyphen between the words:

We will address both these data quality issues in a later exercise.

**TIP**

This kind of visual inspection of data is a good first step, but more thorough data validation is usually required. As this is an introduction, we won't go into details regarding validation. If you want to learn more about data validation with FME, check out this article.

**CONGRATULATIONS**

By completing this exercise, you have learned how to:
- Use FME Workbench feature caching to open a dataset in the FME Data Inspector
- Sort data in the FME Data Inspector Table View
Schema and Data Model

Transforming a dataset's structure requires using FME to manipulate schemas. FME uses the term "schema," but you may know this as the data model.

Schema Concepts

A schema defines the structure of a dataset. Each dataset has its unique schema; it includes layers, attributes, and other rules that define or restrict its content.

For example, the schema of an Excel workbook of (simplified) business data for a local bank chain might look like this:

<table>
<thead>
<tr>
<th>Worksheet</th>
<th>Column</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Name</td>
<td>String</td>
</tr>
<tr>
<td>Customers</td>
<td>Address</td>
<td>String</td>
</tr>
<tr>
<td>Customers</td>
<td>Phone</td>
<td>String</td>
</tr>
<tr>
<td>Customers</td>
<td>Account Number</td>
<td>Integer</td>
</tr>
<tr>
<td>Customers</td>
<td>Branch ID</td>
<td>Integer</td>
</tr>
<tr>
<td>Locations</td>
<td>Branch Name</td>
<td>String</td>
</tr>
<tr>
<td>Locations</td>
<td>Branch ID</td>
<td>Integer</td>
</tr>
<tr>
<td>Locations</td>
<td>Address</td>
<td>String</td>
</tr>
<tr>
<td>Locations</td>
<td>Phone</td>
<td>String</td>
</tr>
<tr>
<td>Locations</td>
<td>Sales</td>
<td>Integer</td>
</tr>
</tbody>
</table>

FME lets you preserve the schema of your data while converting it to a new format, optionally letting you change the schema to alter the structure of your data. For example, you could remove the Locations worksheet from the output data, or make a new attribute that classified bank branches by their sales bracket.

Schema Representation

When you create a new workspace, FME scans the source datasets. It creates a reader whose feature types appear on the left side of the workspace canvas and a writer whose feature types appear on the right side of the workspace canvas:

TIP

Each object in this illustration represents a subdivision in the source dataset. Remember, in FME terminology these objects are called feature types. Multiple feature types mean there are multiple layers or tables in the source dataset.

Reader Schema

For the reader, you can view more information about the schema by clicking the cog-wheel icon on each feature type object:
This Feature Type dialog has several tabs. Under the Parameters tab is a set of general parameters, such as the name of the feature type (in this case Libraries) the allowed geometry types, and the name of the parent dataset:

The User Attributes tab shows a list of attributes. Each attribute has information about its name, data type, width, and number of decimal places:

Each feature type has a different name and can also have a completely different set of attributes. All of this information goes to make up the reader schema. It is literally "what we have."

**Writer Schema**

As with the reader, each writer has a set of detailed schema information accessed by opening the dialog for a feature type:
By default, the writer schema ("what we want") is a mirror image of the source, so the output from the translation will be a duplicate of the input. This feature allows users to translate from format to format without further edits (Quick Translation).

**Schema Editing**

If "what we want" is different to the default schema definition, we have to change it using a technique called **schema editing**. This process involves altering the writer schema to customize the structure of the output data. One good example is renaming an attribute field.

After editing, the source schema still represents "what we have," but the destination schema now truly does represent "what we want."

**Schema Mapping**

When using Generate Workspace, the reader and writer schemas will be identical. However, when edits occur, these connections are usually broken.

**Schema mapping** is the process of connecting the reader schema to the writer schema in a way that ensures the correct reader features go to the correct writer feature types and the correct reader attributes go to the correct writer attributes.

FME permits mapping from source to destination in any arrangement desired. There are no restrictions on what feature types or attributes may be mapped.
Now that you created a workspace to translate the 3-1-1 data to CSV, the 3-1-1 department has requested that you remove the underscores from the attribute names and add a Date attribute. We can use a transformer to make these changes to the schema.

1) Start Workbench

Start Workbench (if necessary) and open the workspace from Exercise 1. Alternatively you can open C:\FMEData2018\Workspaces\IntroToDesktop\Ex2.3-Begin.fmw.

2) Rename Feature Type

Currently, the destination schema matches the source. However, the end user of the data has requested changes to the schema.

Inspect the writer feature type parameters by double-clicking it or clicking its cogwheel. Click in the field labelled CSV File Name and change the name from "CA18COV1 - Service Request Loca" to "311-requests":

Now when the workspace runs, the output will be named 311-requests.csv.

3) Update Attributes

Inspect the user attributes by clicking on the User Attributes tab. They will look like this:
Let's get rid of those underscores in the attribute names and add a new attribute, called `Date`, that we will supply values to later. Carry out the following actions:

<table>
<thead>
<tr>
<th>Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case_Type</td>
<td>Rename attribute to Case Type</td>
</tr>
<tr>
<td>Hundred_Block</td>
<td>Rename attribute to Hundred Block</td>
</tr>
<tr>
<td>Street_Name</td>
<td>Rename attribute to Street Name</td>
</tr>
<tr>
<td>Local_Area</td>
<td>Rename attribute to Local Area</td>
</tr>
<tr>
<td>Date</td>
<td>Create attribute Date of Type string</td>
</tr>
</tbody>
</table>

You can rename attributes by clicking in the Name cell and making your edit. You can add an attribute by clicking in the blank Name cell at the bottom of the table and entering a new attribute name. Alternatively, you can use the + button to add a new row. You can give it the Type string by clicking the drop-down menu under Type and selecting string.

Once you have made these changes to the writer schema, the attribute list should now look like this:
Now click OK.

You might notice that the triangles next to the attribute names we edited and created on the writer feature type have changed color to red! We call these triangles **ports**. When they are on the left side of an object, they are called **input ports**, while triangles on the right side are called **output ports**. You can notice the attributes we edited have changed color to yellow.

**TIP**

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.
4) Save the Workspace

Save the workspace. It should now look like this:

![Diagram of workspace]

5) Add a Transformer for Schema Mapping

Let's use a transformer to map our old schema onto our new one. Transformers are objects we add to the canvas to modify our data during the translation. We can use the AttributeManager.

To add a transformer, click on the feature connection (dark black line) from reader to writer feature type:

![Diagram of feature connection]

Start to type the phrase “AttributeManager.” As you type, FME searches for a matching transformer, a feature called Quick Add. When the list is short enough for you to see the AttributeManager, select it from the dialog (double-click on it, or hit Enter):
Doing so will place an AttributeManager transformer:

6) Set Parameters

View the AttributeManager parameters by double-clicking it. It will look like this:

**TIP**

For a great tip on adding transformers, see #5 in our list of The Top Ten FME Tips of All Time!
You might have to resize your AttributeManager dialog to see all four columns. You can resize dialogs by hovering over the bottom right corner until you see an expander cursor, and then clicking and dragging.

Notice that all of the attributes on the stream in which it is connected automatically appear in the dialog.

Where the Input Attribute field is `Case_Type`, click in the Output Attribute field. Click on the button for the drop-down list and in there choose `Case_Type` as the new attribute name to use:
In response, the Action field will change to read Rename.

**TIP**

"Case Type" appears in the list because it already exists on the writer feature type. If we had done this step before editing the writer schema, we would have had to enter the new attribute name in this dialog manually. Note that this feature (looking ahead to the attributes on the writer feature type) is only needed in some related transformers, e.g. AttributeRenamer and AttributeCopier. Normally drop-down attribute lists contain just the attributes on incoming features.

Click OK to close the dialog. Now in the Workbench canvas window, you will see the Case Type attribute is flagged with a green arrow, to confirm that the attribute has a value.
Reopen the AttributeManager dialog and repeat this step for the remaining attributes with underscores.

8) Create a **Date** Attribute

We also want to add a new attribute, **date**. We could use several different transformers to accomplish this; again, there are often many solutions to a problem in FME. For example, we could use the ExpressionEvaluator or StringConcatenator. In this case, let's keep our workspace efficient by creating the attribute within the AttributeManager.

Go to the bottom of the Attribute Actions table and notice the last row is empty except for the text `<Add new Attribute>`. Click this text and type in **Date**. You might see it appears as an option in the drop-down menu; FME will look ahead to your writer schema to help you find attributes. After you type in or select **Date**, the Action cell changes to **Set Value**:

![AttributeManager Parameters](image)

After clicking OK, you should see the input ports on the writer feature type all turn green. Note that we do not have a **Date** attribute in our source data; therefore, we will not be mapping that attribute yet and its port will remain red.

9) Run Workspace and Inspect the Output

Save the workspace.

Then, let's use the feature caching and partial runs. Click on the AttributeManager to select it, and then click Run To This:
Once the cache updates, click the green icon on AttributeManager to inspect the changes to your schema.

Looking at the Table View, we can see our schema has been edited to remove the underscores from attribute names and to add a Date attribute (which is empty for now):

<table>
<thead>
<tr>
<th>Case Type</th>
<th>Hundred Block</th>
<th>Street Name</th>
<th>Local Area</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pothole - Repair</td>
<td>13##</td>
<td>E 60TH AV</td>
<td>Sunset</td>
<td></td>
</tr>
<tr>
<td>Pothole - Repair</td>
<td>##</td>
<td>PS NO MATCH ST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water Leaks/Breach</td>
<td>34##</td>
<td>WORTHINGTON...</td>
<td>Renfrew-Colling...</td>
<td></td>
</tr>
<tr>
<td>Streets - General</td>
<td>Intersection</td>
<td>OAK ST and W ...</td>
<td>Fairview</td>
<td></td>
</tr>
</tbody>
</table>

CONGRATULATIONS

By completing this exercise, you have learned how to:

- Edit the attributes of a writer schema
- Edit the output layer name on a writer schema
- Add transformers to a workspace
- Carry out schema mapping with the AttributeManager transformer
FME Translations Quiz

Which of the following statements is true about the Generate Workspace dialog?

You must define both the reader and writer format. You must define the reader format, but writer format is optional. The reader format is optional, but you must define the writer format. Both the reader and writer formats are optional. Generate Workspace requires both a reader and writer format. You can use the Format field blank (it will show <Guess from Dataset>) and FME will try to guess your format. However, you must still eventually supply a value to this field. In more advanced training modules, you can learn how to set a Generic reader or writer, which can use any format FME supports.

The FME Data Inspector is a fully-functioned spatial data analysis and cartography tool.

True False The FME Data Inspector is not designed to replace GIS software, but rather to provide a quick and useful way to inspect the result of FME workspaces.

Which of the following scenarios would be well-suited to using feature caching? Check all that apply.

Reading from a large database Reading from a large web dataset Running a production workspace Running a simple workspace with a Creator and a single Emitter to send an email Using partial runs to incrementally develop a workspace with a complicated workflow Workspaces that read large datasets or data that is slow to access, including databases or data on a network, can benefit from feature caching. Read the data once to cache it and then use Run From This or Run To This.

The initial process of feature caching takes longer than running the workspace without feature caching on, so it is not a good idea to keep feature caching on with a production workspace.

A very simple workspace with only one or two transformers, neither of which produce many features, will not benefit from feature caching.

Using partial runs with feature caching is a great way to quickly build and test sections of your workspace.

Which of the following is an example of schema editing or mapping? Check all that apply.

You create a workspace that removes observations from a CSV file if they are older than a target date. You create a workspace that joins customer data from a CRM to social media exposure data gathered by your Social Media Manager. You create a workspace that adds updated AutoCAD drawings to a centralized Esri ArcGIS geodatabase, removing three unwanted attributes and renaming two others in the process. You create a workspace that extracts a group of records from an Oracle database based on a spatial filter. Removing records (by an attribute value like observation date or spatial location) does not count as schema editing or mapping. Although the content of the written data is changed, its structure remains the same. In the CRM example, joining data adds new attributes. In the AutoCAD example, attributes are removed or renamed. These operations are schema editing, because you are changing the structure of the final dataset.

Why is the input port on the Date attribute of the writer feature type red?

The feature type's User Attributes tab did not appropriately define attribute parameters. FME is not connected to system date-time. The Date attribute on the reader feature type is not exposed. There is a Date attribute defined on the writer schema, but it has no value. We created a Date attribute on the writer schema by adding it to the User Attributes tab. However, there is nowhere in the workspace - either in the read data or data created with transformers - to supply values to this attribute. Therefore, it displays with a red color.
FME Transformations

45 minutes

Learning Objectives

After completing this unit, you will be able to:

- Explain what transformers do in FME.
- Discuss common transformer categories.
- Locate and place transformers using Quick Add.
- Set transformer parameters.

Data Transformation

You have already learned how to *translate* data in FME. In this unit, you will learn how to transform data to suit your needs.

What is Data Transformation?

*Data Transformation* is FME's ability to manipulate data. Whenever you analyze, restructure, alter, or edit data, you are transforming it. The transformation step occurs during the process of format translation. Data is read, transformed, and then written to the chosen format.
Data Transformation Possibilities

Data transformation can alter the structure or content of data, or both together. Transforming the **structure** of the data could be called ‘reorganization.’ This process includes the ability to merge data (as in the image above), divide data, re-order data, and define custom data structures. Transforming the structure of a dataset is carried out by manipulating its schema.

Transforming the **content** of the data could be called ‘revision.’ Manipulating a feature's geometry or calculating new attribute values is the best example of how FME can transform content.

As a whole, transformation lets you take the data you have and turn it into the data you want.
Transformers

Besides Schema Editing and Schema Mapping, transformation can be carried out using objects called transformers.

Transformers

As the name suggests, a transformer is an FME Workbench object that carries out the transformation of features. There are lots of FME transformers, each of which carries out a different operation.

Transformers are connected somewhere between the reader and writer feature types, so that data flows from the reader, through a transformation process, and on to the writer.

Transformers appear in the canvas window as rectangular, light-blue objects.

Transformer Parameters

Each transformer may have many parameters (settings). Parameters can be accessed (like feature types) by clicking the cogwheel icon:

![Transformer Parameters](image)

Alternatively, if the Parameter Editor window is open, parameters can be found there by clicking on the transformer (or any other canvas object):

![Parameter Editor](image)

Color-Coded Parameter Buttons

The parameter button on a transformer is color-coded to reflect the status of the settings.

A blue parameter button indicates that the transformer parameters were checked and amended as required and that the transformer is ready to use.
A yellow parameter button indicates that the default parameters have not yet been checked.

A red parameter button indicates that there is at least one parameter for which FME cannot supply a default value. The parameter must be provided with a value before using the transformer.

Transformer Ports

Far from having just a single input and output, a transformer can have multiple input ports, multiple output ports, or both. This 2DForcer transformer has a single input and output port.

This Clipper has multiple input and output ports. Notice that not all of them are – or need to be – connected.

This Inspector has just a single input port...

...whereas this Creator has only a single output port!

Transformer Attributes

Click on the drop-down arrow of a transformer output port to see all of the attributes that exit the transformer. This list includes all changes applied within the transformer.
This feature lets one visualize which attributes have been created, lost, or otherwise transformed within the transformer.
We know from the last unit that we have missing values in our data. We can use a filtering transformer to remove features with missing values.

1) Start Workbench

Start Workbench (if necessary) and open the workspace from Exercise 2.4. Alternatively you can open C:\FMEData2018\Workspaces\IntroToDesktop\Ex3.1-Begin.fmw.

2) Add Tester Transformer

Click a blank spot on the canvas. Start to type the phrase “Tester.” When the list is short enough for you to see the Tester, select it from the dialog (double-click on it):

Doing so will place a Tester transformer. Click and drag the Tester to move it near the feature connection line between the reader feature type and the AttributeManager. As you do so, notice that a small green arrow appears on the left of the Tester. Drag until this arrow is over the feature connection line:
Once the feature connection line is highlighted, let go of your click to connect the Tester:

3) Set Parameters

Inspect the parameters for the Tester transformer (either its dialog or in the Parameter Editor window). Click in the Left Value field and from there click the down arrow and choose Attribute Value > Local Area:
For the Operator field click the cell and select "Attribute Has a Value." We do not need to fill in the Right Value field for this operator. Your Test Clauses table should look like this:
There might be missing values in other attributes, but we are only concerned with cases where Local_Area is missing, so we will only fill in one test clause.

Click OK to accept the values and close the dialog.

**TIP**

You could also accomplish this procedure with the NullAttributeValueMapper. In FME there is usually more than one way to solve your problem.

**4) Run the Workspace using Feature Caching**

Now that you know how to use feature caching, we can use it to test our new transformer. Select the Tester transformer. Then click the "Run To This" icon (note that the objects that will run are highlighted in green):
Your workspace will run. If you set up the Tester properly, it should do so without errors. If you pay attention while the translation is running, you can see the feature counts increasing as features flow through the workspace. By the time the translation is complete, 4,122 features will be filtered out to the Tester's Failed port. These features did not have values for Local_Area and are removed from the rest of the translation.

5) Explore the Text Editor

In Exercise 2.4 we added a Date attribute, but it does not have any values yet.

You can enter in the value for your Date attribute in the Attribute Value column. If you were to type in "January 1st, 2000," all of the features would receive that value as a constant. However, what we want to do is create a date attribute that combines (or "concatenates") our existing date-time fields: Year, Month, Day, Hour, and Minute.

We can do that using FME's Text Editor, which allows us to construct an expression to generate the value for our attribute. To open it, click the Attribute Value cell, and then click the drop-down arrow that appears. You will see "Open Text Editor" and "Open Arithmetic Editor" as options. Click "Open Text Editor":
The text editor - as you would expect - allows you to construct a text value. It includes all the usual string-handling functionality you would need, such as concatenation, trimming, padding, and case changing.

The text editor looks like this:

Notice the menu on the left-hand side. Existing attributes are listed here and were added into the string by double-clicking them. Also, notice the other menu options. The essential functions for text are the String Functions:
The Arithmetic Editor contains similar functions, as well as mathematical operators to calculate values for attributes.

6) Create Values for Date Attribute

Let's combine existing attributes to give date some values. In the left-hand panel under “FME Feature Attributes,” you will see all the incoming attributes listed. Double-click the attribute Year. Doing so adds @Value(Year) to the Text Editor window. When the translation runs, this code will take the value of Year for each feature. We can combine multiple attribute values to construct the date. In this case, we will create a basic year/month/day attribute, but you could use the Text Editor to format your date however you like.

Type in a forward-slash / and then double-click the attribute Month, type in another /, and then double-click the attribute Day. You should see this in the Text Editor:

Alternatively, you can copy and paste the following into the Text Editor:

@Value(Year)/@Value(Month)/@Value(Day)

Click OK and OK again. Save your workspace. Then, use Run To This on the AttributeManager.
TIP

Many FME Text and Arithmetic Editor functions come from the Tcl programming language. If you do not want to type out code, most results can also be accomplished using transformers. For more information, refer to the Documentation.

7) Inspect the Data

Inspect the AttributeManager feature cache and examine the Table View window. You can see that the underscores have been removed from attribute names and the date attribute now correctly displays the full date/time in FME standard format:

<table>
<thead>
<tr>
<th>Case Type</th>
<th>Hundred Block</th>
<th>Street Name</th>
<th>Local Area</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pothole - Repair</td>
<td>13##</td>
<td>E 60TH AV</td>
<td>Sunset</td>
<td>2017/1/1</td>
</tr>
<tr>
<td>Water Leaks/Bre...</td>
<td>34##</td>
<td>WORTHINGTON...</td>
<td>Renfrew-Colling...</td>
<td>2017/1/1</td>
</tr>
<tr>
<td>Streets - General...</td>
<td>Intersection</td>
<td>OAK ST and W ...</td>
<td>Fairview</td>
<td>2017/1/1</td>
</tr>
<tr>
<td>Sign - Repair</td>
<td>12##</td>
<td>W 10TH AV</td>
<td>Fairview</td>
<td>2017/1/1</td>
</tr>
</tbody>
</table>

Advanced Exercise

If you want to use a more realistic - but also more complicated - example, you can generate a standard FME date/time stamp. To do so, copy and paste the following code into the Text Editor:

```
@TimeZoneSet(@Format(%04d,@Value(Year))@Format(%02d,@Value(Month))@Format(%02d,@Value(Day ))@Format(<%02d>,@Value(Hour))@Format(%02d,@Value(Minute))00,, local)
```

The details aren’t important here, but if you want them, read on. In short, we are using the @TimeZoneSet function to add a UTC offset to each date/time stamp. We are forming the date/time stamp by combining the attributes from our source data into a string matching FME date/time format: yyyyymmddhhss-UTC. Our original data had single digit months, days, hours, and minutes missing the leading zero, so we use the @Format function to fix that problem. This new attribute will be easier to use in FME workflows.

CONGRATULATIONS

By completing this exercise, you have learned how to:
- Filter data using the Tester transformer
- Use transformer parameters to create attributes that match the writer schema
- Construct an attribute value using the Text Editor
Locating Transformers

Even experienced FME users find the full list of transformers a daunting sight. With over five hundred (500) transformers, FME possesses a lot of functionality; probably a lot more than a new user realizes and much of which would be very useful to them. This section helps you find the transformer you need, even if you didn’t know you needed it.

Although the transformer list can look a bit overwhelming, do not panic! The reality is that most users focus on 20-30 transformers that are relevant to their day-to-day workflow. You do not need to know every single transformer to use FME effectively.

Quick Add

One of the easiest ways to look for the transformer you need is to click a blank part of the canvas and type relevant keywords. This action will open the Quick Add menu, allowing you to search for transformers.

Search terms can also be full or partial words:

![Quick Add Menu](image)

By default, Quick Add does not look in transformer descriptions, so the search term must be the actual name of a transformer:

![No Results](image)

However, Quick Add will search in the transformer descriptions if you press the Tab key with the dialog open:
Quick Add results include aliases - for example, alternate names or renamed transformers - and also include transformers found in the FME Hub:

Finally, you can use CamelCase to search for transformer names, which is particularly useful for transformers with long names, e.g., "rcvr" for the RasterCellValueRounder:

Transformer Gallery

You can also look for transformers in the transformer gallery, which exists both as a window in Workbench and online. There are several ways to locate transformers using the Gallery.

Transformer Categories

Transformer categories are a good starting point from which to explore the transformer list. Transformers are grouped in categories to help find a transformer relevant to the problem at hand.
Although all of them are important, the most commonly used transformers are in these categories:

- **Attributes**: Operations for attribute/list management
- **Calculated Values**: Operations that return a calculated value
- **Filters and Joins**: Operations for dividing and merging data flows
- **Geometries**: Operations that create geometry or transform it to a different geometry type
- **Spatial Analysis**: Operations that return the result of a spatial analysis
- **Strings**: Operations that manipulate string contents, including dates

Click on the expand button to show all transformers within a particular category.

**Sorting Transformers**

The online Transformer Gallery has the added benefit of allowing one to sort transformers by their popularity (Sort By > Most Used):

It also allows one to view related transformers. Clicking on a transformer takes you to its page. Underneath the transformer description is a related transformers section:
The FME Hub

The FME Hub is a facility for sharing FME functionality such as custom transformers, web connections, and formats:

Transformers from the hub are shown in Quick Add with a green color and a small, downwards-pointing arrow, to denote that they will be downloaded if selected.

TIP

The FME Workbench Help tool displays information about transformers. Click on a transformer and press the F1 key to open the help dialog. This tool is linked to FME Workbench so that a transformer selected (in the gallery or on the canvas) triggers content to display in the Help tool.
Another useful - and printable - piece of documentation is the *FME Transformer Reference Guide*.
### Exercise 3.2 Locating Transformers

<table>
<thead>
<tr>
<th>Data</th>
<th>3-1-1 case location details (XLS hosted on FTP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Goal</td>
<td>Find a transformer to fix inconsistent data</td>
</tr>
<tr>
<td>Demonstrates</td>
<td>Using Transformer Gallery and Quick Add</td>
</tr>
<tr>
<td>Start Workspace</td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex3.2-Begin.fmw</td>
</tr>
<tr>
<td>End Workspace</td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex3.2-Complete.fmw</td>
</tr>
</tbody>
</table>

In Exercise 2.2, we found that some of our values for the `Local Area` attribute were inconsistent. In this exercise, we will find an appropriate transformer to address this problem. As we just covered, there are many options for searching for transformers. In this case, we need a transformer that can **find** specific text **strings** in an **attribute** and **replace** them with another value. Specifically, we want to find values of `Local Area` and replace them as such:

<table>
<thead>
<tr>
<th>Find</th>
<th>Replace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dunbar Southlands</td>
<td>Dunbar-Southlands</td>
</tr>
<tr>
<td>Arbutus Ridge</td>
<td>Arbutus-Ridge</td>
</tr>
</tbody>
</table>

We could use any of the resources available (Transformer Gallery in Workbench, Transformer Gallery online, the FME Transformer Reference Guide, and Quick Add search) to find such a transformer.

#### 1) Examine Transformer Gallery Categories

Go to the FME Transformer Gallery and click the category drop-down menu:

![Transformer Gallery](https://www.safe.com/transformers/)

Which category do you think we might need to look in? We are trying to modify the values of an **attribute**, so it makes sense to start our search in the **Attributes** category. Select Attribute from the drop-down to filter by these transformers.
Well, that's a good start, but there are still 30 transformers in this category. We could go through one-by-one and read each description until we find the right one, but that would take too long. Instead, we should search for some keywords that describe what we want to do.

Try searching for these terms:

- Find
- Replace

These terms don't yield any transformers in this category. Maybe we should try a different approach.

2) Try a Different Category

Is it possible that our transformer might be in a different category? Return to the drop-down and take a look. There is a **Strings** category! That could work, since we are trying to modify a text (i.e., string) value. Select the **Strings** category.

This category has even more transformers in it: 43! But let's try searching transformers in this category with our keywords again:

- Find
- Replace

"Find" results in the **ListSearcher**, which searches lists (a special kind of attribute value), not attributes, so that won't work.

"Replace" results in three transformers:

- **StringReplacer** - Replaces substrings matching a string or regular expression in the string contained in the source attribute.
- **StringPairReplacer** - Replaces characters in the value contained in the source attribute based on replacement key-value pairs.
- **CharacterCodeReplacer** - Sets the result attribute to the character whose numeric code was contained in the source code attribute (or the entered integer).

It sounds like either of the first two will work! As an experienced FME user, I know that either will work, but the **StringReplacer** will be a bit more straight-forward, so let's use that transformer.

**TIP**

There are often multiple solutions to a given problem in FME. We could accomplish the same result by using:

- An **AttributeValueMapper**
- An **AttributeManager** with **Conditional Values**
- An **AttributeCreator** with **Conditional Values**
- A **TestFilter** and **AttributeCreators**
- **Testers** and **AttributeCreators**
- **AttributeValidators** and **AttributeCreators**
- A **StringPairReplacer**

Did you find another way? Congratulations! Please let us know at train@safe.com.

3) Start Workbench

Now, let's add the StringReplacer to our workspace.
Start Workbench (if necessary) and open the workspace from Exercise 3.1. Alternatively you can open C:\FMEData2018\Workspaces\IntroToDesktop\Ex3.2-Begin.fmw.

4) Add a StringReplacer

Add a StringReplacer to the canvas between the AttributeManager and the writer feature type by clicking the feature connection line and typing in StringReplacer to view it in Quick Add, then hitting enter:

Once it is added, double-click it to open its parameters. This transformer requires you to choose an attribute to modify, and then supply a string for Text to Replace (i.e., find) and Text to Replace (i.e., replace). You can also specify what happens to values that don't have a match, which defaults to <No Action>. The source values in the incoming data are replaced by any matching destination values, or otherwise receive a default value.

As you can see, because we can only enter a single find and a single replace value, we will have to use two StringReplacers, one for each data problem.

Fill in the parameters like this:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Local Area</td>
</tr>
<tr>
<td>Text To Replace</td>
<td>Arbutus Ridge</td>
</tr>
<tr>
<td>Replacement Text</td>
<td>Arbutus-Ridge</td>
</tr>
</tbody>
</table>

We will replace any values with a space to those with a dash - this fits the City of Vancouver specification.

Your dialog should look like this:
Click OK.

5) **Add Second StringReplacer**

We can add another StringReplacer by right-clicking our first and clicking Duplicate or using the shortcut Ctrl + D:
Now we need to connect it to the data flow. Click and drag your second StringReplacer; you will notice a small green arrow in the top left corner. Move this arrow over the feature connection line between the first StringReplacer and the writer feature type. You will see the feature connection line is highlighted. Release your click to attach the transformer at that point:
Double-click the second transformer to open its parameters. We can leave some of the parameters from our first transformer, but make sure you change the following:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text To Replace</td>
<td>Dunbar Southlands</td>
</tr>
<tr>
<td>Replacement Text</td>
<td>Dunbar-Southlands</td>
</tr>
</tbody>
</table>

Once you have filled those in, click OK.

### 6) Inspect Result

Now let’s inspect our data to confirm we’ve fixed the problem. Click the second StringReplacer to select it, and then click Run To This:

Once the cache is loaded, click the green icon on the second StringReplacer to open it in Data Inspector. We can use the same procedure from Exercise 2.3 of searching our Table View to confirm our data has changed. Type our original values (“Arbutus Ridge” and “Dunbar Southlands”) into the search bar in the bottom left of the Table View to search for them. They should both yield zero results, whereas if you search for the version with the dash, you should get 2,236 and 4,062 results, respectively:

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Results</th>
<th>Filtered / Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arbutus Ridge</td>
<td></td>
<td>0 filtered / 93223 row(s)</td>
</tr>
<tr>
<td>Arbutus-Ridge</td>
<td></td>
<td>2,236 filtered / 93223 row(s)</td>
</tr>
<tr>
<td>Dunbar Southlands</td>
<td></td>
<td>0 filtered / 93223 row(s)</td>
</tr>
<tr>
<td>Dunbar-Southlands</td>
<td></td>
<td>4,062 filtered / 93223 row(s)</td>
</tr>
</tbody>
</table>

**CONGRATULATIONS**

By completing this exercise, you have learned how to:
- Search for a transformer using the online Transformer Gallery
- Use a StringReplacer to fix inconsistent data values
Common Transformations

Most Valuable Transformers

If you have a thorough understanding of the most common transformers, then you have a good chance of being a very efficient user of FME Workbench.

Anyone can be proficient in FME using only a handful of transformers if they are the right ones!

The Top 30

The list of transformers on the Safe Software website is ordered by most-used, calculated from user feedback. Having this information tells us where to direct our development efforts in making improvements, but it also gives users a head-start on knowing which of the (500+) FME transformers they’re most likely to need in their work.

The following table (last updated October 2018) provides the list of the most commonly used 30 transformers. The Tester transformer is consistently number one in the list every year, highlighting its importance.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Transformer</th>
<th>Rank</th>
<th>Transformer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tester</td>
<td>16</td>
<td>DuplicateFilter</td>
</tr>
<tr>
<td>2</td>
<td>AttributeCreator</td>
<td>17</td>
<td>FeatureReader</td>
</tr>
<tr>
<td>3</td>
<td>AttributeManager</td>
<td>18</td>
<td>StringReplacer</td>
</tr>
<tr>
<td>4</td>
<td>FeatureMerger</td>
<td>19</td>
<td>VertexCreator</td>
</tr>
<tr>
<td>5</td>
<td>Inspector</td>
<td>20</td>
<td>StatisticsCalculator</td>
</tr>
<tr>
<td>6</td>
<td>AttributeKeeper</td>
<td>21</td>
<td>SpatialFilter</td>
</tr>
<tr>
<td>7</td>
<td>TestFilter</td>
<td>22</td>
<td>Sorter</td>
</tr>
<tr>
<td>8</td>
<td>Creator</td>
<td>23</td>
<td>AttributeExposer</td>
</tr>
<tr>
<td>9</td>
<td>AttributeRenamer</td>
<td>24</td>
<td>Bufferer</td>
</tr>
<tr>
<td>10</td>
<td>Reprojector</td>
<td>25</td>
<td>Dissolver</td>
</tr>
<tr>
<td>11</td>
<td>Aggregator</td>
<td>26</td>
<td>GeometryFilter</td>
</tr>
<tr>
<td>12</td>
<td>AttributeRemover</td>
<td>27</td>
<td>ListExploder</td>
</tr>
<tr>
<td>13</td>
<td>AttributeFilter</td>
<td>28</td>
<td>FeatureJoiner</td>
</tr>
<tr>
<td>14</td>
<td>Clipper</td>
<td>29</td>
<td>AttributeSplitter</td>
</tr>
<tr>
<td>15</td>
<td>Counter</td>
<td>30</td>
<td>CoordinateExtractor</td>
</tr>
</tbody>
</table>

Categories

Besides the obvious transformers for transforming geometry (Clipper, Bufferer, Dissolver) and the obvious transformers for transforming attribute values (StringReplacer, Counter) there are some other distinct groups of transformers.

Managing Attributes

These transformers - mostly named the Attribute<Something> - are primarily for managing attributes (creating, renaming, and deleting) for schema mapping purposes. However, they can also be used to set new attribute values or update existing ones.

Filtering

These transformers - mostly named the <Something>Filter - subdivide data as it flows through a workspace. Commonly the filter is a conditional filter, where features are output according to the results of a test or condition.

Data Joins
Joins are the opposite action to filtering; they are when separate streams of data are combined as they flow through a workspace. Like filtering, there is a condition to be met - in this case matching key values - that determines how and where the join takes place.

**FME is Broad**

FME has an extensive scope. It can connect to hundreds of formats and has hundreds of transformers. Some users only use one format and a handful of transformers, while others use many more. All of these are valid ways to use FME. Do not be worried if you feel overwhelmed at first; many users do! Through completing training, you will learn about a few essential transformers. After that, we have a plethora of resources to help you learn about more formats and transformations:

- Free training
- Knowledge Base Tutorials
- Q&A Forum
- Webinars, demos, a blog and more!
Exercise 3.3a

<table>
<thead>
<tr>
<th>Data</th>
<th>3-1-1 case location details (XLS hosted on FTP)</th>
</tr>
</thead>
</table>
| Overall Goal | Add a transformer to clean up schema
Create a summary table |
| Demonstrates | Common transformer procedures |
| Start Workspace | C:\FMEMgdata2018\Workspaces\IntroToDesktop\Ex3.3-Begin.fmw |
| End Workspace | C:\FMEMgdata2018\Workspaces\IntroToDesktop\Ex3.3-Complete.fmw |

Your manager has some more requests for your workspace. The department wants to be able to provide a summary of the number of cases by department, by Local Planning Area, in addition to the original data with an edited schema. This summary table will help the city assess demand for different services in different areas of the city. In this exercise, we will use a transformer to generate a summary table.

Before we go through the steps for doing so, try to find which transformer to use!

Use the resources available (Transformer Gallery in Workbench, Transformer Gallery online, the FME Transformer Reference Guide, and Quick Add search) to find such a transformer. Then we'll go through the steps to add it.

OK, go for it! When you have found your solution, go to the next page to check your answer.

If you want some additional hints, click below:

▼ Hint 1
The transformer is in the Calculated Values category of the Transformer Gallery.

▼ Hint 2
The transformer you are looking for has Calculator in the name.

▼ Hint 3
To create a summary table, you need to calculate the sum of cases. The sum is a statistic of your attributes.

▼ Hint 4
You might have found the AttributePivoter or the SummaryReporter; these transformers might be used in this case, but we are actually looking for a different one that works better.

Additional Resources

If you have used the above resources and are still having trouble, try looking on the FME Knowledge Center.
Exercise 3.3b | Using Common Transformers

<table>
<thead>
<tr>
<th>Data</th>
<th>3-1-1 case location details (XLS hosted on FTP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Goal</td>
<td>Add a transformer to clean up schema</td>
</tr>
<tr>
<td></td>
<td>Create a summary table</td>
</tr>
<tr>
<td>Demonstrates</td>
<td>Common transformer procedures</td>
</tr>
<tr>
<td>Start Workspace</td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex3.3-Begin.fmw</td>
</tr>
<tr>
<td>End Workspace</td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex3.3-Complete.fmw</td>
</tr>
</tbody>
</table>

Did you find a transformer that works? The easiest one to use is the **StatisticsCalculator**. Let's add it to our workspace.

---

**TIP**

*Again, there are often multiple solutions to a given problem in FME. We could accomplish roughly the same result by using:*

- An **AttributePivoter**
- An **Aggregator** and a **ListHistogrammer**
- Some overly-complicated combination of **AttributeCreators**, **AttributeFilters**, **Counters**, and **Sorters**

1) **Start Workbench**

Start Workbench (if necessary) and open the workspace from Exercise 3.2. Alternatively you can open C:\FMEData2018\Workspaces\IntroToDesktop\Ex3.3-Begin.fmw.

2) **Add a StatisticsCalculator**

Click on blank canvas, then type "StatisticsCalculator" and hit **Enter** when it appears on the Quick Add menu to add it. Click and drag from the Output port of StringReplacer_2 to the Input port of StatisticsCalculator. Then, click and drag the StatisticsCalculator to place it between and slightly above your StringReplacer_2 and writer feature type.

Your workspace should look like this:
We aren't going to connect the StatisticsCalculator to our feature type just yet; we'll discuss why in step 5 below.

3) Set StatisticsCalculator Parameters

Double-click the StatisticsCalculator to open its parameters. Fill them in like this:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group By</td>
<td>Department and Local Area</td>
</tr>
<tr>
<td>Attributes to Analyze</td>
<td>Case Type</td>
</tr>
<tr>
<td>Total Count Attribute</td>
<td>Cases</td>
</tr>
</tbody>
</table>

Additionally, make sure you remove the values from all the other Calculated Attributes (Minimum Attribute, Maximum Attribute, etc.). You can do this by selecting the parameter and deleting it with the Delete or Backspace key, or by clicking the drop-down arrow and selecting Clear Value:

When finished, your parameters should look like this:
Click OK and use Run To This on the StatisticsCalculator type to summarize your data. With these parameters, the transformer will add an attribute, Cases, to your data, which sums count of Case Type. By setting Group By to Department and Local Area, we get the total count for each unique combination of department and local area.

4) Inspect the Cache

Let's inspect the cache to ensure our data looks correct. Click on the green inspect cache icon on the Summary output port of the StatisticsCalculator. The StatisticsCalculator illustrates that transformers vary in the number of input and output ports they have. The Summary port outputs a summary table of the results, resulting in a new stream of features (here, 329), while the Complete and Cumulative ports add the results of the summary to every incoming feature (here, 93,223).

In the Data Inspector, you should see this in the Table View:
This result is a good start, but now that our data's schema has changed, we have different attributes than those on our existing writer feature type. We can add a new writer feature type to write the summary table to. We'll do that in the next section.

CONGRATULATIONS

By completing this exercise, you have learned how to:
- Use the StatisticsCalculator to generate a summary table.
- Use group-by to group features in a transformer.
- Observe that some transformers have multiple output ports creating different streams of data.
FME Transformations Quiz

Which transformer below is in the top ten most-used transformers in the Web category?

AttributeCreator AttributeKeeper AttributeManager Creator Generalizer Tester TestFilter Going to the online Transformer Gallery allows one to view only transformers in the Web category and to sort by most used.

Match the string and the transformer returned by Quick Add:

lineco
Quick Add will search transformer names for whatever you type in, including if you skip beginning characters. Some users prefer to type the middle of transformer names to find them faster, for example, “reac” for AreaCalculator, or to use CamelCase, e.g., “rbic” for RasterBandInterpretationCoercer.

rea
Quick Add will search transformer names for whatever you type in, including if you skip beginning characters. Some users prefer to type the middle of transformer names to find them faster, for example, “reac” for AreaCalculator, or to use CamelCase, e.g., “rbic” for RasterBandInterpretationCoercer.

rbic
Quick Add will search transformer names for whatever you type in, including if you skip beginning characters. Some users prefer to type the middle of transformer names to find them faster, for example, “reac” for AreaCalculator, or to use CamelCase, e.g., “rbic” for RasterBandInterpretationCoercer.

drape
Quick Add will search transformer names for whatever you type in, including if you skip beginning characters. Some users prefer to type the middle of transformer names to find them faster, for example, “reac” for AreaCalculator, or to use CamelCase, e.g., “rbic” for RasterBandInterpretationCoercer.

attributeexpl
Quick Add will search transformer names for whatever you type in, including if you skip beginning characters. Some users prefer to type the middle of transformer names to find them faster, for example, “reac” for AreaCalculator, or to use CamelCase, e.g., “rbic” for RasterBandInterpretationCoercer.

Your colleagues have come up with a list of different scenarios, and want you to search for a transformer to carry them out. Select the correct transformer for each case.

We have some lines of text in a file and want to read that text and add it as an attribute.
From FME Help for AttributeFileReader: "Reads the contents of a file and stores them as the value for the specified attribute.”
You want to control the font size and color of your tables in an Excel workbook output.
From FME Help for ExcelStyler: "Sets Excel row and cell styling using attributes on output features destined for the Excel Writer.”
You have a list of customers, including their company name and industry, and want to create a table summarizing the number of customer companies by industry.
From FME Help for AttributePivoter: "Restructures and regroups incoming features based on specified Group by attributes and calculates summary statistics to form a Pivot table output.” The AttributePivoter can accomplish many features of Excel's Pivot Tables. Similar restructuring can also be accomplished with the StatisticsCalculator.
We have a text string and want to find out how many characters the string contains.
From FME Help for StringLengthCalculator: "Calculates the length of strings and the number of bytes in a blob.”
We have a set of addresses and for each address want to find the geographically closest two libraries.
From FME Help for NeighborPairFinder: "Finds the closest two Candidate features within some maximum distance of each Base feature and some minimum separation in heading between the Candidates and the Base.”
FME Workflows

30 minutes

Learning Objectives

After completing this unit, you will be able to:

- Understand how data flows through a workspace.
- Create a workspace with multiple formats.
- Create a workspace with multiple feature types.
- Employ basic best practice techniques (bookmarks and annotations) in building your workspaces.

Workflows

A basic workspace in FME reads in data in one format, transforms it, and writes it out, optionally in a different format.

However, there are multiple other methods for constructing more advanced workspaces and for directing the flow of data through a workspace in unique ways.

Some example uses for these techniques might be:

- To branch your data into multiple streams
- To design large-scale workspaces a small section at a time
- To read data from multiple formats within a single workspace
- To carry out actions after a dataset is written
- To use data stored on web services
- To test run individual parts of a workspace

In this unit we will cover some of these different FME workflows, illustrating the flexibility of FME for data transformation.
**The Workflow**

**Transformers in Series**

Much like a set of components in an electrical circuit, a series of Workbench transformers can be connected to have a cumulative effect on a set of features.

Even with the large number of transformers available in FME, users frequently need a combination - or chain of transformers - instead of a single one.

A string of transformers that graphically represent an overall workflow is a crucial concept of FME:

![Diagram of transformers in series](image)

In this example, a DuplicateFilter transformer removes duplicate polygon features. A Dissolver transformer merges each remaining (unique) polygon with its neighbor where there is a common boundary. Finally, each merged area gains an ID number from the Counter transformer.

**Transformers in Parallel**

A *stream* is a flow of data represented by connections in the workspace. A key concept in FME is the ability to have multiple parallel streams within a workspace.

**Multiple Streams**

Multiple streams are useful when a user needs to process the same data but in many different ways. A workspace author can turn one stream into several, or combine several streams of data into one, as required:

![Diagram of multiple streams](image)

Here an author is creating three data streams, each of which is processed separately then combined back into a single stream.

**Creating Multiple Streams**

Creating multiple data streams can occur in several ways. Sometimes a transformer with multiple output ports (a Tester transformer is a good illustration of this) will divide (or filter) data with several possible output streams:

![Diagram of Tester transformer](image)

Here data is divided into two streams, one of which is not connected to anything.

Additionally, a full stream of data can be duplicated by simply making multiple connections out of a single output port. This method creates a set of data for each connection:
Here FME reads eight features but, because there are multiple connections, creates multiple copies of the data.

**Bringing Together Multiple Streams**

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

Here, three streams of data converge into a writer feature type:

The data accumulates into 12 distinct features in the output dataset.

If you wanted to combine or merge these features in some way, you need to carry out a **join**, also known as an **intersection**. This requires that the feature share a common attribute, or key, upon which to base the join. For example, if you had a table with neighborhood demographic data and points of libraries (including an attribute of which neighborhood they were in), you could join the table of demographic data to the libraries based on their shared neighborhood name attribute. Then your library points would have neighborhood demographic data as attributes.

To carry out this joining of data requires a specific transformer such as the **FeatureMerger** or **FeatureJoiner**.
Multiple Readers and Writers

An FME workspace is not limited to any particular number of readers or writers; it can have any number of readers and writers, use any number of formats, and does not need to contain an equal number of readers and writers.

For example, the Navigator window shows this workspace contains two readers and three writers, of different data types and formats!

![Navigator window showing readers and writers](image)

**TIP**

It's important to note that readers and writers don't appear as objects on the Workbench canvas. Their feature types do, but readers and writers do not.

Instead, they are represented by entries in the Navigator window, as in the above screenshot.

Adding Readers and Writers

Additional readers or writers are added to a translation using the Quick Add menu:

![Quick Add menu](image)

...Or by selecting Readers > Add Reader (Writers > Add Writer) from the menu bar:
This action opens a dialog, similar to the Generate Workspace dialog, in which you can define the parameters for the new reader or writer:

You can add as many readers and writers as you require in this way.

**TIP**

A reader can also be added by dragging a dataset from a file system explorer and dropping it onto the Workbench canvas.

**Removing a Reader or Writer**

If a reader or writer is no longer required, then it can be removed using options on the menu bar:
Alternatively, it’s possible to right-click a reader/writer in the Navigator window and choose the Delete option.

**Updating a Reader or Writer**

Readers and writers can be updated so that older workspaces have the speed and functionality available in a newer version of FME. You can update a transformer by right-clicking the reader/writer in the Navigator window and choosing the Update option:

For readers, this tool provides the option to either update the reader or to also update the list of read feature types. This way the workspace can be updated if the source data changes. Another way to update feature types is Reader > Update Feature Types on the menu bar.

**Multiple Feature Types**

Adding new readers and writers lets you read or write a new file. However, sometimes you want to read or write data to a different layer or table within an existing dataset. You can accomplish this by adding feature types. Remember, feature types belong to a reader or a writer.

You can add a feature type to an existing reader or writer by picking Readers > Import Feature Types or Writers > Add Feature Type:


Adding a reader feature type lets you read a new layer or table from the dataset of an existing reader; adding a writer feature type lets you write a new layer or table in the dataset of an existing writer.
Exercise 4.1  

<table>
<thead>
<tr>
<th>Data</th>
<th>3-1-1 case location details (XLS hosted on FTP)</th>
</tr>
</thead>
</table>
| Overall Goal | Add additional readers and writers  
Create a report using the HTMLReportGenerator |
| Demonstrates | Creating more complex FME workflows |
| Start Workspace | C:\FME\Data2018\Workspaces\IntroToDesktop\Ex4.1-Begin.fmw |
| End Workspace | C:\FME\Data2018\Workspaces\IntroToDesktop\Ex4.1-Complete.fmw |

City planners are undertaking strategic planning and want to know which local planning areas will need additional resources. They have requested reports of the 3-1-1 requests organized by local planning areas with tables and charts.

You have decided to create an HTML report with tables and charts to share with them. This task requires adding an additional writer to our workspace. In this exercise, we will add an HTML writer feature type and a transformer to generate the HTML report.

1) Start Workbench

Start Workbench (if necessary) and open the workspace from Exercise 3.3. Alternatively, you can open C:\FME\Data2018\Workspaces\IntroToDesktop\Ex4.1-Begin.fmw.

2) Add an Excel Writer

Because your manager wants both the original data with a new schema and the summary table, we need a new feature type for the results of the StatisticsCalculator. Let's write it back to Excel; many FME workflows write back to the same format they read!

Click Writers > Add Writer:

- **Add Writer...**
- **Add Feature Type...**
  - **Import Feature Types...**
  - **Update Feature Types...**
  - **Enable/Disable Feature Types...**
  - **Remove Feature Types...**
  - **Remove Writers...**
- **Move Feature Types...**
- **Redirect to FME Data Inspector**

Then fill the Add Writer dialog out:

<table>
<thead>
<tr>
<th>Format</th>
<th>Microsoft Excel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>C:\FME\Data2018\Output\Training\311-requests-summary.xlsx</td>
</tr>
<tr>
<td>Sheet Definition</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

Your dialog should look like this:
Because we want our feature type to copy the attributes coming out of the StatisticsCalculator Summary port, we changed the Sheet Definition method from Manual to Automatic. Feature types using this mode will automatically adjust their schema to match connected features.

Click OK. The Feature Type dialog will open and you will be prompted to enter the feature type name. Type in 311-requests-summary and click OK.

Move this new feature type above the original feature type and connect it to the Summary output port of the StatisticsCalculator:
Now we have a new Excel writer feature type that will write out the results of the StatisticsCalculator. We'll use this in the next exercise.

3) **Add an HTML Writer**

Here is another way to add a reader or writer: click on a blank space on the canvas and type HTML. You can use the mouse or ↑ and ↓ to browse the Quick Add menu. You should see HTML listed as an option under Writers:
Double-click it or press Enter to add an HTML writer. Use the following parameters:

<table>
<thead>
<tr>
<th>Writer Format</th>
<th>HTML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Writer Dataset</td>
<td>C:\FMEData2018\Output\Training\report.html</td>
</tr>
</tbody>
</table>

Your dialog should look like this:

Click OK to add the writer feature type to your canvas. Click and drag to move it above your 311-requests-summary writer feature type.
4) Add an HTMLReportGenerator

To create an HTML report, we will use an HTMLReportGenerator. We want to add it after the StatisticsCalculator, so it uses our summarized data, but we don't want to write out to Excel. So, we'll branch our data stream. Add an HTMLReportGenerator and connect it to the Summary port of the StatisticsCalculator. Then, connect the Output port of the HTMLReportGenerator to the HTML writer feature type:
Double-click the HTMLReportGenerator to open its parameters. First, set the Group By to Local Area. Doing so will tell the transformer to create a report for each local area separately.

Next, click where it says Chart (Bar) under Page Contents. This table is used to add elements to the report. From the drop-down, select Header:

Click somewhere under Content Settings to update the parameters for a Header block. For Text, choose the Local Area attribute. For Header Level, pick H1. These settings give us a header with the local area name in each section of the report. Your dialog should look like this:
Click the cell under Header in Page Contents and select Table from the drop-down:

Click somewhere under Content Settings to update the parameters for a Table block. Under Column settings, we'll create two columns, one for Department and one for Cases. Notice that if you type "Department" or "Cases" into the Column Name, FME will automatically choose the attribute value instead of a constant. To fix this, right-click the cell and choose Open Text Editor, then type in the name of the column. This method will store a constant value instead of supplying an attribute. Your dialog should look like this:

These settings will make an HTML table listing the number of cases by department.
Finally, let's add a Chart (Bar) to the Page Contents. Fill out the parameters like this:

Click OK. You can Run To This on the HTMLReportGenerator and inspect the cache, but all you will be able to see is the raw HTML in Data Inspector. We need to write this data out to an HTML file to view it in a browser.

5) Save your Workspace

Don't forget to save your workspace regularly.

6) View Your HTML Report

Click on the HTML feature type and select Run To This. Once it has run, click on the Open Containing Folder button to open C:\FMEData2018\Output\Training. You should see reports.html. Open it with your preferred web browser to inspect the heading, tables, and charts:
Arbutus-Ridge

<table>
<thead>
<tr>
<th>Department</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>311 Contact Centre</td>
<td>14</td>
</tr>
<tr>
<td>Business Planning &amp; Services</td>
<td>128</td>
</tr>
<tr>
<td>CSG - Inspections</td>
<td>41</td>
</tr>
<tr>
<td>CSG - Licenses</td>
<td>55</td>
</tr>
<tr>
<td>CSG - Licenses &amp; Inspections</td>
<td>4</td>
</tr>
<tr>
<td>DBL - Inspections</td>
<td>21</td>
</tr>
<tr>
<td>DBL - Licensing &amp; Animal Services</td>
<td>38</td>
</tr>
<tr>
<td>ENG - Office of the Deputy City Engineer</td>
<td>112</td>
</tr>
<tr>
<td>ENG - Solid Waste</td>
<td>610</td>
</tr>
<tr>
<td>ENG - Streets</td>
<td>830</td>
</tr>
<tr>
<td>ENG - Transportation</td>
<td>97</td>
</tr>
<tr>
<td>ENG - Water &amp; Sewer</td>
<td>275</td>
</tr>
<tr>
<td>Interdepartmental Initiatives</td>
<td>1</td>
</tr>
<tr>
<td>Public Safety - Fire</td>
<td>10</td>
</tr>
</tbody>
</table>

Perfect! The planning department can use this report to allocate resources for the next year.

CONGRATULATIONS

By completing this exercise, you have learned how to:
- Work with multiple writers
- Generate a report using HTMLReportGenerator
Best Practice

If a workspace runs to completion and produces the output you want, it can’t be bad, can it? Well, yes it can. It’s not enough just to put together a functioning workspace; it’s also vital to use FME in a manner that is both efficient and scalable.

What is Best Practice?

In general terms Best Practice means the best way of doing something; in other words, carrying out a task effectively and efficiently.

Despite the word ‘best,’ we’re not presuming the ideas here will meet every need and occasion. The best description of this concept I’ve heard – and one that fits well here – is:

“a very good practice to consider in this situation based on past experience and analysis.”

In this section we’ll talk about a small aspect of best practice - using annotations and bookmarks to make your workspace understandable to others. Both of these methods allow you to document how and why your workspace works as it does. They allow other users - or yourself in the future - to understand what each section does.

Annotation

Annotation is a crucial method for a clear and comprehensible design.

Annotation helps other users understand what is supposed to be happening in the translation and also supports the creator when returning to a workspace after a long interval.

User annotation is a comment created by the user. It can be connected to a workspace object (transformer or feature type), can be connected to a workspace connection, or can float freely within the workspace.

Adding Annotation

To create user annotation, right-click the canvas and select Insert Annotation, or use the shortcut Ctrl+K. You can attach annotation to canvas objects (feature types, transformers, etc.) by doing the above with an object selected.

Bookmarks

A bookmark, like its real-world namesake, is a means of putting a marker down for easy access.

With FME the bookmark covers an area of the workspace that is usually carrying out a specific task, so a user can pick it out of a broader set of transformers and move to it with relative ease.

Adding a Bookmark

To add a bookmark, right-click the canvas and select Insert Bookmark, use the shortcut Ctrl+B, or click the Bookmark icon on the toolbar:

Whereas a traditional bookmark marks just a single page in a book, the FME bookmark can cover a wide area of the canvas. Multiple bookmarks can divide a single workspace into different sections.
**TIP**

If you have any objects on the workspace canvas selected when you create a bookmark, the bookmark is automatically expanded to include those items.

---

**Resizing a Bookmark**

To resize a bookmark hover over a corner or edge and then drag the cursor to change the bookmark size or shape.

Click on the bookmark cogwheel to edit its properties:
**Exercise 4.2a**

---

### Best Practice: Bookmarks and Annotations

<table>
<thead>
<tr>
<th>Data</th>
<th>3-1-1 case location details (XLS hosted on FTP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Goal</td>
<td>Use bookmarks and annotations to organize your workspace</td>
</tr>
<tr>
<td>Demonstrates</td>
<td>Following best practice in developing FME workspaces</td>
</tr>
<tr>
<td>Start Workspace</td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex4.2-Begin.fmw</td>
</tr>
<tr>
<td>End Workspace</td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex4.2-Complete.fmw</td>
</tr>
</tbody>
</table>

Our final workspace is a bit complicated! We have a reader feature type, three writer feature types, and nine transformers. It should look something like this (click to expand):

![Diagram of workspace]

If one of your colleagues were to open this workspace, it would take them some time to figure out what exactly it was doing. Let’s make their lives easier by following best practice, using bookmarks and annotations to comment on the workspace. This step is also friendly to future you: if you return to this workspace in the future, you will be able to remember what it does by looking at the bookmarks and annotation.

Use the skills covered in the previous unit to create bookmarks for related sections of your workspace and annotations that explain what each transformer is doing. Make sure you add the following:

- Add bookmarks and annotation to your reader and writer feature types, if you don't already have them from Exercise 2.1
- Add bookmarks for transformation steps
- Add annotation to describe what transformers are doing
- Feel free to move all your canvas objects around to ensure your workspace is nicely laid out

---

112
Exercise 4.2b

**Best Practice: Bookmarks and Annotations**

<table>
<thead>
<tr>
<th>Data</th>
<th>3-1-1 case location details (XLS hosted on FTP)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overall Goal</strong></td>
<td>Use bookmarks and annotations to organize your workspace</td>
</tr>
<tr>
<td><strong>Demonstrates</strong></td>
<td>Following best practice in developing FME workspaces</td>
</tr>
<tr>
<td><strong>Start Workspace</strong></td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex4.2-Begin.fmw</td>
</tr>
<tr>
<td><strong>End Workspace</strong></td>
<td>C:\FMEData2018\Workspaces\IntroToDesktop\Ex4.2-Complete.fmw</td>
</tr>
</tbody>
</table>

Is your workspace now following best practice for bookmarks and annotation? It should look something like this (click to expand):

You can also check your workspace against the Complete exercise listed above.

**CONGRATULATIONS**

By completing this exercise, you have learned how to:
- Add, edit, and remove bookmarks
- Add, edit, and remove user annotation
FME Workflows Quiz

Bringing together multiple streams of features will automatically join them based on shared attributes.

True False Bringing together streams accumulates features. To join them based on a common key requires using a join transformer. Readers can have multiple feature types and feature types can belong to multiple readers.

True False Readers and writers can contain multiple feature types, but not the other way around.

Who might benefit from your use of annotation and bookmarks as part of best practice/style? Check all that apply.

Other FME users in your organization Customers to whom you deliver FME workspaces End users of the data produced by the workspace Yourself in the future, coming back to edit the workspace Anyone who might have to edit or understand your workspace will benefit if it uses bookmarks and annotation effectively.
Module Wrap-up

Congratulations on finishing the Introduction to FME Desktop module. You are now prepared to continue on your FME journey.

Learning Objectives Review

Here are the learning objectives you have met by completing the module:

After completing the Getting Started unit, you can:

- Describe FME and what it does.
- Distinguish between FME Desktop, FME Server, and FME Cloud.
- Open a workspace.
- Run a workspace.

After completing the FME Translations unit, you can:

- Define what a workspace is and generate one in FME Workbench.
- Inspect data using FME Data Inspector.
- Explain the difference between FME Workbench and FME Data Inspector.
- Explain what a schema is and edit it in FME Workbench.
- Explain what schema mapping is and do it using transformers.
- Turn feature caching on and off and use it when building a workspace.
- Use partial runs with feature caching.

After completing the FME Transformations unit, you can:

- Explain what transformers do in FME.
- Discuss common transformer categories.
- Locate and place transformers using Quick Add.
- Set transformer parameters.

After completing this FME Workflows unit, you can:

- Understand how data flows through a workspace.
- Create a workspace with multiple formats.
- Create a workspace with multiple feature types.
- Employ basic best practice techniques (bookmarks and annotations) in building your workspaces.

Product Information and Resources

Here are some resources to learn more about FME.

Training

We offer live online instructor-led training that builds on this introduction. You can also watch recordings of these training courses.

Safe Software Web Site

The Safe Software website is the official information source for all things FME. It includes information on FME products, Safe Software services, FME solutions, FME support and Safe Software itself.
Safe Support Team

Behind FME are passionate and fun experts, ready to help you succeed, with a support team philosophy built on the principle of knowledge transfer.

You can request product support through a Support Case (web/email) or using a Live Chat.

Your Local Partner

Safe Software has partners and resellers around the world to provide expertise and services in your region and your language.

You can find a list of official partners on the Safe Software Partners Page.
Safe Software Blog

The Safe Software blog provides technical information and general thoughts about FME, customers' use cases, and data interoperability. It includes articles, videos, and podcasts.

FME Documentation

Use the Help function in FME Workbench to access help and other documentation for FME Desktop. Alternatively, access the Documentation online.

Community Information and Resources

Safe Software actively promotes users of FME to become part of the FME Community.

The FME Community

The FME Community is a one-stop shop for all community resources, plus tools for browsing documentation and downloads.
Knowledge Base

The FME Knowledge Base contains a wealth of information; including tips, tricks, examples, and FAQs. There are sections on both FME Desktop and FME Server, with articles on topics from installation and licensing to the most advanced translation and transformation tasks.

Forums

FME community members post FME-related messages, ask questions, and share in answering other users’ questions. Members earn "reputation" and "badges," and there is a leaderboard of the top-participating users. Join the conversation to see how the community helps each other with their FME projects!

Ideas Exchange

FME development is very much user-driven. The Ideas Exchange gives users the chance to post their ideas for new FME functionality, or improvements to existing functionality, and allows everyone to vote on the proposed ideas. The more votes an idea gets, the more likely it is to be implemented!

The FME Channel

This FME YouTube channel has screen cast demos and other video content about FME. Besides this, there are a host of explanatory and helpful videos, including recordings of most training and tutorials.