Article

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

# Converting from XML (Simple XML Reading Example)

Aug 3, 2022 • Knowledge

**Product Type**
FME Desktop

**FME Version**
2022.0

**Tutorial:** Tutorial: Getting Started with XML (/s/article/tutorial-getting-started-with-xml) | **Previous:** XML Reader Configuration (/s/article/xml-reader-configuration) | **Next:** How to Read XSD-Driven XML (/s/article/How-to-read-XSD-Driven-XML)

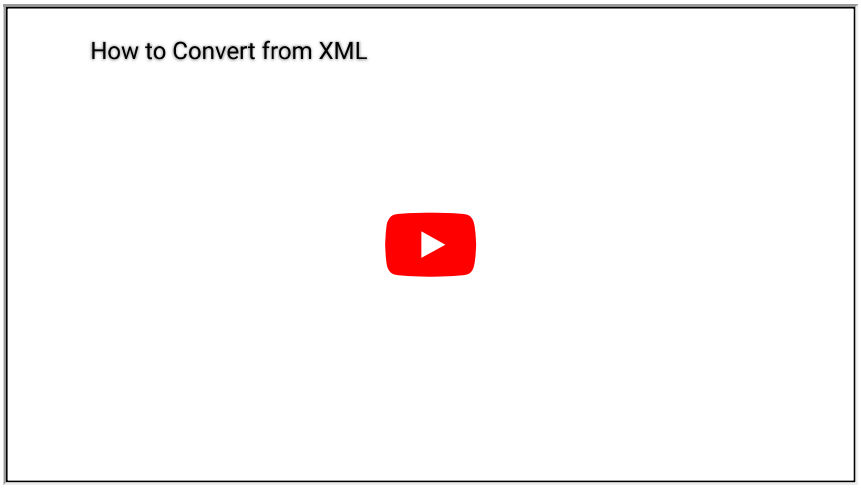
## Overview

Most often, XML is read with FME using predefined XML profiles for specific XML formats, or FME's XML reader with Feature Paths. Other approaches to reading include XfMaps (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/xml/xfMap.htm), or FME's Text File reader using XQuery (https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/XQuery/XQuery_functions.htm) and other XML processing transformers. These approaches are considered advanced and are covered in other articles. In this article, we will focus on feature paths.

## XML Profiles

The first thing to check is whether or not the data you are trying to read already is supported in FME as a specific XML format or profile. For frequently used XML profiles, these are often implemented as their own FME format reader and writer. FME supports more than 40 different XML formats. Using these formats is always easier than trying to configure the generic XML reader/writer. Examples of XML formats supported in FME include: Open Street Map (OSM), Google KML (https://community.safe.com/s/article/getting-started-with-kml), GPX (https://community.safe.com/s/article/reading-gpx-data), RSS (https://community.safe.com/s/article/reading-and-writing-georss-and-rss-feeds), LandXML (https://community.safe.com/s/article/creating-landxml-files), and GML (https://community.safe.com/s/article/reading-xmlgml), to name a few. To see a complete list, open up the reader or writer gallery and type XML in the search field.

## Video




*This video was recorded using an older version of FME. The user interface may be different, but the concepts are the same.*

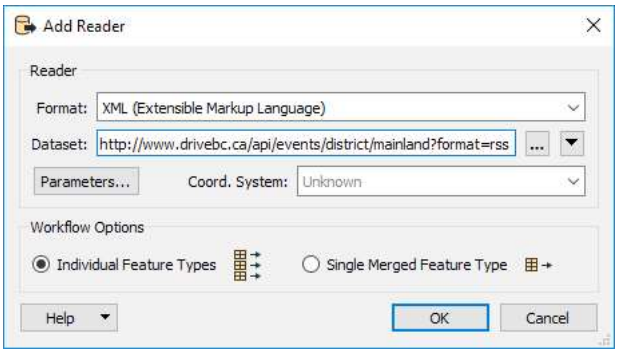
# Step-by-Step Instructions

In the following exercise, we will see how to read XML elements as records and flatten their member elements into attributes. We will also see how to extract the geometries.

### 1. Open FME Workbench

Open FME Workbench and create a blank workspace. Click on the Reader button to add a new reader. In the reader parameters, type in XML for Format and select the XML (Extensible Markup Language). Then for Dataset, paste in the following URL:

```
http://www.drivebc.ca/api/events/district/mainland?format=rss (http://www.drivebc.ca/api/events/district/mainland?format=rss)
```

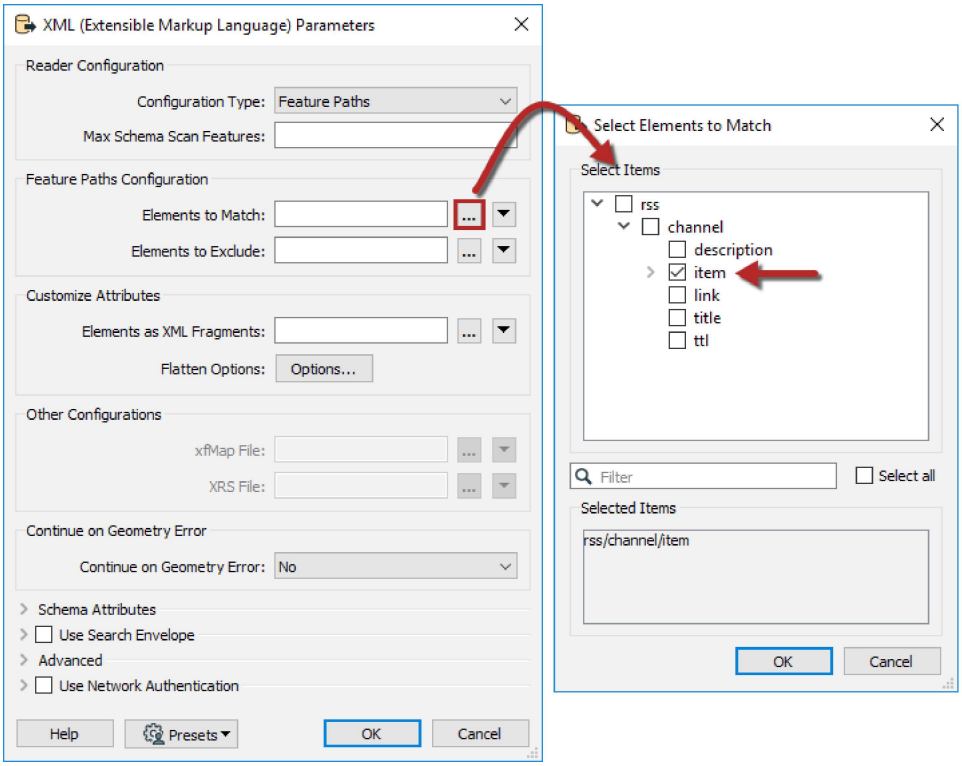



The DriveBC (https://www.drivebc.ca/) API records all of the traffic and weather events throughout British Columbia, Canada.

### 2. Open the Reader Parameters

Now click on the Parameters button, still inside the reader, these parameters will affect how FME reads in the XML file. Click on the ellipsis next to Elements to Match to open the dialog.

Expand RSS by clicking on the arrow, then expand channel, once channel is expanded, select item. Click OK three times.



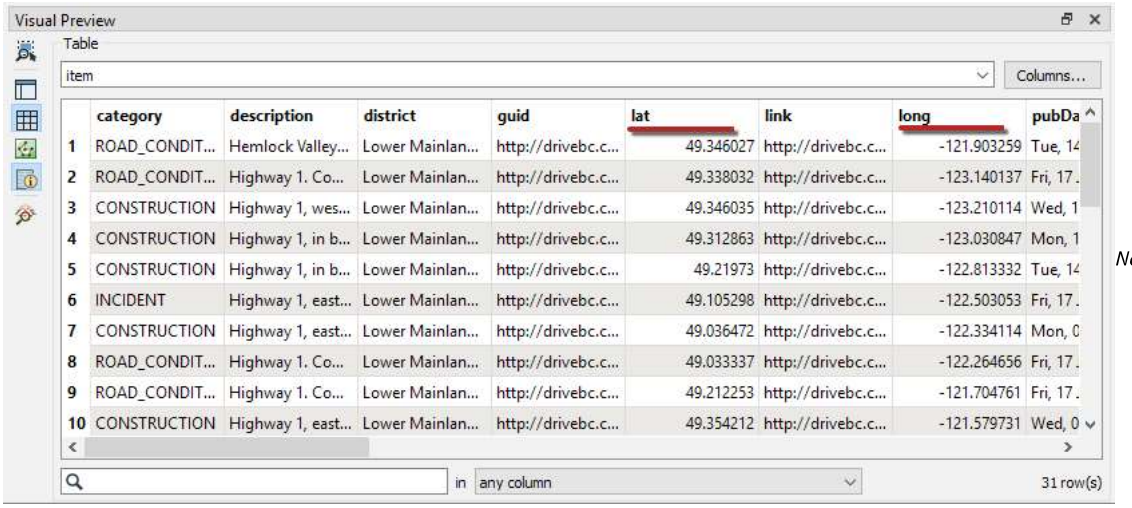

Note that we could also have just typed "item". The complete path is only important if the element type you are matching occurs at different levels in the dataset. We only selected the element we want to become a feature type. We didn't check RSS or channel since this would create very large features - e.g. one feature per dataset or feature type.

### 3. View the Dataset

Before we can manipulate our data any further, we should look at the structure of the data. Since this is a live data stream, the data will appear different than what is in the following screenshots.

Ensure Feature Caching is enabled, run the workspace, and then click on the Inspect Cached Features button. Optionally, you can connect an Inspector transformer and then run the workspace.

When viewing the data, you will see various road incidents in British Columbia with a latitude and longitude values. We can turn this data into points.

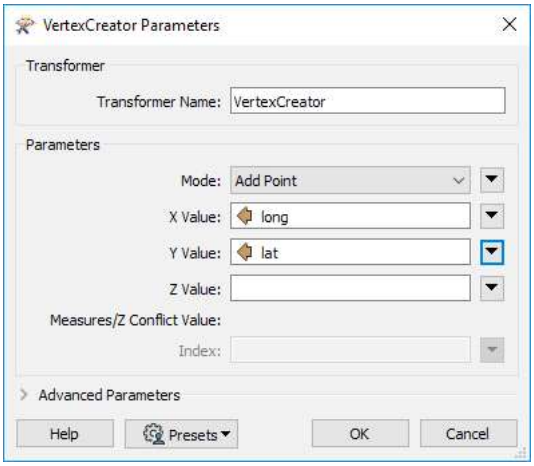



*Note: If you inspect the data and there are no current incidents being reported, download the provided dataset.*

### 4. Add a VertexCreator

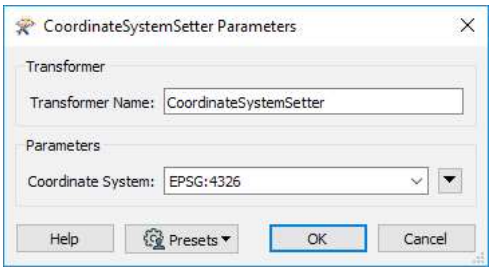
Add a VertexCreator transformer to the canvas and connect it to the item reader feature type.

In the parameters, set the X Value to long and then the Y Value to lat and then click OK.




### 5. Add a CoordinateSystemSetter

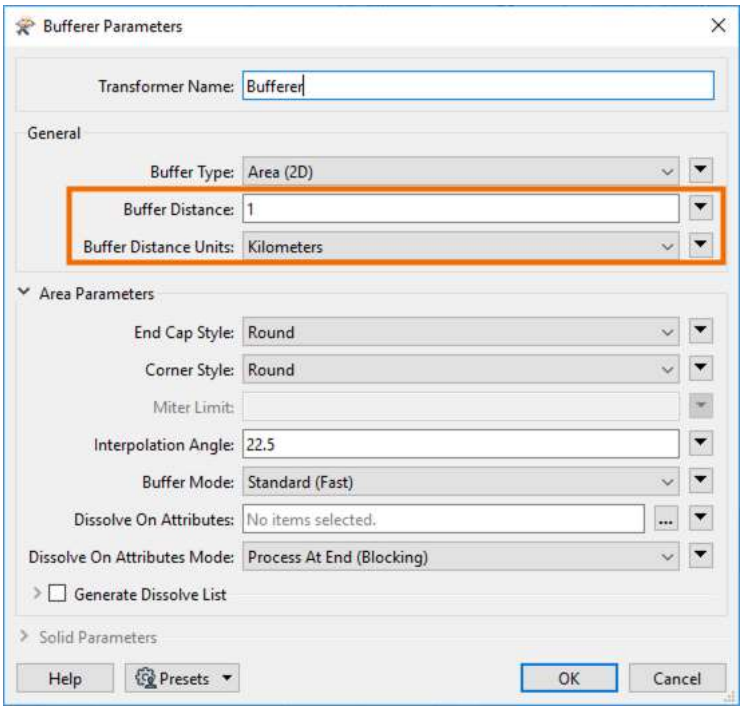
When adding XML data, a coordinate system needs to be set, so to do this, we will use the CoordinateSystemSetter transformer. Add a CoordinateSystemSetter to the canvas and in the parameters set the Coordinate System to EPSG:4326 and then click OK.




### 6. Buffer Points

Now that we have the coordinate system set, we can add a buffered area surrounding the points. This buffer will show us a larger affected area so that the user can avoid the congested traffic that might occur from the incident.

Add a Bufferer transformer to the canvas and connect it to the CoordinateSystemSetter. In the parameters, set the Buffer Unit Distance to Kilometers and then set the Buffer Unit to 1. Click OK.



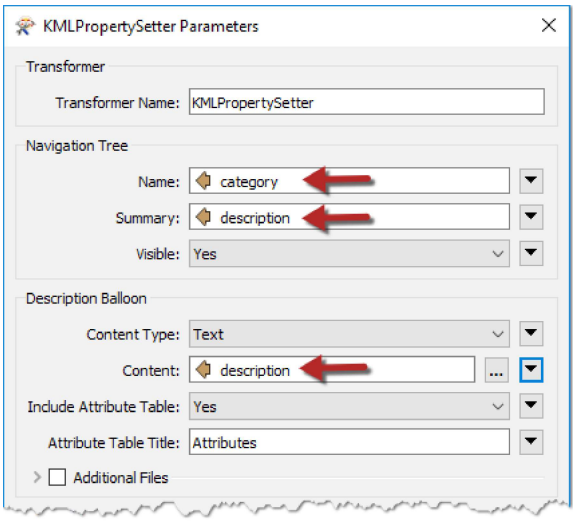

### 7. Add KMLPropertySetter

We want to set the properties for the KML file we will be writing out shortly. These properties will appear in the Google Earth navigation tree as well as when you

click on the point the KML balloon pops up.

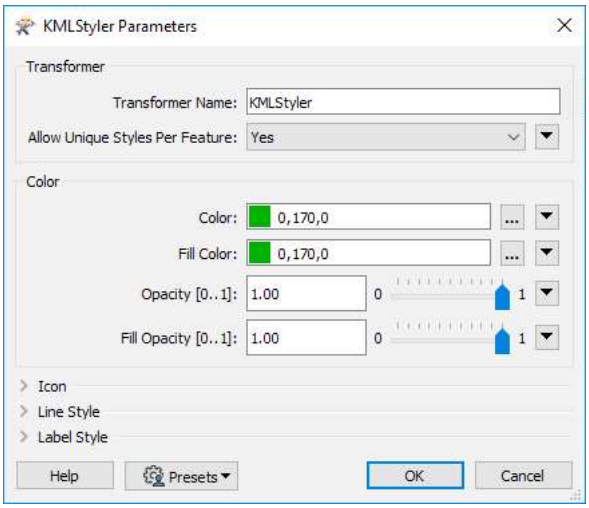
Add a KMLPropertySetter transformer to the canvas and connect it to the Output port on the GeographicBufferer. In the parameters, under Navigation Tree, set Name to the category attribute and then set Summary to the description attribute.

Then under Description Ballon, set the Content to the description attribute. Click OK to accept the parameters.



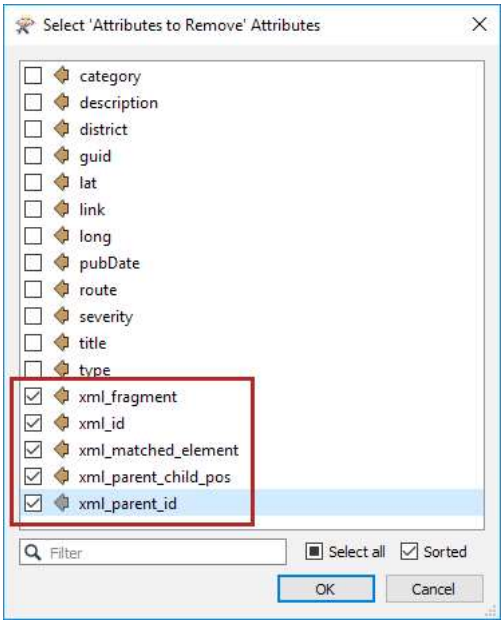

### 8. Add KMLStyler

The KML now needs to be styled. Add a KMLStyler transformer to the workspace and connect it to the KMLPropertySetter. In the parameters, change the Color and Fill Color to green, then set the Opacity and Fill Opacity to 1. This will make our buffered areas solid green.



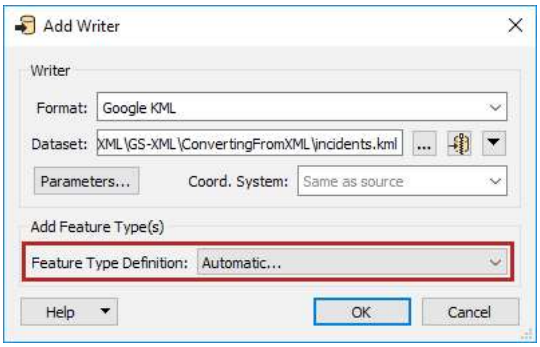

### 9. Clean Up Attributes

One final step before we can write out the data is to remove any excess attributes that we do not need. Add an AttributeRemover transformer to the canvas and open the parameters. In the parameters, click on the ellipsis next to Attributes to Remove and select any of the attributes that start with xml_.




### 10. Add a KML Writer

Click on the Writer button to add a new writer to the canvas. For Format, select Google KML, and then for Dataset, browse to a location to save the file and name it incidents.kml. Set the Feature Type Definition to Automatic and then click OK.

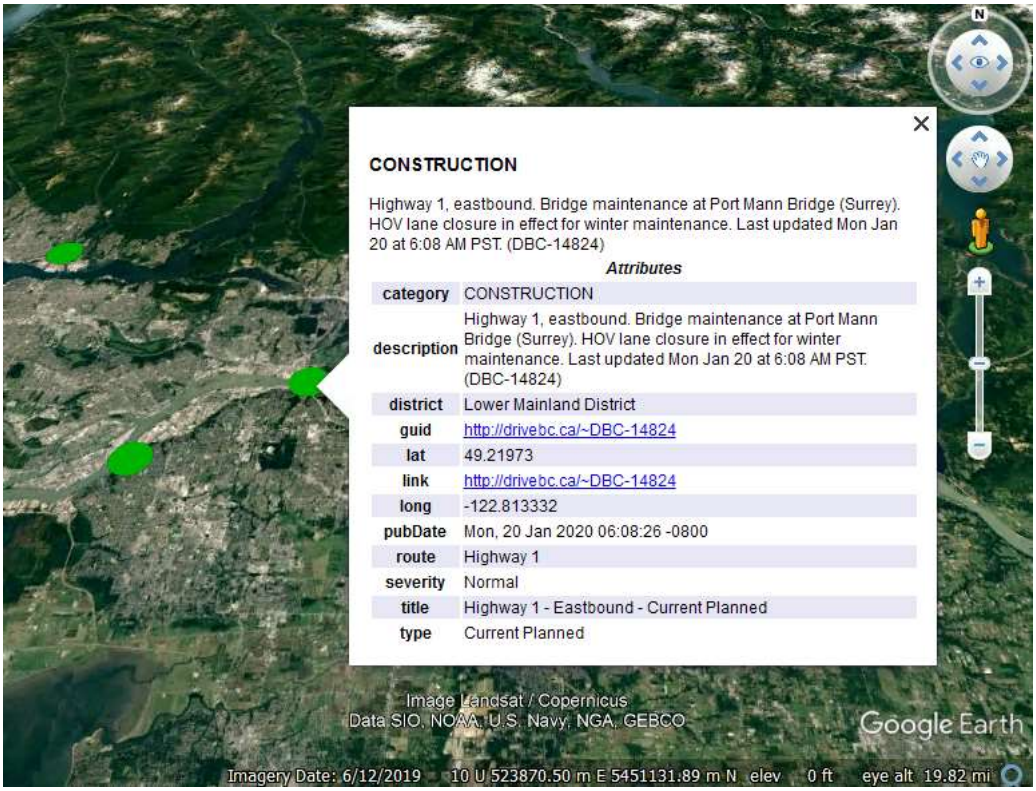



In the Feature Type dialog, set the Feature Type Name to Incidents and click OK. Connect the Incidents writer feature type to the AttributeRemover.

### 11. Run the workspace.

Run the workspace. Browse to the location where you saved the incidents.kml file and double-click on it to open it in Google Earth.

In Google Earth, notice the name and description in the navigation tree, also click on one of the points to open the information balloon and see the description we added to the balloon.




Congratulations! You have now read real-time XML data from the web, parsed it into features, and generated georeferenced geometries. You could now write this out to any format you choose simply by adding the appropriate writer (we've used KML in this example) and duplicating the source feature type to the destination.

## Continue to the next article: How to Read XSD-Driven XML (/s/article/How-to-read-XSD-Driven-XML)

## Data Attribution

The data used here originates from data made available by the British Columbia Government (https://www2.gov.bc.ca/gov/content/data/open-data/open-government-licence-bc). It contains information licensed under the Open Government License - British Columbia

**First Published Date**
7/29/2020, 12:15 AM

**Last Published Date**
8/3/2022, 9:43 PM

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

Sort by:
Latest Posts

LizAtSafe (/s/profile/0050c00000CeGV0AAN) (Employee) published a new version of this Knowledge.
August 3, 2022 at 9:43 PM (/s/feed/0D54Q00009ggOTrSAM)

Like Comment

Log In to Comment

trentatsafe (/s/profile/005a000000CdnWnAAJ) (Employee) published a new version of this Knowledge.
July 29, 2022 at 1:39 PM (/s/feed/0D54Q00009fwWPNSA2)

1 view

Like Comment

Log In to Comment

fmelizard (/s/profile/0054Q00000Ewu8dQAB) (Employee) published this new Knowledge.
July 29, 2020 at 12:15 AM (/s/feed/0D54Q000080hlHZSAY)

1 comment 33 views

Like Comment

cbmgit (/s/profile/0054Q00000Ex2TKQAZ)
2 years ago

Nice example , When I stuck in XML, for validation or for formatting I use this tool https://jsonformatter.org/xml-formatter (https://jsonformatter.org/xml-formatter) . Thank for in-depth Step-by-Step Instructions for transforming

Like

Log In to Comment

jason_balmut (/s/profile/0054Q00000Ewx3gQAB)
November 4, 2016 at 1:30 PM (/s/feed/0D54Q000080hn3oSAA)

I think this is a great basic start to reading XML feeds for point features and the attached workbench is easy to adapt to your own source data. I would really like to see an addition to the workbench for the linear features which are much more complex in how they have to be extracted.

3 comments 25 views

Like Comment

More comments 1 of 3

kimo (/s/profile/0054Q00000EwvAfQAJ)
2 years ago

I don't want to write an XML document - who would?? I prefer relational databases and SQL, not nested object-oriented structures. I had thought that XSD files would defined a schema or at least a structure that would automatically build a set of tables without manually picking out each element and each list. But sadly it seems that XSD files are not used for that, it is better to ignore them when unpacking XML. My task is to unpack the NZ companies XML file. It has 1.4M Entities - how hard is that, but wait, each entity has nested elements at least 10 levels deep. Each person is in a role which has a

Expand Post

Like

Log In to Comment

Follow

Files (1) (/s/relatedlist/ka14Q000001DWx5QAG/AttachedContentDocuments)

1a. Converting from XML
Aug 3, 2022 • 58KB • zip

View All

(/s/relatedlist/ka14Q000001DWx5QAG/AttachedContentDocuments)

## Related Articles

Reading XML - Simple Approach using Feature Paths (/s/article/reading-xml-simple-approach-using-feature-paths)

Simple Examples Using the RasterExpressionEvaluator Transformer (/s/article/simple-examples-using-the-rasterexpressionevaluato)

XML Reading with XfMaps: Flattening Example - Extracting all Attributes from an XML File (/s/article/xml-reading-with-xfmaps-flattening-example-extract)

How to Convert XML to Microsoft Excel (/s/article/how-to-convert-xml-to-microsoft-excel)

XfMap Example -- Reading JTS XML Test Files (/s/article/xfmap-example-reading-jts-xml-test-files)





**Land Acknowledgement —**

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