

Reading and Writing XML with FME 2011

Feb 2011

www.safe.com



Dean Hintz
Senior Analyst Professional Services



FME 2011: Reveal Your Hidden XML Powers...

www.safe.com



XML – A Format Monster?

www.safe.com



*Scary XML
threatening
to derail
your
project?*



XML – A Format Monster?

www.safe.com



*Scary XML
threatening
to derail
your
project?*



Fear Not!

**With FME you
can subdue
this beast...**



- Opportunity & Challenges
- XML Concepts
- Approaches in FME – Reading, Writing and Processing
- Reading XML in FME
 - Feature Maps (Reader and Transformers)
 - XQuery
 - XFMaps
- Processing XML: Transformers
- Writing XML
 - XMLTemplater
 - Formatting and Validation
- What's Coming (XMLTemplateMaker)
- Summary



- Opportunity: XML / GML are growing in use for exchange formats and metadata, because they are open and self-documenting.
- Challenge: This doesn't make it easy to read or write.
- XML documents are often very nested or object oriented.
- GIS is geared to working with flat or relational structures.



```
<?xml version="1.0" encoding="UTF-8"?>
<FeatureCollection>

  <Feature>
    <attribute1>John</attribute1>
    <attribute2>Vancouver</attribute2>
    <activeDate>
      <from>11-22-99</from>
      <to>12-11-09</to>
    </activeDate>
    <Coordinate_BOX id="101">
      <coords>-123.1,49.25 -122.9,49.15</coords>
    </Coordinate_BOX>
  </Feature>

  <Feature>
    <attribute1>June</attribute1>
    <attribute2>Surrey</attribute2>
    <activeDate>
      <from>02-25-05</from>
      <to>9-15-10</to>
    </activeDate>

    <Coordinate_BOX id="102">
      <coords>-122.8,49.12 -122.5,49.0</coords>
    </Coordinate_BOX>
  </Feature>

</FeatureCollection>
```

Relational

attribute1	attribute2	Active Date_From	Active Date_To
John	Vancouver	11-22-99	12-11-09
June	Surrey	02-25-05	9-15-10



- XML Elements
 - Child elements
 - Element attribute
- GML Objects
 - Object property
 - Object attribute
- XML Schema - xsd's
- Namespaces
 - `<ps-f:geometry>`



- XML Profile (GML, KML, GeoJSON etc)
- Feature Path (reader parameter)
- XQuery
- XFMaps
- XRS

*No one right answer,
Often a mixed approach is needed*



- FME supports reading more than 30 XML formats.
- Simply type XML in the Formats Gallery to see most of them.
- Examples: GML, KML, GeoRSS, WFS, GPX, CityGML, OSM, NAS, TIGERGML, LandXML, etc
- Don't forget to check the reader parameters
- Sometimes our profile reader does not support some custom schema variations.
- In this case using our XML reader and processing in a workspace can allow you to read virtually any XML, even if it is complex GML.

XML Formats in FME 2011

www.safe.com



Reader and Writer Gallery



Description	Short Name	Extensions	Type	Read	Write	Coord. Sys.	Licensed
Aeronautical Information Exchange ...	AIXM	.xml	File	•	•		•
Australian Asset Design and As Cons...	ADAC	.xml	File	•	•	•	•
BC MoF Electronic Submission Frame...	ESF	.gml, .xml, .gz	File	•	•	•	•
CityGML	CITYGML	.gml, .xml, .gz	File/URL	•	•	•	•
Dutch TOP10 GML	TOP10	.gml, .xml, .gz	File	•	•		•
ESRI Geodatabase (XML)	GEODATABASE_XML	.xml, .z, .zip	File	•		•	•
GeoRSS/RSS Feed	GEORSS	.xml	File/URL	•	•	•	•
German AAA GML Exchange Format (...)	NAS	.gml, .xml, .gz	File/URL	•		•	•
GML (Geography Markup Language)	GML	.gml, .xml, .gz	File/URL	•	•	•	•
GML SF-0 (Geography Markup Langu...	GMLSF	.gml, .xml, .gz	File/URL	•	•	•	•
GML v2.1.2 (Geography Markup Lang...	GML2	.gml, .xml, .gz	File		•	•	•
GPS eXchange Format (GPX)	GPX	.xml, .gpx	File	•	•	•	•
INSPIRE.protectedSites	INSPIRE_PROTECTEDSITES	.xml, .gml, .gz	File/URL	•		•	•
Land Victoria Incremental Update For...	IUF	.xml	File	•		•	•
Landonline (Land Information New Ze...	LANDONLINE	.xml	File	•		•	•
LandXML	LANDXML	.xml	File	•	•	•	•
Microsoft MapPoint Web XML	MSPOLYXML	.xml	File	•	•		•
NEN 3610 (GML)	NEN3610	.gml, .xml, .gz	File/URL	•		•	•
OpenStreetMap (OSM) XML	OSM	.osm	File/URL	•	•	•	•
OS (GB) MasterMap	DNF	.gml, .xml, .gz	File/URL	•		•	•
OS VectorMap Local	OSVECTORMAPLOCAL	.gml, .xml, .gz	File	•		•	•
RADARSAT-2 XML	RADARSAT2	product.xml	File/Directory	•		•	•
Swedish I2K (Interface 2000)	G2K	.gml, .xml, .gz	File	•	•		•
Swiss INTERLIS (ili2fme)	CH.EHI.FME.MAIN	.xml, .gml, .ili, .xtf, .itf	File	•	•		•
Trimble JobXML	JOBXML	.xml, .jxl	File	•			•
U.S. Census Bureau TIGER/GML	TIGERGML	.gml, .xml, .gz	File/URL	•		•	•
U.S. Environmental Protection Agenc...	EPA_GDXML	.xml	File	•		•	•
XDK (XML format for Danish DSFL)	XDK	.xml, .xdk, .gz	File	•			•
XML (Extensible Markup Language)	XML	.xml, .gml, .gz	File/URL	•	•		•

Example Source XML

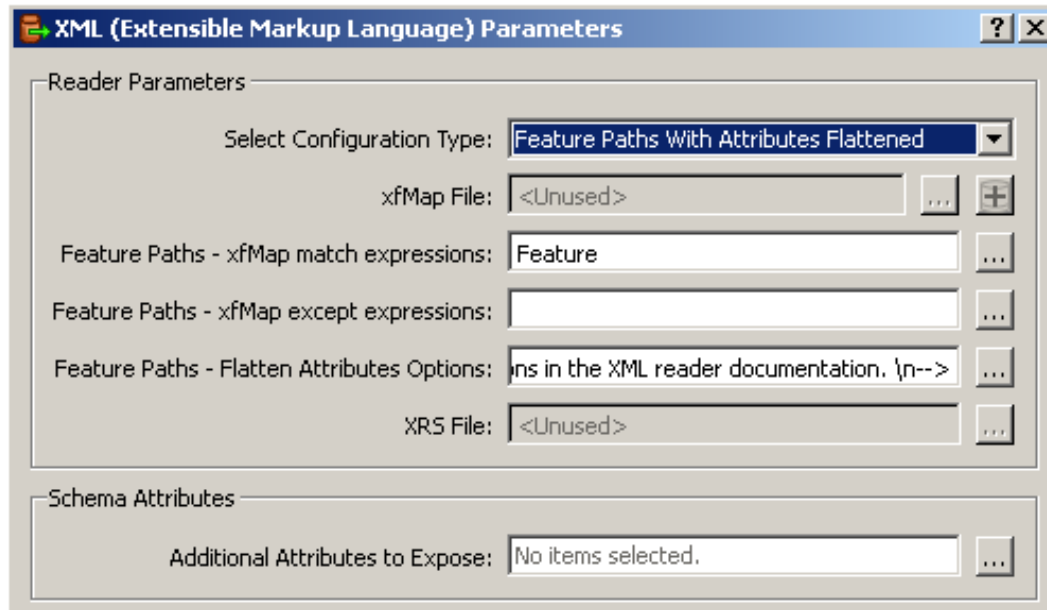
www.safe.com



```
<?xml version="1.0" encoding="UTF-8"?>
  <FeatureCollection>
    <Feature>
      <attribute1>John</attribute1>
      <attribute2>Vancouver</attribute2>
      <activeDate>
        <from>11-22-99</from>
        <to>12-11-09</to>
      </activeDate>
      <Coordinate_BOX id="101">
        <coords>-123.1,49.25 -122.9,49.15</coords>
      </Coordinate_BOX>
    </Feature>
    <Feature>
      <attribute1>June</attribute1>
      <attribute2>Surrey</attribute2>
      <activeDate>
        <from>02-25-05</from>
        <to>9-15-10</to>
      </activeDate>
      <Coordinate_BOX id="102">
        <coords>-122.8,49.12 -122.5,49.0</coords>
      </Coordinate_BOX>
    </Feature>
  </FeatureCollection>
```

Feature Path Option

www.safe.com



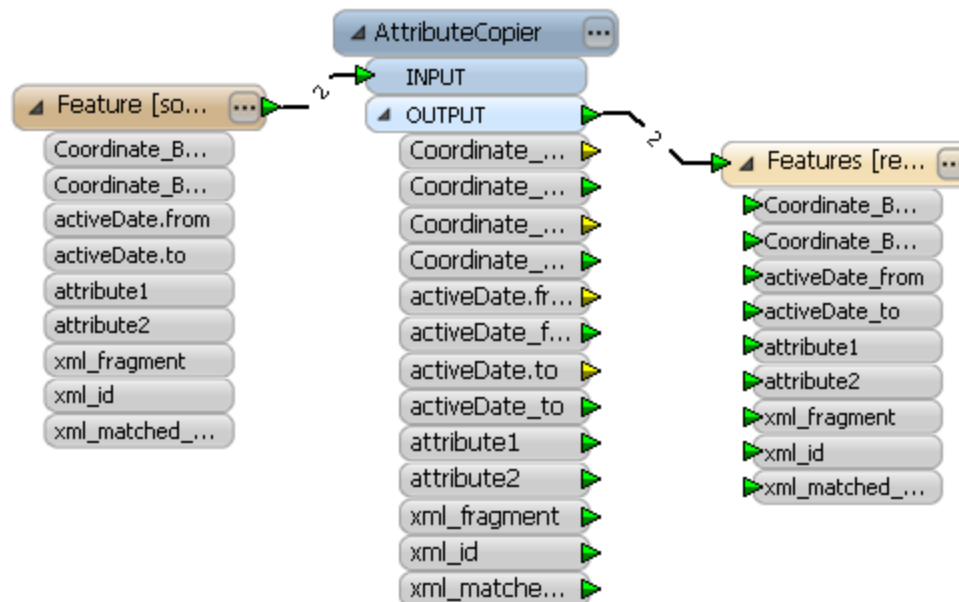
- Choose XML Reader and select XML source
- Select Configuration: Feature Paths with Attributes
- Set xfMap match expression: node to represent features (e.g. Feature in this case)
- Define any exceptions



Alternate method for defining exceptions for things you want to process elsewhere (e.g. geometry), or elements that you don't want to clutter up your schema. More restrictive than except expression in the path settings.

```
<structure
  separator="."
  map-empty-elements="yes"
  skip-matched="yes"
  cardinality="+"
  except="Coordinate_BOX"
  structure-prefix=""
  child-position-attribute=""
  attribute-identifier=""/>
```

Feature Path Workspace and CSV Result



Coordinate_BOX_coords	Coordinate_BOX_id	activeDate_from	activeDate_to	attribute1	attribute2	xml_id	xml_matched_element
-123.1,49.25 -122.9,49.15	101	11-22-99	12/11/2009	John	Vancouver	id-Feature-1.1	Feature
-122.8,49.12 -122.5,49.0	102	02-25-05	9-15-10	June	Surrey	id-Feature-1.2	Feature

Feature Path in Workbench

www.safe.com



- Can also use XMLFragmenter with Feature Path expressions
- Use an AttributeExposer to expose attributes that have been created
- Use FME geometry generation transformers for post processing such as 2DPointAdder

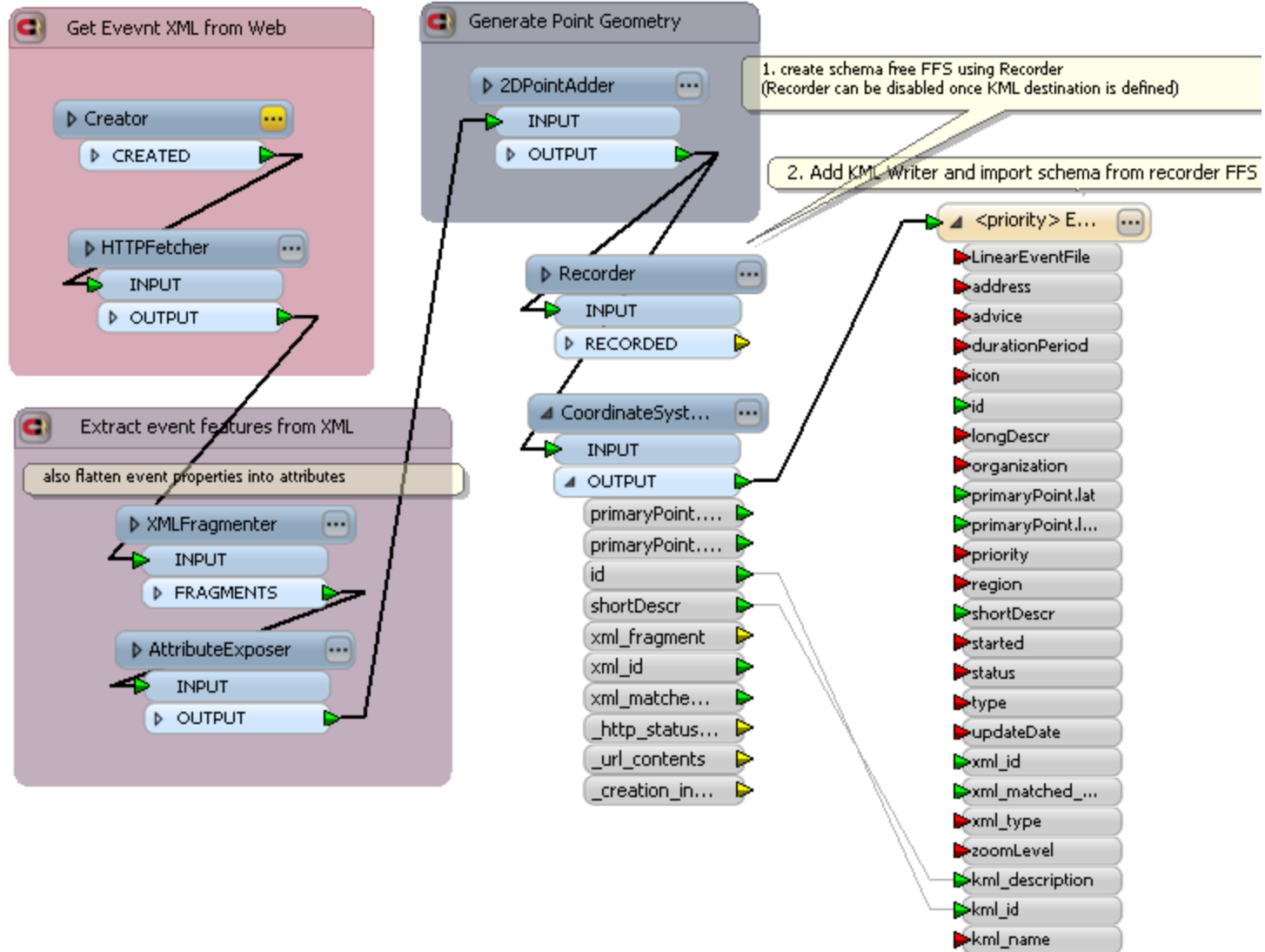
A screenshot of the XMLFragmenter Parameters dialog box. The dialog is titled "XMLFragmenter Parameters" and has a standard Windows window border with a question mark and close button in the top right. It is divided into several sections:

- Transformer:** A text field labeled "Transformer Name:" containing the text "XMLFragmenter".
- XML Source:** A section containing three fields: "XML Source Type:" with a dropdown menu set to "XML file"; "XML File:" with a dropdown menu set to "VL\\Examples\\XPath\\source.xml" and a browse button "..."; and "XML Attribute:" with a dropdown menu.
- Feature Paths Match:** A section containing three fields: "Feature Paths Match Type:" with a dropdown menu set to "Embedded expressions"; "Match Expressions:" with a text field containing "Feature" and a browse button "..."; and "Match Attribute:" with a dropdown menu.
- Feature Paths Except (Optional):** A section containing three fields: "Feature Paths Except Type:" with a dropdown menu set to "Embedded expressions"; "Except Expressions:" with a text field and a browse button "..."; and "Except Attribute:" with a dropdown menu.
- Attribute Flattening:** A section containing two fields: "Flatten Fragment into Attributes:" with a dropdown menu set to "Yes"; and "Flatten Options:" with a text field containing "XML reader documentation. \\n-->" and a browse button "...".

At the bottom of the dialog, there are four buttons: "Help", "Defaults" (with a dropdown arrow), "OK", and "Cancel".

Live Traffic Incidents Workspace: XML Web Feed to KML

www.safe.com



Source XML from HTTP Feed

www.safe.com



```
<Events>
- <event>
  <id>2942</id>
  <status>Planned</status>
  <priority>Minor</priority>
  <type>Construction</type>
- <shortDescr>
  E on Hammond Rd : reduced to one lane between Bonson Rd and Maple Meadows Way (
</shortDescr>
<longDescr/>
<primaryPoint lat="49.21721" long="-122.6783"/>
<secondaryPoint lat="" long=""/>
<updateDate>Jun 17 2008 9:58AM</updateDate>
<durationPeriod>64153</durationPeriod>
<started>6/16/2008 10:47:00 AM</started>
<ended/>
<address>Hammond Rd E Between Bonson Rd And Bonson Rd</address>
<organization>City of Pitt Meadows</organization>
<region>Pitt Meadows</region>
<zoomLevel>11</zoomLevel>
<advice/>
- <icon>
  /ATIS/App_Themes/ATIS/images/Grid/cur-construction.gif
</icon>
<Expr1/>
<LinearEventFile>D:\SrvApps\iMove\iMoveXML\LinearEvent2942.xml</LinearEventFile>
</event>
- <event>
  <id>2943</id>
  <status>Planned</status>
  <priority>Minor</priority>
  <type>Construction</type>
- <shortDescr>
  S on Harris Rd : closed intermittently between Streamstone Walk and Mitchell Rd from 10:
</shortDescr>
```

Result KML in Google Earth

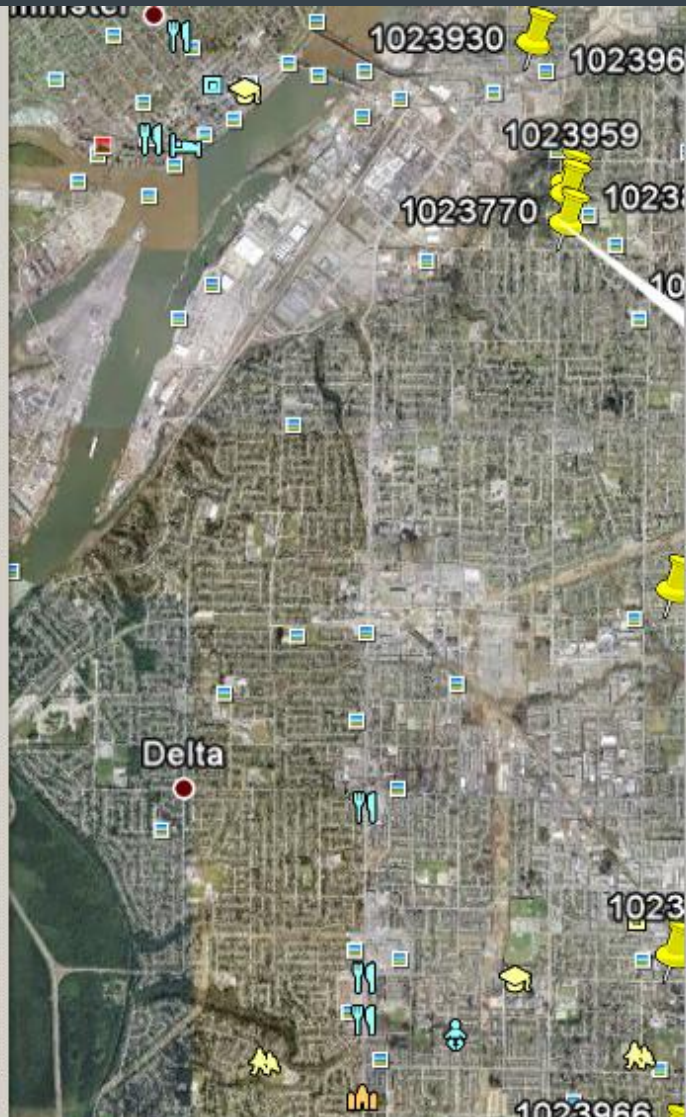
www.safe.com



Fly to e.g., 37.407229, -122.107162

Places

- events
- Major
 - [1023482](#)
Northbound on Highway 17 : 28th Ave closed due
 - [1023483](#)
Southbound on Highway 17 : 28th Ave closed due
 - [1023588](#)
Northwestbound on turning lane : Permanently
 - [1023770](#)
Northbound and Southbound on 128
 - [1023773](#)
232nd St : reduced to one lane between 80th
 - [1023837](#)
Northbound and Southbound on 128
 - [1023901](#)
5 on 41B St : reduced to one lane between
 - [1023902](#)
Southbound on 57B St : closed south of Deltaport
 - [1023914](#)
Northbound and Southbound on 150th
 - [1023930](#)



1023770

Northbound and Southbound on 128 Street : reduced to one lane between 104 Avenue and 107AAvenue due to Roadwork (Construction)

Attributes	
LinearEventFile	
address	128th St N Between 105th Ave And 105A Ave
advice	
durationPeriod	120120
icon	/ATIS/App_Themes/ATIS/images/Grid/cur-construction.gif
id	1023770
longDescr	
organization	City of Surrey
primaryPoint_lat	49.19407
primaryPoint_long	-122.8677
priority	Major
region	Surrey
shortDescr	Northbound and Southbound on 128 Street : reduced to one lane between 104 Avenue and 107AAvenue due to Roadwork (Construction)
started	11/29/2010 7:00:00 AM
status	Forecast
type	Construction
updateDate	Nov 29 2010 10:35AM
_xml_id	id-event-1.20
_xml_matched_element	event

XML Processing with XQuery Expressions: XQueryExtractor

www.safe.com



XML:

```
<Feature>  
  <attribute1>John</attribute1>  
  <attribute2>Vancouver</attribute2>  
</Feature>
```

XQueries:

```
//attribute1 = <attribute1>John</attribute1>  
string(//attribute1) = John  
fme:set-attribute("name", data(//attribute1))  
Result: Name = John
```



- Namespaces: If your source XML has declared namespaces, you won't get any matches unless you declare the namespace in your query.

XML:

```
<gml:Feature>  
  <attribute1>John</attribute1>  
  <attribute2>Vancouver</attribute2>  
</gml:Feature>
```

XQuery:

```
declare namespace x = "http://www.opengis.net/gml";  
string(//x:attribute1)
```



fme:has-attribute(<string>)

- Returns a boolean (true/false) value if the current feature has the specified attribute.

fme:get-attribute(<string> [, <default value?])

- Returns the value of an attribute. The first parameter contains the attribute name, and the second an optional default value.

fme:set-attribute(<string>, <value>)

- Sets the value of a feature attribute. The first parameter defines the attribute name and the second value contains the value.

XQueryUpdater - Provides updates to an XML document using XQuery Update expressions (retains XML structure).

XQueryExploder - Uses XQuery expressions to generate new features

XQueryExtractor - Uses XQuery expressions to extract portions of XML text into feature attributes

XQuery Expressions: Extracting Element Attributes

www.safe.com



XML:

```
<Coordinate_BOX id="101">  
  <coords>-123.1,49.25 -122.9,49.15</coords>  
</Coordinate_BOX>
```

XQuery:

```
string(//Coordinate_BOX/@id)
```

Used to extract attributes of XML / GML elements such as xml_id or gml_id, CRS, geometry traits etc.

FME XML Reading: enabled by XFMaps

www.safe.com



- Reading an XML file is not as simple as pointing FME at XML and XSD files. You need to define how you want to interpret the XML.
- Must define:
 - what **node level** becomes a **feature type**
 - what **child elements** under that node are to become **fields** or **geometries** of that feature type.
- In FME, xfMaps are one way to do feature mapping from XML to FME.
- Reading for most custom GML/XML formats in FME are done using xfMaps.



- **FeatureMap**
 - Define Node level to make into a feature
- **Feature Content Map**
 - Define content for your features
- Reference Map (optional)
 - Define placeholders to store variables
- Schema Type (optional)
 - Define Data structures independent of data content

Step 1: Define Feature Type

www.safe.com



```
<?xml version="1.0"?>
<xfMap>
<feature-map>
  <mapping match="Feature">
    <feature-type> <literal expr="Feature"/>
    </feature-type>
  </mapping>
</feature-map>
```

Step 2: Define Feature Content

www.safe.com



```
<feature-content-map>  
  <mapping match="attribute1">  
    <attributes>  
      <attribute>  
        <name> <literal expr="attr1"/> </name>  
        <value> <extract expr="."/> </value>  
      </attribute>  
    </attributes>  
  </mapping>  
</feature-content-map>
```



Using the xfMap:

- Add XML data source
 - Choose xfMap configuration type in the XML reader parameters
 - Browse to xfMap file
- /OR/
- Use XMLFeatureMapper and specify the source xml attribute or file path for the source XML and the path or expression for your xfMap

Result:

- Feature 1: attr1 = "John"
- Feature 2: attr1 = "June"

Multiple Attributes

www.safe.com



```
<mapping match="attribute1 attribute2">
  <attributes>
    <attribute>
      <name><matched expr="local-name"/> </name>
      <value> <extract expr="."/> </value>
    </attribute>
  </attributes>
</mapping>
```



```
<mapping match="activeDate">
  <attributes>
    <attribute>
      <name> <literal expr="date_start"/> </name>
      <value> <extract expr="./from" /> </value>
    </attribute>
    <attribute>
      <name> <literal expr="date_end"/> </name>
      <value> <extract expr="./to" /> </value>
    </attribute>
  </attributes>
</mapping>
```



```
<mapping match="Coordinate_BOX">  
  <geometry activate="xml-box">  
    <data name="data-string">  
      <extract expr="./coords"/>  
    </data>  
  </geometry>  
</mapping>
```

Completed XFMap

www.safe.com



```
<?xml version="1.0"?>
<xfMap>
  <feature-map>
    <mapping match="Feature">
      <feature-type> <literal expr="Feature"/> </feature-type>
    </mapping>
  </feature-map>
  <feature-content-map>
    <mapping match="attribute1 attribute2">
      <attributes> <attribute> <name> <matched expr="local-name"/>
      </name> <value> <extract expr="."/> </value> </attribute> </attributes>
    </mapping>
    <mapping match="activeDate">
      <attributes> <attribute> <name> <literal expr="date_start"/> </name>
      <value> <extract expr="./from" /> </value> </attribute> <attribute> <name>
      <literal expr="date_end"/> </name> <value> <extract expr="./to" /> </value>
      </attribute> </attributes>
    </mapping>
    <mapping match="Coordinate_BOX"> <trait> <name> <literal expr="id"/> </name>
      <value> <extract expr="@id"/> </value> </trait>
      <geometry activate="xml-box"> <data name="data-string">
      <extract expr="./coords"/> </data> </geometry>
    </mapping>
  </feature-content-map>
</xfMap>
```


Result In Viewer

www.safe.com



The screenshot shows the FME Universal Viewer interface. The main window displays a feature in a viewer. The feature is a polygon with a green cross-hatch pattern. The interface includes a menu bar (File, View, Window, Help), a toolbar with various icons, and a Viewport showing the feature. The Viewport also contains a red rectangle and a green cross-hatched rectangle. The right-hand side of the interface displays a metadata panel with the following information:

Feature: 1 of 1
Feature Type: Feature
Coord Sys: Unknown

Attribut...	Attribute Value
attribute1	June
attribute2	Surrey
date_end	9-15-10
date_start	02-25-05
fme_geo...	fme_polygon
fme_type	fme_area
xml_type	xml_area

Bounding Box:
Lower Left: (-122.8,49.
Upper Right: (-122.5,49.

Geometry Type: IFMEPolygon
Boundary:
Geometry Type: IFMELine
Number of Coordinates:
0: (-122.8,49)
1: (-122.8,49.119999999
2: (-122.5,49.119999999
3: (-122.5,49)
4: (-122.8,49)

Coord Sys X: -122.4922, Y: 49.0869 NUM



- Flatten
- Dynamic Schema
- Geometry traits
 - Reference Maps (inheriting parent ids etc)
- Aggregates / multi-geometries / optional
- Features within features
- Troubleshooting
- XRS – a script that orchestrates multiple XFMaps, often used in FME XML Readers



```
<mapping match="Feature/*" >  
  <structure separator="_"  
  cardinality = "*/+"/> >  
</mapping>
```

Note: same results as XMLFragmenter with Flatten, except that here you have the flexibility to flatten where it makes sense and configure custom processes for geometry etc where it doesn't.



```
<Feature>  
  <attribute1>John</attribute1>  
  <attribute2>Vancouver</attribute2>  
  <activeDate>  
    <from>11-22-99</from>  
    <to>12-11-09</to>  
  </activeDate>  
</Feature>
```

```
attribute1 = John  
attribute2 = Vancouver  
activeDate_from = 11-22-99  
activeDate_to = 12-11-09
```



```
<mapping match="property"  
  except = "geometry">  
  <attributes>  
    <attribute type="sequenced">  
      <name>  
        <extract expr="@fieldName"/>  
      </name>  
      <value> <extract expr="."/> </value>  
    </attribute>  
  </attributes>  
</mapping>
```



```
<FeatureMember>  
<property fieldName="status" type="string">Existing</property>  
<property fieldName="date_installed" type="date">1996-10-  
  26</property>  
<property fieldName="owner_type" type="string">Company  
  Owned</property>  
<property typeName="owner_name" type="string">ABC  
  Net</property>
```

XFMap RESULT:

Status = Existing

Date_installed = 1996-11-27

owner_type = Company Owned

owner_name = ABC Net



```
<trait>  
<name> <literal expr="id"/> </name>  
  <value> <extract expr="@id"/> </value>  
</trait>
```

XFMap Result:

Id = 101



Things to remember:

- Start simple
- Use templates / patterns
- FME only scans each element once. If matches overlap, first one wins. Use 'except'.
- Within a mapping, only child elements are directly accessible
- Mapping will be used wherever it matches
- XFMap on FMEpedia
- XFMaps in Reader's Writers – XML docs

XML Processing: XML Transformers

www.safe.com



- **XMLFeatureMapper** – make features with xfmap
- **XMLFormatter** – clean up XML formatting
- **XMLFragmenter** – extract XML fragments and generate features and attributes
- **XMLTemplater** – template + FME values = XML
- **XMLValidator** – check syntax and schema
- **XQueryExtractor** – extract XML elements
- **XQueryExploder** – create new features
- **XQueryUpdater** – update XML (replace values)
- **XSLTProcessor** – Process XML with xslt script
- **XMLNamespaceDeclarer** – Add required namespaces



- AttributeExposer
- Creator
- AttributeFileReader
- GeometryReplacer
- GeometryExtractor (GML 3.2.1 etc)
- HTTPFetcher
- Attribute / String processors (Concatenator)
- List processors (ListBuilder, ListConcatenator)
- SchemaMapper



XMLTemplater allows you to merge FME attributes into an XML template to build up XML text for writing by the Text File writer

- The XMLTemplater transformer reads one or more template files containing a mix of XML and XQuery statements.
- It then substitutes attribute values into the template by processing the embedded XQuery syntax (merge operation).
- A number of FME-specific XQuery functions are available
- Once processed, the XML can be:
 - written to a file with the Text File writer
 - written as an attribute to a database
- We used to write XML using XSLT but this is a lot harder to learn and also a lot slower



fme:get-attribute("")

- Substitute FME feature attribute into XML at the function location.

fme:get-list-attribute("")

- Concatenate all the specified list values together at the function's location. No for loop needed.

fme:get-xml-attribute("")

- Extract xml attribute without converting <, > to >, <.

fme:get-xml-list-attribute("")

- Extract XML list at function location – ideal for inserting feature type XML into dataset template.

XMLTemplater Example Process

www.safe.com



- Read a CSV file and generate an XML file with a specific schema – reverse process to XPath example above.
- Use one XMLTemplater to build the feature XML and another to assemble the features into the dataset XML.
- Each different level of hierarchy requires another XMLTemplate - particularly if there are one to many parent / child relationships.
- The XML templates are embedded in the XMLTemplaters, though they can also be maintained externally.



1. Read the CSV
2. Build feature XML per record. The feature template contains the desired XML structure with `fme:get-attribute()` commands where we want to insert FME attribute values.
3. Build feature list per feature type.
4. Insert feature list into dataset XML template. The dataset template contains the XML document header and footer and an `fme:get-xml-list-attribute()` command to insert all the pre merged XML features into the dataset.
5. Pretty print and validate XML before output - XMLFormatter, XMLValidator
6. Write with Text File writer as its schema free – can write out any XML we create (XML writer - flat tables only).



Source CSV:

id	_xmax	_xmin	_ymax	_ymin	_zmax	_zmin	name	city	end	start
101	-122.9	-123.1	49.25	49.15	0	0	John	Vancouver	12-11-09	11-22-99
102	-122.5	-122.8	49.12	49	0	0	June	Surrey	9-15-10	02-25-05

Desired result, one record (starting point for our templates):

```
<?xml version="1.0"?>
  <FeatureCollection>
    <Feature>
      <attribute1>John</attribute1>
      <attribute2>Vancouver</attribute2>
      <activeDate> <from>11-22-99</from> <to>12-11-09</to>
    </activeDate>
      <Coordinate_BOX id="101"> <coords>-123.1,49.25 -
        122.9,49.15</coords> </Coordinate_BOX>
    </Feature>
  </FeatureCollection>
```



```
<Feature>
  <attribute1> {fme:get-attribute("name")} </attribute1>
  <attribute2> {fme:get-attribute("city")} </attribute2>
  <activeDate>
    <from> {fme:get-attribute("start")} </from>
    <to> {fme:get-attribute("end")} </to>
  </activeDate>
  <Coordinate_BOX id="{fme:get-attribute("id")}">
    <coords> {fme:get-attribute("_xmin")},
    {fme:get-attribute("_ymax")}
    {fme:get-attribute("_xmax")},
    {fme:get-attribute("_ymin")} </coords>
  </Coordinate_BOX>
</Feature>
```

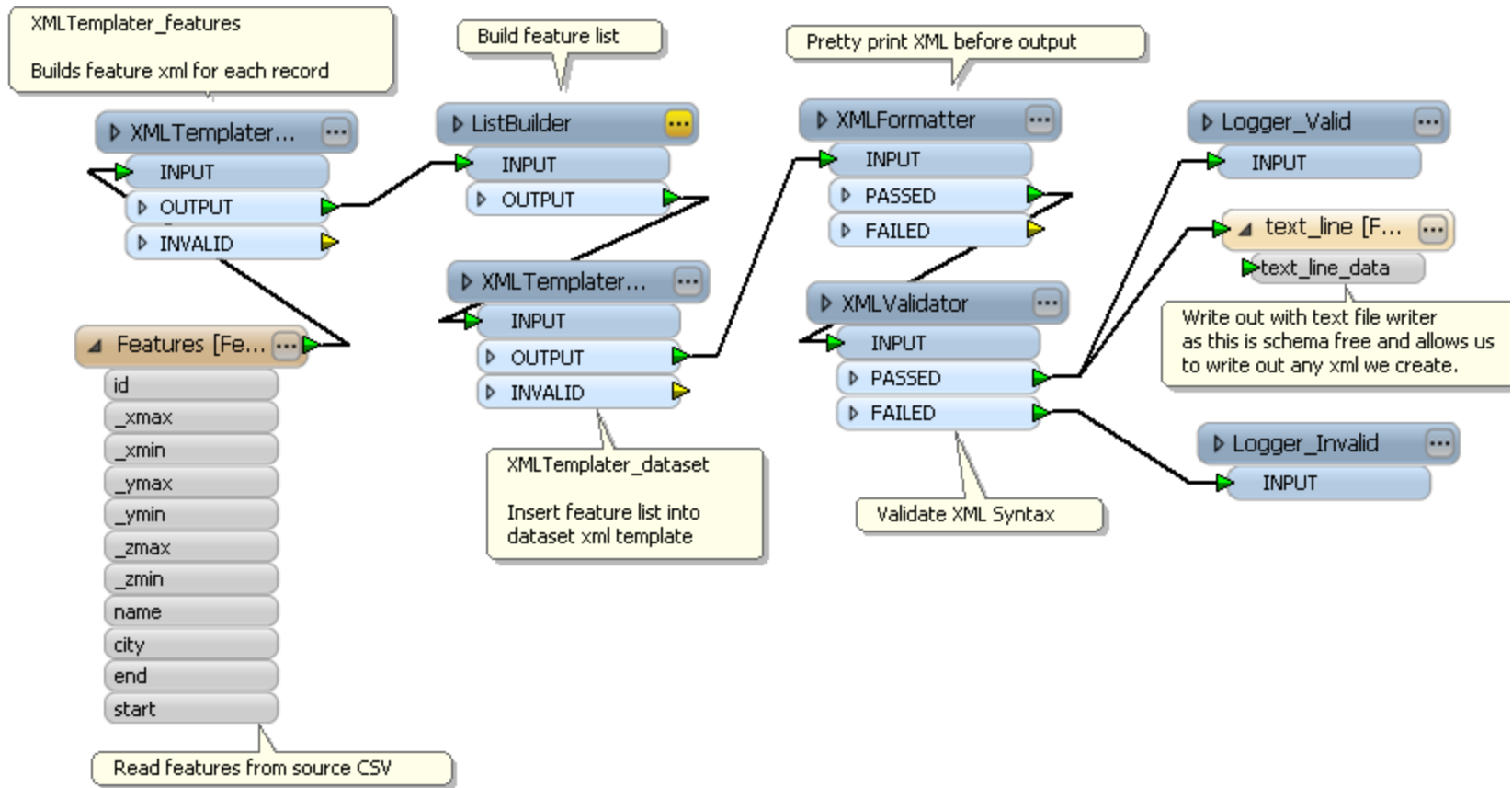



```
<?xml version="1.0"?>  
<FeatureCollection>  
  {fme:get-xml-list-attribute("_list{}.features")}  
</FeatureCollection>
```

- The dataset template concatenates the feature XML together and insert it into the dataset structure.
- First roll up your features into an XML list and then in the XMLTemplater read that list into the dataset template using `fme:get-xml-list-attribute()`

XMLTemplater Workspace

www.safe.com



XMLTemplater Result

www.safe.com



```
0 10 20 30 40
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <FeatureCollection>
4 <Feature>
5   <attribute1>John</attribute1>
6   <attribute2>Vancouver</attribute2>
7   <activeDate>
8     <from>11-22-99</from>
9     <to>12-11-09</to>
10  </activeDate>
11  <Coordinate_BOX id="101">
12    <coords>-123.1,49.25 -122.9,49.15</coords>
13  </Coordinate_BOX>
14 </Feature>
15 <Feature>
16   <attribute1>June</attribute1>
17   <attribute2>Surrey</attribute2>
18   <activeDate>
19     <from>02-25-05</from>
20     <to>9-15-10</to>
21   </activeDate>
22   <Coordinate_BOX id="102">
23     <coords>-122.8,49.12 -122.5,49</coords>
24   </Coordinate_BOX>
25 </Feature>
26 </FeatureCollection>
```





- Good to format the XML before output.
- Makes it easier to review in case some of our concatenation processes interfere with white space
- Remove redundant namespace declarations, unneeded xsi:schemaLocations
- Check the XML for validity with the XMLValidator. Two settings:
 - Syntax Only
 - Syntax and Application schema (need xsd)



- Each feature type normally has 1 `fme:get-xml-list-attribute` in the dataset template
- For highly nested structures, one feature type may be stored inside another. Requires a new XMLTemplate at every level such as:

Dataset template

|

--- XML template for Parent Feature Type

|

-----XML template for child feature type

- For geometries, can use a separate template to build up XML geometry and a `fme:get-xml-attribute` command to insert that into the feature template.
- The GeometryExtractor can extract your source feature geometry as XML, GML 3.1.1, GML 3.2.1, etc, for insertion into your feature template.



- XMLTemplateMaker – addresses last major gap in our XML workflow.
- Scan XSD and create sample XML templates based on user selection for node etc
- This can be used to serve as an XML feature or dataset template when you only have the schema xsd and no source XML data
- Very useful for INSPIRE where little data previously exists, and it can be difficult to 'read' the xsd and manually create XML that is compliant (try it sometime!)



- Option to suppress fragments name space declarations
- XMLFlattener – simplified version of XMLFragmenter which just flattens
- XFMapGenerator? – use FME to auto-generate XFMaps that you can later modify to add things like geometry processing, complex features etc
- Improved list processing. 2011 has a bug that reads lists onto a feature as `_list{0}`, `_list{1}` etc. This can be manually edited, but in 2012 this will just be read as `_list{}`.



- Opportunity, Challenges & Concepts
- Approaches in FME – Reading, Writing and Processing
- Reading XML in FME
 - Feature Maps (Reader, XMLFragmenter, flatten option)
 - XQuery (XQueryExtractor, XQueryUpdater)
 - XFMaps (XMLFeatureMapper)
- Processing XML: Transformers (XQueryUpdater)
- Writing XML
 - XMLTemplater
 - GeometryExtractor (GML 3.2.1)
 - Formatting and Validation (XMLFormatter, XMLValidator)
- Coming in 2012 (XMLTemplateMaker)

XML Monster – Not So Scary

www.safe.com



Well, it may not be exactly child's play, but with these FME tools you too can tame that XML Monster near you!



Thank You!

www.safe.com



Questions?

Contact the XML X-men:

support@safe.com

Also see FMEpedia:

www.fmepedia.com/index.php/XMLTemplater

www.fmepedia.com/index.php/Xfmap

www.fmepedia.com/index.php/XQueryUpdater

