SAFE SOFTWARE

www.safe.com

FME® Transformer Reference Guide

FME® Transformer Reference Guide

This guide contains a high-level summary of each transformer's functionality.
For detailed information, you can select *FME Transformers* from the Workbench Help menu, or visit *www.safe.com/support/onlinelearning/documentation.php*

CONTENTS

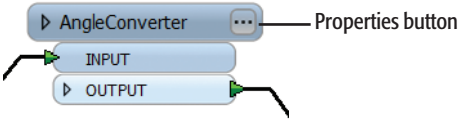
What is a Transformer?

A transformer is an FME Workbench object that carries out the restructuring of features. FME contains over 400 different transformers to carry out different types of restructuring. In the Workbench interface, transformers are stored in the Transformer Gallery and grouped in categories applicable to their associated functionality.

The Basics: Placing and Editing Transformers

There are many ways to place a transformer on the Workbench canvas. To start, however, you can simply double-click the transformer name and it will appear in the workspace.

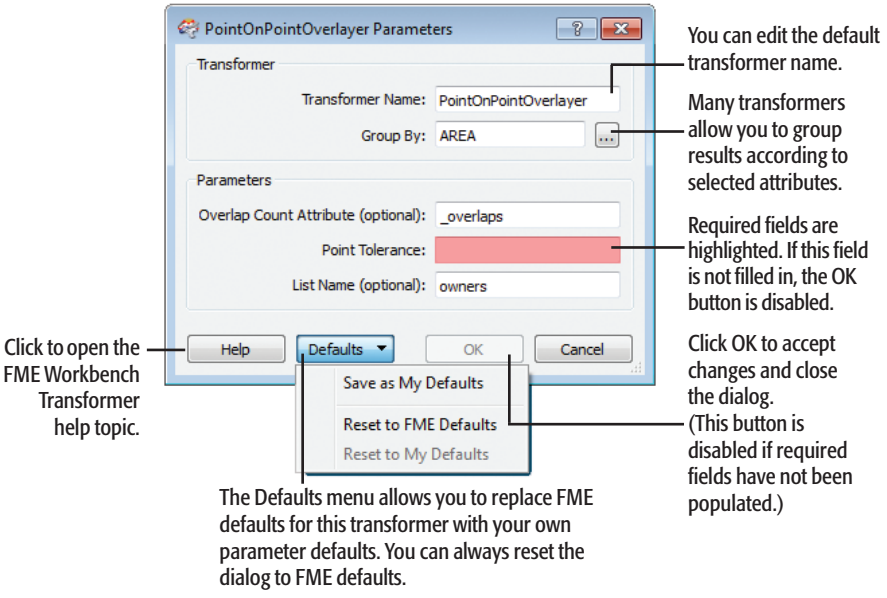
Every transformer has a Properties button on the right of the transformer.

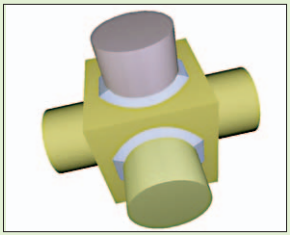


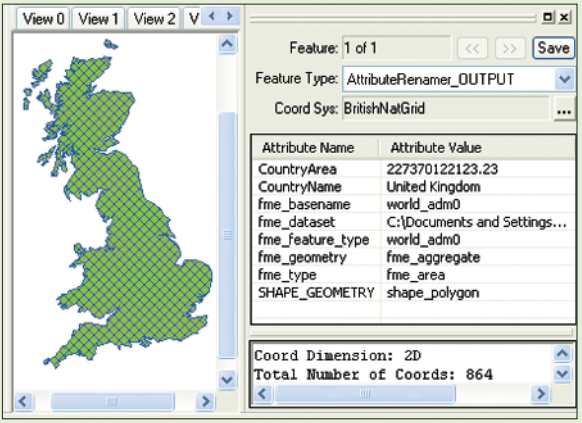
This button is color-coded to show the status of the settings.

-  If the Properties button is the same color as the transformer, you can use the transformer with its existing parameters.
-  A yellow Properties button indicates that the transformer contains default settings, but you have not yet accepted them. You can use the transformer in this state, but your workspace may produce unexpected results.
-  A red Properties button means that there is at least one parameter for which FME cannot supply a default value. You must provide a value for all required fields before you can use the transformer in the workspace.

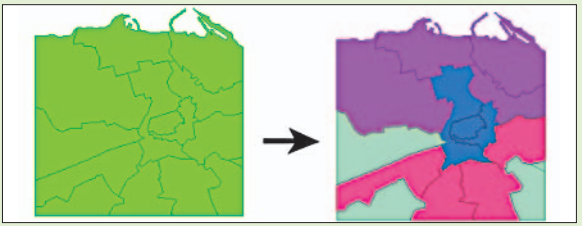
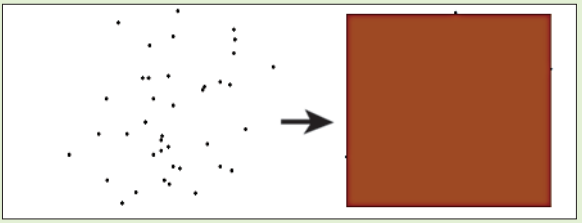
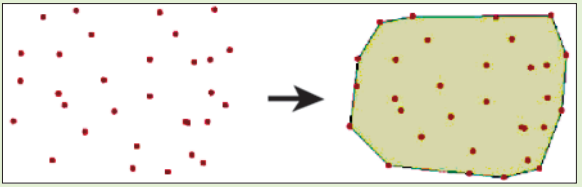
When you click a Properties button, the dialog that appears will usually have some of the common elements shown in this example. The content of this dialog depends on the transformer, and sometimes even on connections to the transformer. Most transformers have some common user interface elements, however, and those are described here.

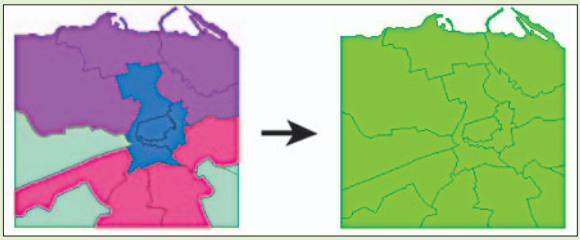
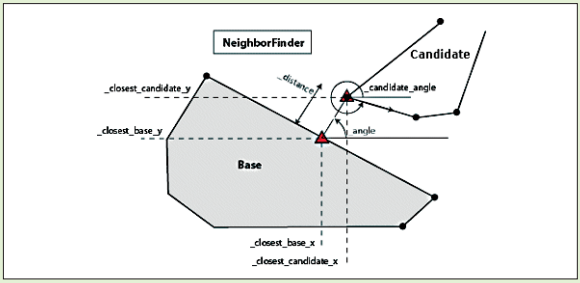


3D – These transformers create and modify three-dimensional surface and solid geometries.	
CSGBuilder	Creates Constructive Solid Geometry (CSG) from pairs of solid geometry features. 
CSGEvaluator	Replaces the geometry of a feature that has CSG. 
Extruder	Creates long, surface or solid geometries with a fixed cross-sectional profile taken from the original geometry of the feature. 
FaceReplacer	Replaces the geometry of a feature from donut or polygon to face. A face is a planar area in 3D space. The planar structure can be a raster, a polygon, or a donut.
NEW **MeshMerger**	Merges mesh features (features with IFMEMesh geometries) into a single output mesh.
SurfaceReverser	Reverses surfaces and solids. On surfaces, it will reorder the coordinates of the surface such that the normal of the output surface is the opposite of the input surface. Vertex normals that exist on the surface will be also be reversed. On solids, it will reverse the underlying surfaces, in effect causing the solid to be turned inside-out.

CALCULATORS – These transformers calculate a value and supply it to a new attribute on a feature.	
AngularityCalculator	Calculates the angularity of a linear or area feature. Angularity indicates the degree of curvature of a feature—the higher the value, the more curved its geometry.
AreaCalculator	Calculates the area of a polygonal object and stores the value in an attribute. The area is calculated in square ground units (the units of the feature's coordinates). 
AttributeRounder	Rounds off an attribute to the specified number of decimal places. `143.178435 ⟶ 143.18`
BaseConverter	Converts an attribute's value from one numeric system (base) to another, putting the resulting value in a new attribute. `Decimal 958713 ⟶ Hex EA0F9`
BoundsExtractor	Extracts the minimum and maximum values of the feature's coordinates into new attributes.
CircularityCalculator	Calculates the circularity of an area feature, which indicates how elongated the feature is.
CoordinateConcatenator	Retrieves the value of all of the feature's coordinates into an attribute, separated by the delimiter characters.
CoordinateCounter	Stores the number of a feature's coordinates into an attribute.
CoordinateExtractor	Retrieves the value of the x, y, and z coordinates at the specified index into attributes.
Counter	Adds a numeric attribute to a feature and assigns a value.
CRCCalculator	Calculates a Cyclic Redundancy Check (CRC) value as directed for a feature and places that value into the specified attribute.

DateFormatter	Reformats and replaces date or time strings into a new date format. The source string can be in almost any date and/or time format. Some valid examples include: • 20091206 15:05 • 20091206150500 • December 6, 2009 • 06 December 09, 15:05 • 3:05pm
DecimalDegreesCalculator	Calculates a decimal degree value from separate degrees, minutes, and seconds (DMS) values, stored in attributes.
DEMDistanceCalculator	Calculates the distance between a number of input vector lines and the elevation values of a reference DEM raster. Outputs a new DEM raster per input line. The data contained in the resulting DEM consists of the 3D distance between the line being considered and the corresponding point on the reference DEM.
DensityCalculator	Determines the density of a group of CANDIDATE features based on the area of a corresponding AREA feature.
DimensionExtractor	Returns the dimension of the feature as a new attribute.
DMSCalculator	Calculates degrees, minutes, and seconds (DMS) from a decimal degrees value stored in an attribute.
ElevationExtractor	Extracts the elevation of the first coordinate and assigns it to the named attribute.
EnvironmentVariableFetcher	Fetches the specified environment variable and includes it in a new attribute.
ExpressionEvaluator	Evaluates an arbitrary Tcl 8.5.2 expression and returns the result in a new attribute.
HoleCounter	Adds a new attribute whose value is the number of holes in the feature.
InsidePointExtractor	Adds attributes holding the coordinates of a point guaranteed to be inside the area feature. The geometry of the feature is not changed by this transformer.
LeftRightSpatialCalculator	Computes relative position of the CANDIDATE input features relative to the BASE input features. The geometry of a CANDIDATE feature is restricted to point and area, whereas BASE features can only be lines.
LengthCalculator	Calculates the length of a feature and adds it as a new attribute.
ModuloCounter	Adds an attribute holding the next integer in a sequence and restarts the count at 0 whenever the sequence reaches a defined maximum value.
OrientationExtractor	Determines the feature's orientation and returns it in the specified Orientation Attribute.
RandomNumberGenerator	Generates a random number between the values in the Minimum Value and Maximum Value parameters. The random number is rounded to the number of digits specified in the Decimal Places parameter.
SpatialRelator	Determines topological (spatial) relationships between sets of features. It tags, but otherwise does not change features when they have certain relationships, such as touches, overlays, intersects, and so forth.
StatisticsCalculator	Calculates statistics based on a designated attribute of the incoming features.
TextureCoordinateSetter	Assigns texture coordinates to surfaces.

COLLECTORS – These transformers operate on collections of features at the same time. The collection of features may be replaced by new features based upon them, have their attributes or geometries merged, or have their orders altered.	
2DGridAccumulator	Replaces the input features with a grid of two-dimensional point or polygon features that have the specified spacing and which cover (at least) the bounding box area of all features that enter the transformer.
Aggregator	Combines feature geometries into aggregates. One aggregate feature is output for each unique combination of values of the attributes specified in the Group By parameter. 
NEW **Amalgamator**	Generalizes polygonal input by connecting nearby geometries. It accepts polygonal geometries (including donuts) as input, and produces triangles that join input features into connected pieces, or amalgams.
AttributeAccumulator	Combines feature attributes. One feature is output for each unique combination of values of the attributes specified in the Group By parameter. The output feature will have no geometry.
BoundingBoxAccumulator	Takes a set of point, linear, polygonal, and/or aggregate features and creates a two-dimensional bounding box, which contains all features. 
CommonSegmentFinder	Tests to see which of the CANDIDATE features have even one segment in common with any BASE feature.
ConvexHullAccumulator	Creates convex hulls for groups of features. 

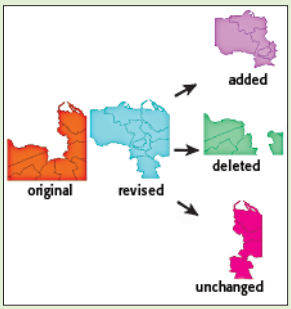
Deaggregator	Decomposes an aggregate feature into its components. 
FeatureHolder	Stores incoming features until they have all arrived and then releases them in their original order.
FeatureMerger	Moves the attributes and/or geometry from one feature to another feature.
NeighborFinder	Finds the closest CANDIDATE feature within a specified maximum distance of each BASE feature. 
NeighborhoodAggregator	Creates aggregates of features based on their proximity to each other. 
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.
Sorter	Sorts features by an attribute's value.

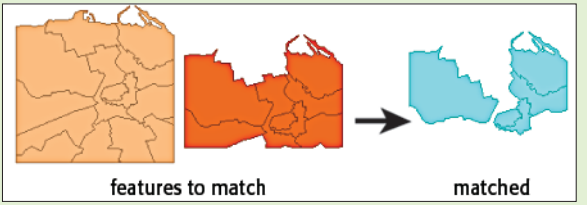
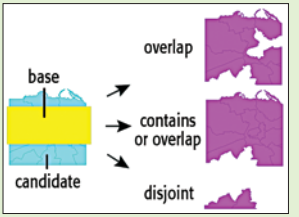
COORDINATE SYSTEMS – These transformers relate to coordinate systems and reprojection.	
AttributeReprojector	Reprojects attributes from one coordinate system to another.
CoordinateSystemDescription Converter	Converts coordinate systems between FME and AutoDesk WKT, EPSG, ESRI WKT, MapInfo, OGC WKT, Oracle SRID, and PROJ.4 representations.
CoordinateSystemExtractor	Retrieves the feature's coordinate system into an attribute.

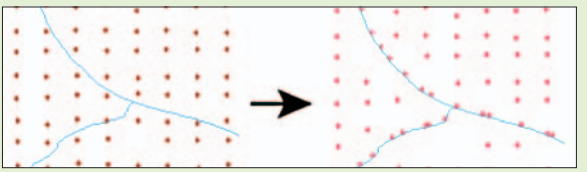
CoordinateSystemRemover	Removes the coordinate system from all input features. This transformer does not reproject features or otherwise modify their geometry.
CoordinateSystemSetter	Tags all features with the specified coordinate system. It does not reproject features or otherwise modify their geometry.
NEW **CsmapReprojector**	Reprojects featurecoordinates from one coordinate system to another using the CS-MAP library.
ESRIReprojector	Reprojects feature coordinates from one coordinate system to another using the ESRI® reprojection library.
GridInQuestReprojector	Reprojects feature coordinates from one coordinate system to another using the Grid InQuest reprojection library.
GtransAttributeReprojector	Reprojects attributes holding coordinate values from one coordinate system to another using the Gtrans reprojection library (from the National Land Survey of Sweden) and the specified translation file.
GtransReprojector	Reprojects features to and from SWEREF99 using the Gtrans reprojection library (from the National Land Survey of Sweden) and the specified translation file.
LatLongToMGRSConverter	Calculates a Military Grid Reference System (MGRS) code based on the latitude and longitude values supplied in a feature's attributes.
MGRSGeometryExtractor	Calculates a Military Grid Reference System (MGRS) code based on the feature's geometry.
MGRSGeometryReplacer	Converts MGRS code to longitude and latitude coordinates. The geometry of an input feature is replaced with a point at the longitude/latitude values obtained from the MGRS code.
MGRSToLatLongConverter	Converts MGRS code to longitude and latitude coordinates. Converts MGRS code to longitude and latitude coordinates.
ReprojectAngleCalculator	Converts a given angle from one coordinate system to another. The transformer calculates the reprojected angle of a line starting at the first coordinate in the feature, with the given length and angle.
ReprojectLengthCalculator	Converts a given distance from one coordinate system to another. The transformer calculates the reprojected length of a line starting at the first coordinate in the feature, with the given length and angle.
Reprojector	Reprojects feature coordinates from one coordinate system to another. 

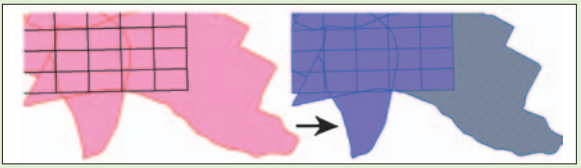
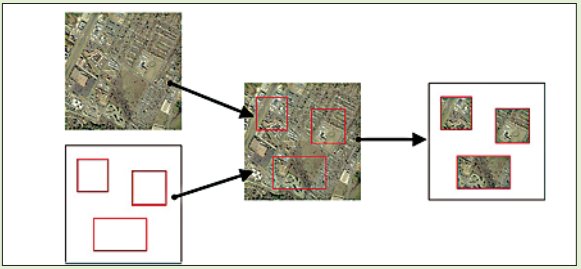
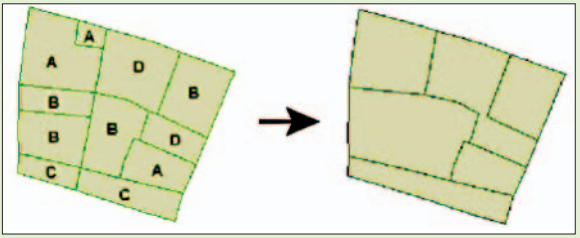
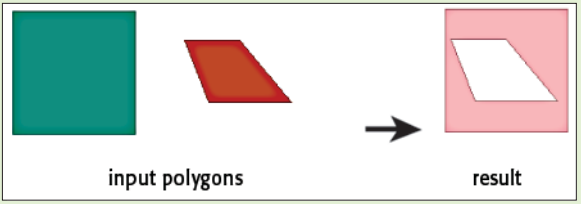
DATABASE – These transformers allow interaction with external databases. Data can be extracted from databases and merged into the feature stream, or merged onto features. You can also execute arbitrary SQL statements.	
ArcSDEQuerier	Performs queries on an ArcSDE spatial database. The queries can have both a spatial and a nonspatial component.

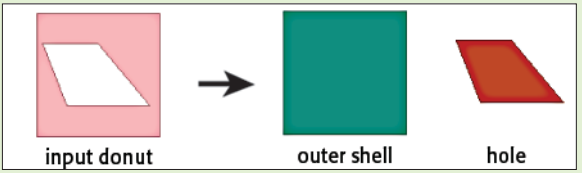
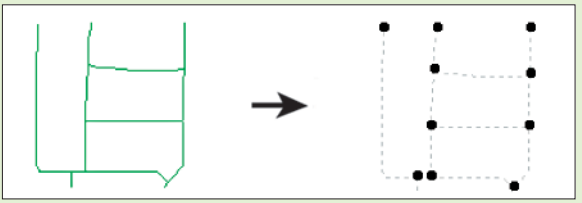
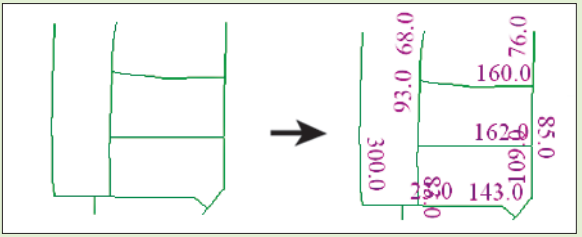
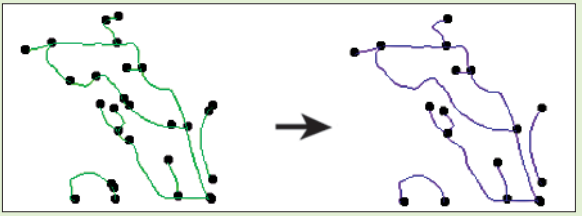
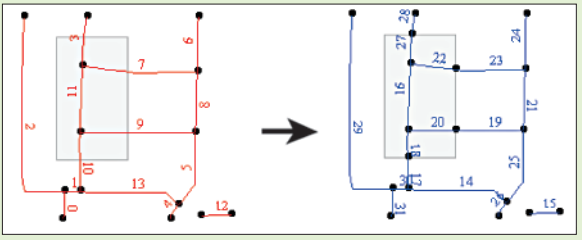
NEW **FeatureReader**	Performs queries against any FME-supported format. The queries can have both a spatial and a nonspatial component.
Joiner	Joins attributes from an external database to other spatial or nonspatial features as they are processed through a translation. Most popular databases are supported.
OracleQuerier	Performs spatial queries against an Oracle Spatial database. The queries can have both a spatial component and a nonspatial component.
SchemaMapper	Maps the schema (attributes and feature types) of features based on a schema mapping table.
NEW **SQLCreator**	Generates FME features from the results of a SQL query against a database. One FME feature is created for each row of the results of the SQL Query.
SQLExecutor	Runs an arbitrary SQL statement against a database.
NEW **SQLQuerier**	Performs SQL queries against a database. One query is issued to the database for each feature that enters the transformer. The results of the query are then output.

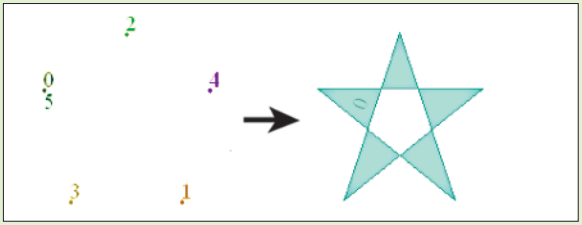
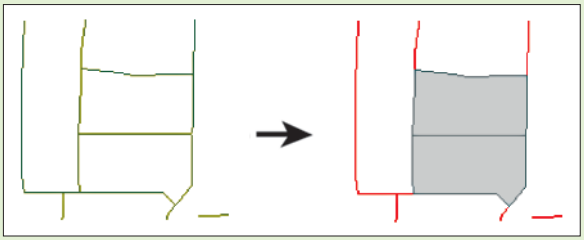
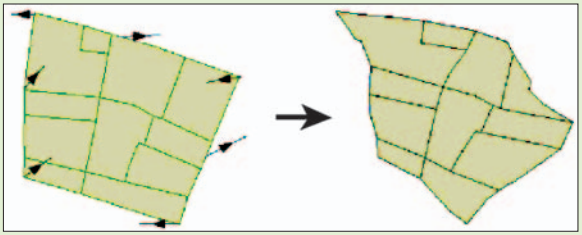
FILTERS – These transformers perform tests on feature geometry and/or attributes, and allow the feature to be routed to different destinations depending on the outcome of the test.	
AggregateFilter	Routes features differently depending on whether their geometry consists of an aggregate of several primitive geometries or a simple, single piece of geometry.
AttributeFilter	Routes features to different output ports depending on the value of an attribute.
AttributeRangeFilter	Performs a lookup on a range-based lookup table and routes the feature to the appropriate output port.
ChangeDetector	Detects changes between two sets of input features. 
ConvexityFilter	Determines whether areas, surfaces, and solids are convex or concave. A polygon is simple when it is not self-intersecting and has a non-zero area. Simple polygons are convex if every internal angle is less than or equal to 180 degrees. All other polygons are considered concave.
DuplicateRemover	Detects duplicate features based on the value of a key attribute.
FeatureTypeFilter	Routes input features to different output ports based on their original feature type.
GeometryFilter	Routes a feature based on its geometry type.

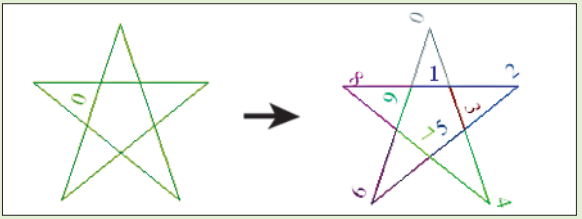
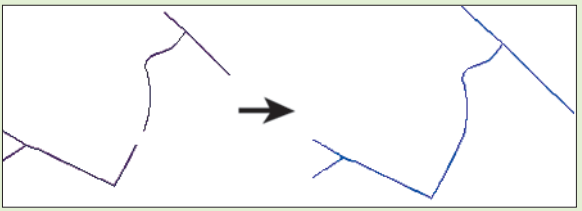
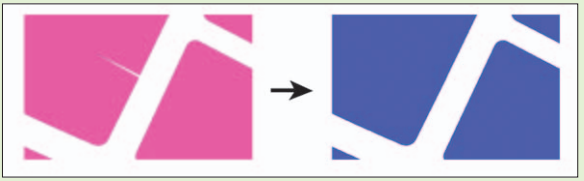
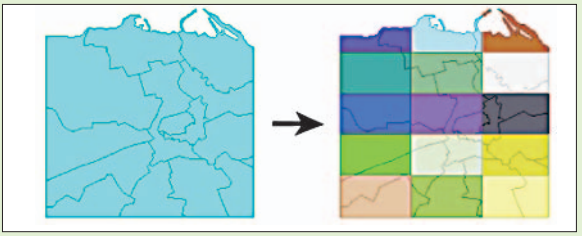
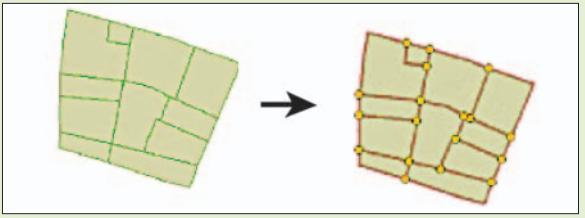
GeometryOGCValidator	Evaluates the simplicity or validity of a geometry feature and routes the feature according to the outcome of the tests.
IndividualGeometriesFilter	Filters aggregate features based on the type of aggregate.
LicenseChecker	Checks whether the license file is valid and the specified product name is licensed, based on a vendor key and vendor registration code.
Matcher	Detects features that are matches of each other. Features are declared to match when they have matching geometry, matching attribute values, or both. 
PlanarityFilter	Filters features based on their planarity. To be planar, a geometry must have all of its points situated in the same plane.
Sampler	Preserves either a total number of features or a sampling of features, depending on the Sampling Type selection.
SpatialFilter	Filters features based on spatial relationships. Each input CANDIDATE feature is compared against all BASE features, based on the selected tests to perform. 
Tester	Evaluates one or more tests on a feature and routes the feature according to the outcome of the tests. The tests can consist of any FME-allowed operands.
TestFilter	Filters features by test conditions to one or more output ports.

GEOMETRIC OPERATORS – These transformers operate on the geometry of individual features, or groups of features. A wide variety of operations are available, including overlays, snapping, line labeling, clipping, and intersection.	
AffineWarper	Performs warping operations on the spatial coordinates of features. It is used to adjust a set of observed features so they more closely match some set of reference features.
AnchoredSnapper	Takes a series of features that match the input specification and performs snapping on the features that lie within the specified tolerance from other features that match the input specification. 

AreaBuilder	Takes a set of topologically connected linework and creates topologically correct polygon features where the linework forms closed shapes.
AreaOnAreaOverlayer	Performs an area-on-area overlay. All input areas are intersected against each other, and resulting area features are created and output. The resulting areas have all of the attributes from all the original features in which they are contained. 
Clipper	Performs a geometric clipping operation. 
Dissolver	Dissolves area features by removing common boundaries to create larger areas. 
DonutBridgeBuilder	Builds connections between donut holes with the outer boundary of a donut, resulting in a polygon-equivalent representation of the input donut.
DonutBuilder	Cuts holes in polygonal features by making polygons completely enclosed in other polygons into holes of the containing polygon. 

DonutHoleExtractor	Splits an area feature that has holes into its component rings. 
Intersector	Computes intersections between all input features, and breaks lines and polygons wherever an intersection occurs. 
Labeller	Interpolates labels along a linear or polygonal feature. 
LineJoiner	Takes non-intersecting lines and connects them into longer lines whenever doing so does not remove a significant node. 
LineOnAreaOverlayer	Performs a line-on-area overlay. Each input line is split at any area boundaries it intersects. 

LineOnLineOverlayer	Performs a line-on-line overlay. During the overlay, all input lines are intersected against each other, and resulting line features are created and output.
NeighborColorSetter	Assigns colors to areas in a coverage such that adjacent areas are colored differently and the total number of colors used is kept small.
NetworkTopologyCalculator	Finds the connected lines that belong to the same network graph.
PathBuilder	Connects input linear features (arcs and lines) in the order they enter, forming path features.
PointConnector	Connects input point features in the order they enter, forming linear or polygonal features. 
PointOnAreaOverlayer	Performs an overlay of points on areas.
PointOnLineOverlayer	Performs an overlay of points on lines. Each input line is split at its closest place to any point within the specified point tolerance.
PointOnPointOverlayer	Performs an overlay of points on points.
PolygonBuilder	Forms polygons from lines. 
RubberSheeter	Performs warping operations on the spatial coordinates of features. It is used to adjust a set of observed features so they more closely match a set of reference features. 

SelfIntersector	Checks each feature and removes self-intersections. 
Snapper	A Snapper snaps end-points or vertex-points of features together if they are within a certain distance of each other and (optionally) if they have one or more attributes in common. 
SpikeRemover	Cleans up feature geometries by removing spikes in two dimensions. 
Tiler	Chops the input features into a series of tiles. This transformer works with both raster and vector data. 
TopologyBuilder	Computes topology on input point, line, and/or area features. 
Triangulator	Breaks an input geometry into triangular units or a mesh.

INFRASTRUCTURE – These transformers provide interaction with the underlying FME translation engine facilities. These include functionality to log features, set feature colors, create individual features and grids of features from nothing, and invoke the FME Viewer on features flowing by.	
2DGridCreator	Creates a grid of two-dimensional point or polygon features, at the origin and uses the offsets specified. Each created feature has a row and column attribute that indicates its position in the grid.
AreaFillColorSetter	Sets the fill color for the feature's area. Formats that support color will then render the interior feature in the set color.
NEW **AttributeCompressor**	Compresses the values of specified attributes. Used in conjunction with the AttributeDecompressor.
AttributeCopier	Copies existing attributes to new attributes with the specified names. The existing attribute remains and a new attribute is created. The new attribute has a different name, but the same value.
AttributeCreator	Adds a number of attributes to the feature and supplies them with constant values. Any feature that enters the transformer emerges with a new set of user-defined attributes.
NEW **AttributeDecompressor**	Decompresses the values of the specified attributes that were compressed by the AttributeCompressor.
AttributeDereferencer	Copies the value of the attribute whose name is held in the source attribute to a newly created attribute.
AttributeExposer	Exposes hidden attributes so that they can be used by other transformers.
AttributeFileReader	Reads the contents of a file and stores them as the value for the specified attribute.
AttributeFileWriter	Writes the contents of the specified attribute to a file.
AttributeRenamer	Renames selected attributes.
AttributeSetter	Sets an existing attribute to a constant value or to the value of another attribute.
Cloner	Makes the specified number of copies of the input features and outputs all copies through its single output port.
Creator	Creates features using the parameters supplied and sends them into the workspace for processing.
NEW **CustomTransformerLooper**	Allows external looping over a selected linked custom transformer.
FeatureTypeExtractor	Adds an attribute containing the original feature type of a feature.
FMEFunctionCaller	Calls the specified FME function, optionally putting the resulting value in the Result Attribute.
GeometryNameExtractor	Retrieves the name of geometry and sets it on the specified attribute.
GeometryNameRemover	Removes the name of the geometry. The component parts of aggregate geometries remain unchanged.
GeometryNameSetter	Applies a name to the geometry of a feature.
IndividualGeometriesSetter	Provides the ability to set up an aggregate where each part is independent from the others and is its own complete geometry.
Logger	Logs each feature to the translation log. All attributes and geometry of the feature will be output.
ParameterFetcher	Adds an attribute to the feature and supplies it with the value of a previously published parameter.

PenColorSetter	Sets the pen color of the feature.
Player	Retrieves features stored in an FME Feature Store file and outputs them into the workspace.
PythonCaller	Processes FME Features using a Python function or class, which can be written in the PythonCaller's source code editor or stored in an external Python module.
PythonCreator	Generates FME features using a Python object referenced by the symbol name parameter. The handler object can be a Python function, a Python class, or any Python object that understands the FME Factory protocol.
RandomColorSetter	Sets a random color for each incoming feature.
Recorder	Saves a copy of all features that enter to a disk file.
SummaryReporter	Writes a summary report of incoming features to a disk file. Features are sorted before they are summarized.
SystemCaller	Runs a program and waits for it to exit before continuing the translation.
TCLCaller	Runs a Tool Command Language (Tcl) command and assigns its return value to an attribute.
Terminator	Causes the translation to end and prints the specified message in the translation log as the reason for the termination.
TransporterReceiver	Provides a mechanism to receive features from another FME workspace running in a different process, which may be located on the same or on a different computer. Used in conjunction with the TransporterSender.
TransporterSender	Provides a mechanism for transporting features to another FME workspace running in a different process, which may be located on the same or on a different computer. Used in conjunction with the TransporterReceiver.
VariableRetriever	Reads the specified variable and puts its value into the specified attribute. This variable must have been previously set using the VariableSetter transformer.
VariableSetter	Creates and sets the specified variable to the specified value. The variable can later be read back into an attribute using the VariableRetriever transformer.
Visualizer	Sends features to the FME Universal Viewer or the FME Data Inspector.

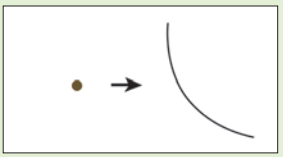
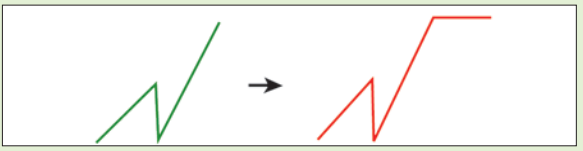
KML – These transformers manipulate feature geometry and/or attributes for output using the Google Earth™ KML Writer.	
KMLPropertySetter	Sets common properties for groups of vector and raster features that are destined for the Google Earth KML Writer.
KMLRegionSetter	Sets the region-related KML attributes for a group of features that are destined for the Google Earth KML Writer.
KMLStyler	Creates a common style for a group of features that are destined for the Google Earth KML Writer.
KMLTimeSetter	Sets the time-related KML attributes for a group of features that are destined for the Google Earth KML Writer.

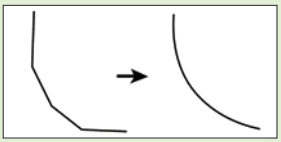
KMLTourBuilder	Generates a KML Tour from the input features. The tour consists of tour stops that correspond to each input feature.
KMLViewSetter	Sets the view-related KML attributes for a group of features that are destined for the OGCKML Writer. Creation of LookAt or Camera views are supported.

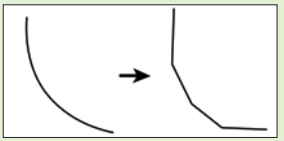
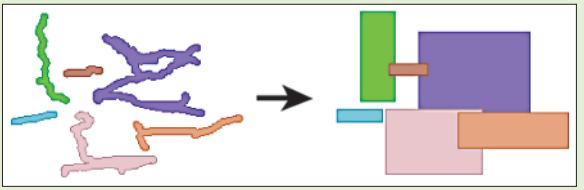
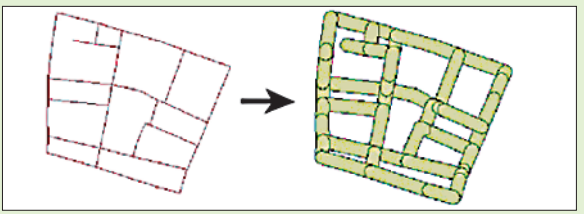
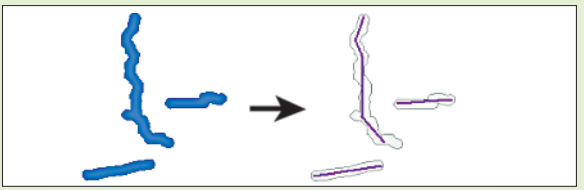
LINEAR REFERENCING – These transformers work with linear referencing data structures on FME features. Transformers are provided for creating and applying measure-related information held in attributes onto the geometry of FME features.	
LengthToPointCalculator	Calculates the length of a feature from its start until the closest spot to a point and adds it as a new attribute. The point coordinates are taken from attributes in the original feature.
MeasureExtractor	Extracts the measures of geometries that match the given type, and places them in attributes or list attributes.
MeasureGenerator	Creates a set of measures attached to the geometry of the feature, where each value is the distance from the start of the line to that vertex, multiplied by the given Multiplier.
MeasureRemover	Removes measures from a feature's geometry.
MeasureSetter	Sets measures on a point, line, arc, area geometry, or a vertex of a linear geometry to attribute values of given attributes or list attributes.
Snipper	Shortens the geometry of a line feature by snipping off specified distances, indices, or measure values from the ends. It operates on features with simple line geometry and polygons without holes.

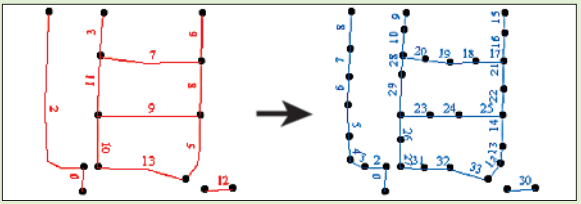
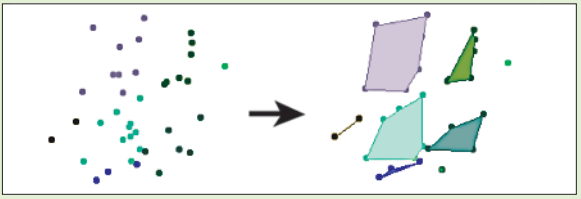
LISTS – These transformers operate on FME attribute lists. Transformers are provided for creating, exploding, searching, and extracting from FME attribute lists.	
AttributeExploder	Creates a new pair of attributes (attribute name/attribute value) from each attribute on the input feature and either outputs these on a new feature or adds them as a list element to the original feature. In both cases, it is possible to either conserve or delete the original attributes and geometry.
ListBuilder	Combines attributes of the input features into a single list structure.
ListConcatenator	Concatenates all values of a list into a single attribute.
ListCopier	Copies a complete attribute list, including all nested attributes, from one list name to another.
ListDuplicateRemover	Removes all duplicate values from a list attribute. In the resulting list, only distinct values for the list attribute will be present.
ListElementCounter	Stores the number of member elements found in the specified list into an attribute.
ListExploder	Explodes each list member on each input feature out into its own feature.
ListHistogrammer	Computes a histogram of the values found in a list and returns these in a new list attribute on the feature.

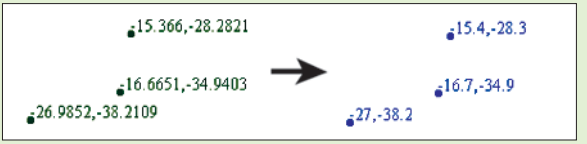
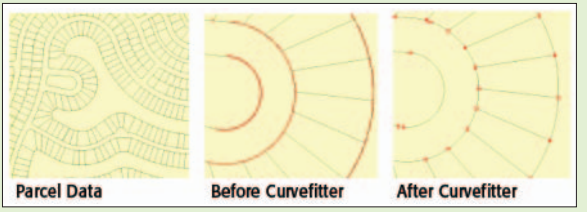
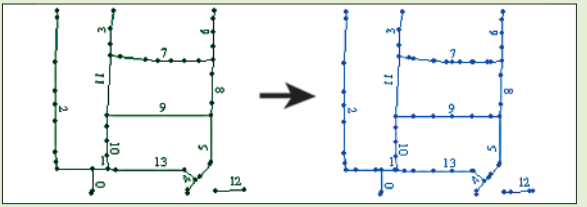
ListIndexer	Demotes the attributes of the list element specified by the index to become main attributes of the feature.
NEW **ListKeeper**	Keeps selected list attributes from incoming features, and removes the rest.
ListPopulator	Takes a series of user attributes attached to a feature and creates a list attribute from them. Example: *myattrib0* *myattrib1* *myattrib2* becomes a list *myattrib{}* containing entries *myattrib{0}, myattrib{1}, myattrib{2}*
ListRangeExtractor	Extracts the minimum and maximum values found in a list.
ListRemover	Removes a list from incoming features.
ListRenamer	Renames the components of a list or the list name.
ListSearcher	Searches a list to find a value and returns the index of the value in the list.
ListSorter	Sorts the elements of the given list into a new list.
ListSummer	Computes the sum of all elements of a list.

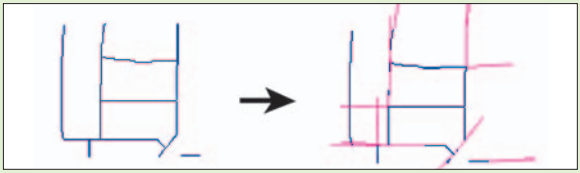
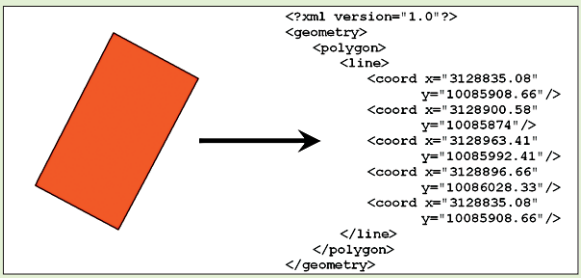
MANIPULATORS – These transformers modify (manipulate) the geometry or attributes of individual features in isolation from other features.	
2DArcReplacer	Replaces the geometry of the feature with a two-dimensional arc whose shape is set by the parameters, which can be either constant floating point values or the values of existing attributes. 
2DBoxReplacer	Replaces the geometry of the feature with a two-dimensional box whose minimums and maximums are fixed values or are taken from attributes in the original feature.
2DEllipseReplacer	Replaces the feature's geometry with a two-dimensional ellipse whose shape is set by the parameters, values, or the values of existing attributes.
2DForcer	Removes any elevation z coordinates that may or may not have been present on the original feature.
2DPointAdder	Adds a two-dimensional point as the last vertex of the feature. If the feature turns into a closed polygon as a result of adding the point, it will be tagged as an area feature. Otherwise, it will be tagged as a line. 

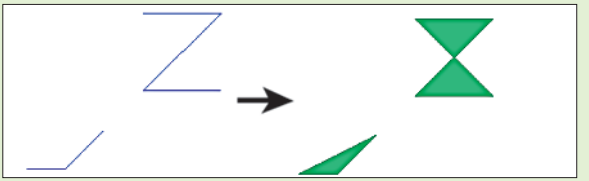
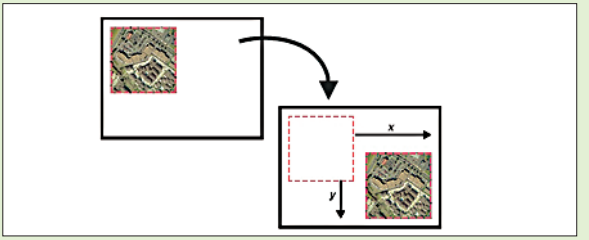
2DPointReplacer	Replaces the feature's geometry with a two-dimensional point whose coordinates are taken from attributes in the original feature. X = 1353.65 Y = 3911.31 ⟶ ●
3DAffiner	Performs 3D affine transformation on the coordinates of the feature. An affine transformation preserves parallelism of lines and planes in geometry. Affine transformations include translations, rotations, scalings, and reflections.
3DArcReplacer	Replaces the feature's geometry with a two-dimensional arc whose shape is set by the parameters, which can be either constant floating point values or the values of existing attributes.
3DForcer	Turns two-dimensional data into three-dimensional data by adding a z-value to every coordinate.
3DInterpolator	Interpolates elevation values along a non-aggregated linear feature from a starting value to an ending value. The resulting feature's elevation monotonically increases (or decreases) from the starting value to the ending value. If the feature was two-dimensional, it becomes three-dimensional. If the feature was three-dimensional, its previous elevations are removed and replaced.
3DPointAdder	Adds a three-dimensional point as the last vertex of the feature. If the feature turns into a closed polygon as a result of adding the point, it will be tagged as an area feature; otherwise,it will be tagged as a line. However, if this was the first point added, it will be tagged as a point.
3DPointReplacer	Replaces the geometry of the feature with a three-dimensional point whose ordinates are taken from attributes in the original feature. If the feature was originally a text feature, it remains a text feature, but its insertion point is moved. All other features become point features. X = 1353.65 Y = 3911.31 ⟶ ● Z = 21.14
Affiner	Performs an affine transformation on the feature's coordinates.
AngleConverter	Converts angles of a feature's geometry and/or attributes from one representation to another.
ArcEstimator	Replaces the geometry of the feature with a two-dimensional circular arc whose shape is estimated from the first, middle, and last point of the linear feature passed in. 
ArcPropertyExtractor	Sets the given attributes to the properties of an arc geometry and works on a single feature at a time.
ArcPropertySetter	Modifies the properties of an arc geometry.
ArcSDEGridSnapper	Simulates the ArcSDE conversion on a feature by performing ArcSDE translation, scaling, and coordinate snapping. Also removes duplicate vertices that result from snapping multiple, formerly separate, vertices to the same grid point.

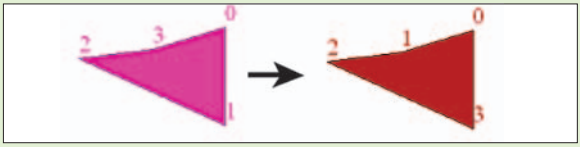
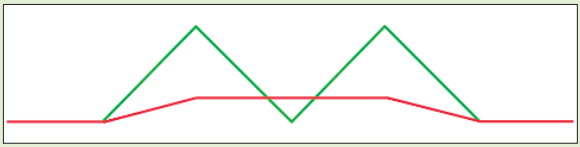
ArcStroker	Converts arc features into lines replacing the feature geometry with a series of edges interpolated along the arc boundary. Ellipse features are converted into polygons by interpolating edges along the elliptical boundary. 
AttributeExpressionRemover	Removes all attributes on incoming features that match a given regular expression. It can also be used to remove large numbers of attributes that have common naming.
AttributeKeeper	Removes all attributes from the feature, except the ones that are selected from the attribute list. This transformer is useful when features have large numbers of unnecessary attributes.
AttributePrefixer	Adds a prefix or suffix to all attributes coming into the transformer. This transformer is useful when you need to quickly rename all attributes. It can be placed before other transformers that may overwrite incoming attributes.
AttributeRangeMapper	Performs a lookup on a range-based lookup table and stores the resulting value, or writes the value to, a new output attribute.
AttributeRemover	Removes the selected attributes from the feature.
BoundingBoxReplacer	Replaces the geometry of the feature with either its two-dimensional bounding box or its two-dimensional minimum oriented bounding box. 
Bufferer	Replaces the geometry of a feature with one that represents the original, padded by a specified width. Each point in the output geometry is the specified distance away from the original geometry. 
CenterLineReplacer	Replaces an area feature with its medial axis, straight skeleton, or a centerline. This transformer works best with long, narrow areas. 

CenterOfGravityReplacer	Replaces the feature's geometry with a point that is the center of mass distribution of the feature. The resulting point may be far outside of the original feature, depending on the feature's shape. 
CenterPointReplacer	Replaces the feature's geometry with a point that is in the center of the feature's bounding box. 
Chopper	Ensures that all features output have less than or equal to the specified maximum number of vertices. This is useful when you're outputting to a format that has limitations on the number of points in lines or areas. It can also be used to crudely simplify complex objects. 
ConvexHullReplacer	Replaces the geometry of the feature with a polygon representing its convex hull. 
CoordinateRemover	Removes one or more coordinates from the geometry of the feature.

CoordinateRounder	Rounds off the feature's coordinates to the specified number of decimal places. Any consecutive points that become duplicates as a result of the rounding are thinned by removing the redundant points. 
NEW **CoordinateSwapper**	Swaps coordinate axes of the input features.
Curvefitter	Smoothes lines derived from line segments, points, or raster data, and replaces a series of line segments with the optimal combination of straight lines and embedded arc segments required to create smooth curving lines. This process provides a truer representation of real-world features and can reduce file sizes by up to 80%. In addition to processing simple line features, the Curvefitter preserves feature topology when smoothing boundaries of adjacent area features. 
Densifier	Adds vertices to the feature by interpolating new coordinates along its definition at some interval of distance. The interval may be along only one of the two primary axes, or it may be along the length of the line segments. 
Displacer	Solves proximity conflicts between features using a variant of the Nickerson displacement algorithm.
DuplicateCoordinateRemover	Checks all geometry elements that are lines for duplicate coordinates. Any consecutive coordinates with the same location are reduced to a single coordinate.
EllipsePropertyExtractor	Sets the given attributes to the properties of an ellipse geometry.
EllipsePropertySetter	Sets the properties of an ellipse geometry as specified.

Extender	Creates two-point extensions to linear features that extend the feature by a user-specified length. This transformer can also output the original feature with the first and last segments stretched by a user-specified amount. 
FilenamePartExtractor	Extracts a part of a filename path and returns the result as a string.
Generalizer	There are four algorithm types: • Generalizing algorithms reduce the density of coordinates by removing vertices. • Smoothing algorithms determine a new location for each vertex. • Measuring algorithms calculate the location of points and return a list of these points (for example, to measure the sinuosity of a feature). • Fitting algorithms replace the original geometry completely, with a new feature fitted to a specified line (for example, to minimize the orthogonal distance to the original).
GeometryCoercer	Resets the geometry type of the feature.
GeometryExtractor	Extracts the geometry of a feature according to the setting of the geometry encoding parameter. The resulting encoded geometry is added to the feature in an attribute. This attribute can later be restored as the feature's geometry using the GeometryReplacer transformer. 
NEW **GeometryInstantiator**	Instantiates a geometry instance into a specific concrete instantiation of geometry definition.
GeometryRefiner	Performs the following refinements on the feature's geometry: • Any homogeneous IFMEAggregate becomes a multi (IFMEMultiCurve, IFMEMultiArea, IFMEMultiPoint, or IFMEMultiText). • Any IFMEAggregate or multi with only one member is replaced by its single part. • Any IFMEDonut with no holes becomes an IFMEPolygon or IFMEEllipse. • Any IFMEPath with only one segment is replaced by that segment. • Consecutive IFMELine segments within an IFMEPath are combined.
GeometryRemover	Completely removes the feature's geometry; for example, when you want to turn spatial data into nonspatial data.
GeometryReplacer	Replaces the feature's geometry according to the setting of the geometry encoding parameter. This transformer is typically used to restore geometry previously extracted into an attribute by the GeometryExtractor.

GeometryTraitExtractor	Copies the specified geometry traits into feature attributes with the same names.
GeometryTraitRemover	Removes geometry traits from a feature's geometry.
GeometryTraitSetter	Copies attributes from a feature into geometry traits.
GMLFeatureExtractor	Constructs GML2 documents from the input features and stores them in the specified attribute for the features that are output by the GML2 port.
GMLFeatureReplacer	Constructs features out of GML documents that are stored in an attribute of the input features.
InsidePointReplacer	Replaces the geometry of the area feature with a point that is guaranteed to be inside the area.
LabelPointReplacer	Replaces the feature's geometry with a label point. For polygons, the text is guaranteed to be inside the original object. For lines or points, the text is guaranteed to be on the original object. 
LineCloser	Turns input linear features into areas by adding their start point as the end point. 
MinimumAreaForcer	Ensures that features with polygon geometry have an area that is equal to, or in excess of, the specified minimum area.
MinimumSpanningCircle Replacer	Replaces feature's geometry with a polygon representing its minimum spanning circle. The minimum spanning circle is defined as the smallest circle that encloses all vertices of the passed in feature.
Offsetter	Adds offsets to the feature's coordinates so that the feature shifts by the specified amount. 

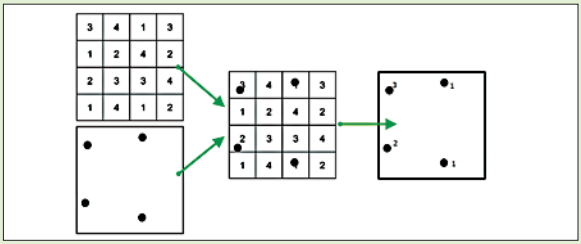
Orientor	Adjusts the orientation of a polygonal feature or the direction of a linear feature. 
PartCounter	Returns the number of parts in the geometry. For multis and aggregates, this is the number of parts, and for paths, this is the number of segments.
PathSplitter	Decomposes a path feature into its component segments. Each output feature contains a copy of the source feature's attributes.
PDFStyler	Sets the common PDF style attributes for a group of features destined for the GeoSpatial PDF Writer.
Rotator	Rotates features in a counterclockwise direction about the specified point by the rotation angle (measured in degrees).
Scaler	The Scaler scales objects to make them bigger or smaller.
SecondOrderConformer	Performs a second-order conformal transformation on the feature's geometry. Depending on the input geometry, a 2D or 3D transformation is performed.
NEW **SherbendGeneralizer**	Uses the Sherbend algorithm to simplify lines by reducing unnecessary details based on the analysis of the line's bends. The generalization process may eliminate, reduce, or combine bends, while resolving conflicts. In this example, three bends are combined into one: 
TextAdder	Sets the feature's geometry to text with the previous geometry as the text location.
TextLocationExtractor	Sets a text feature's geometry to the location of the text.
TextPropertyExtractor	Sets the given attributes to a text geometry's properties.
TextPropertySetter	Sets the properties of a text geometry to the specified properties.
TextStroker	Takes as input a text string, rotation, height and width multiplier, then outputs aggregates that describe the outline of the text.

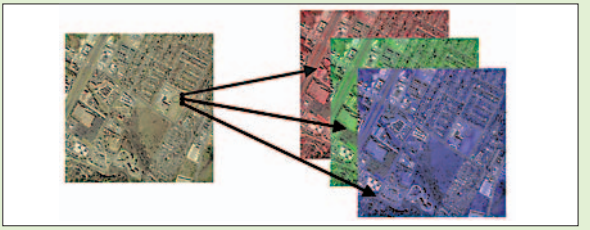
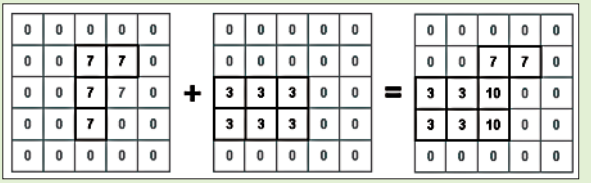
MRF – These transformers repair geometry, particularly during data migration from CAD to GIS. They are built upon the MRFCleanFactory, which is an integration of MRF Geosystems Corporation's cleaning technology into FME. All of the transformers in this category are available as an extra-cost package from Safe Software.	
MRF2DCleaner	Fixes geometric problems in input data, such as line overshoots and undershoots within the user-specified tolerance. It is useful for multi-layer and multi-tolerance two-dimensional data cleaning.
MRF2DConflator	Changes a feature's geometry to match that of another when both have approximately the same shape and location, and have matching endpoints.

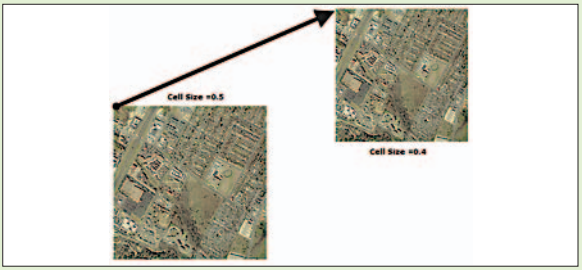
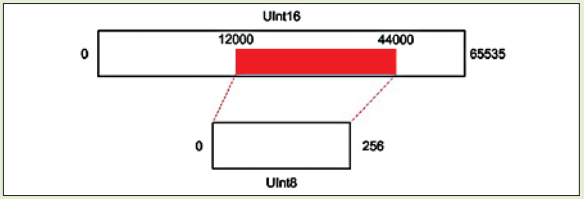
MRF2DDangleRemover	Removes features that have at least one free endpoint and have lengths smaller than the specified amount.
MRF2DDuplicateRemover	Deletes duplicated features. Features are considered to be duplicates if their geometries are within tolerance. Only features with a smaller tolerance remain after cleaning.
MRF2DExtender	Extends arcs and lines that are within the specified tolerance to correct undershoots while maintaining line-work direction.
MRF2DGeneralizer	Removes a number of vertices from lines. The number of vertices removed is controlled by a weeding tolerance.
MRF2DIntersector	Computes intersections between all input features, breaking arcs and lines wherever an intersection occurs.
MRF2DJoiner	Joins connected features to form longer ones. A pair of linear features becomes a candidate for joining only when the two are connected at a given node or end point.
MRF2DShortGeometryRemover	Removes features that have lengths smaller than the specified tolerance.
MRF3DCleaner	Fixes geometric problems in input data such as line overshoots and undershoots within the user-specified tolerance. It is useful for multi-layer and multi-tolerance three-dimensional data cleaning.

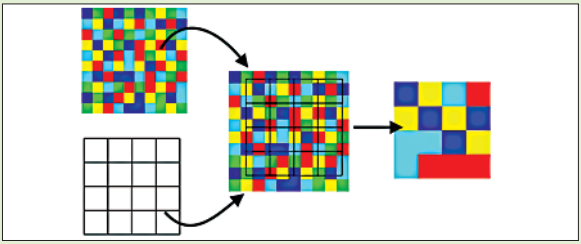
NETWORK – These transformers operate on linear features that are connected in a network, performing operations such as priority calculation and orientation correction.	
NetworkCostCalculator	Computes and assigns the cost of the shortest path from a source object to each connected object as the Z-values or measure values of the input features.
NetworkFlowOrientor	Fixes the flow (direction) of each edge or linear feature in the network to fit the downstream direction to the destination node.
ShortestPathFinder	Computes the shortest path from a source node to a destination node in a network based on the length of the input or the weight of the edges.
StreamOrderCalculator	Computes the order (Strahler or Horton) of streams in a river network.
StreamPriorityCalculator	Calculates the primary and secondary streams of multiple stream networks.

POINT CLOUD – These transformers create, use, and output point cloud features. They operate only on data consisting of point clouds.		
NEW NEW	**PointCloudCoercer**	Decomposes all point clouds into points.
NEW	**PointCloudCombiner**	Combines multiple point clouds into a single point cloud.
NEW	**PointCloudCreator**	Creates a new point-cloud feature with the specified size and components and sends it into the workspace for processing.
NEW	**PointCloudPropertyExtractor**	Extracts the properties of a point-cloud feature and exposes them as attributes.
NEW	**PointCloudSplitter**	Splits a single point-cloud feature into multiple point-cloud features.
NEW	**PointCloudThinner**	Outputs point-cloud features that have fewer points than the original input features.

RASTERS – These transformers create, use, and output rasters. They operate on data consisting of a regularly spaced grid of values.	
ImageRasterizer	Draws input point, line, and polygon features onto a color raster filled with the background color.
NumericRasterizer	Draws input point, line, and polygon features onto a numeric raster filled with the background value. The z coordinates of the input vector features are used to generate pixel values.
PointOnRasterValueExtractor	Takes in a number of point features and a single reference raster. For each input point feature, a point is created after the reference raster and output. 
RasterBandAdder	Adds a new band to a raster. The added band will have the same value in all cells and the same raster-level properties as other bands in the raster.
RasterBandCombiner	Merges multiple overlapping raster features into a single raster feature.
RasterBandInterpretation Coercer	Alters the underlying interpretation of the selected bands of the raster geometry on the input features, using the specified conversion options.
RasterBandKeeper	Removes all bands of a raster, except for those that are selected. The RasterSelector can be used to modify the selection.
RasterBandMinMaxExtractor	Extracts the band minimum and maximum values, palette minimum and maximum keys, and palette minimum and maximum values of a raster feature, and exposes them as attributes.
RasterBandNameSetter	Sets the name of selected bands on a raster.
RasterBandNodataRemover	Removes any existing nodata values from the selected bands of a raster feature.
RasterBandNodataSetter	Identifies the nodata value on a raster feature at the band level. All selected bands on an input raster feature receive the same specified nodata value.
RasterBandOrderer	Specifies the order of bands in a raster. Bands are reordered according to the input band indices.
RasterBandPropertiesExtractor	Extracts the band and palette properties of a raster feature and exposes them as attributes.
RasterBandRemover	Removes the selected bands of a raster.

RasterBandSeparator	Separates the bands and palettes from each input raster feature into one or more output raster features based on the number of input bands and palettes. 
RasterCellCoercer	Decomposes all input numeric raster features into individual points or polygons. 
RasterCellOriginSetter	Sets the raster's cell origin.
RasterCellValueCalculator	Performs an arithmetical operation on a pair of rasters. The first selected band of raster A is combined with the first selected band of raster B, the second selected band of raster A is combined with the second selected band of raster B, and so on. 
RasterCellValueReplacer	Replaces a range of values in the source raster with a new single value.
NEW **RasterCellValueRounder**	Rounds off raster cell values.
RasterCheckpointer	Sets a checkpoint in the raster processing, which forces previous processing to occur immediately and saves the current state to disk when complete.
RasterConsumer	Requests all the tiles from the raster geometry.
RasterExpressionEvaluator	Evaluates expressions on each cell in a raster, such as algebraic operations or conditional statements.
RasterExtentsCoercer	Replaces the geometry of input raster features with a polygon that covers the extents of the raster.
RasterExtractor	Serializes the geometry of the feature into the Raster Blob Attribute based on the selected writer format.
RasterGCPExtractor	Extracts the coordinate system and the Ground Control Points (GCPs) from the raster feature and exposes them as attributes.
RasterGCPSetter	Sets the GCP on a raster with the specified Column (pixel), Row (line), x Coordinate, y Coordinate and z Coordinate.

RasterGeoreferencer	Georeferences a raster with the specified parameters. 
RasterInterpretationCoercer	Alters the underlying interpretation of the bands of the raster geometry on the input features, using the specified conversion options. 
RasterMosaicker	Mosaics multiple raster features into a single raster feature. 
RasterNumericCreator	Creates a feature with a raster of the specified size with a numeric value and sends it into the workspace for processing. This transformer is useful for creating a very large image with a user-specified width and height.
RasterPaletteAdder	Creates a palette from an attribute and adds this palette to all selected bands on a raster.
RasterPaletteExtractor	Creates a string representation of an existing palette and saves it to an attribute.
RasterPaletteGenerator	Generates a palette out of the selected bands of a raster.
RasterPaletteInterpretation Coercer	Alters the underlying interpretation of the palettes of the raster geometry on the input features, using the specified conversion options.
RasterPaletteNodataSetter	Identifies the nodata value on a raster feature at the palette level.
RasterPaletteRemover	Removes the selected palettes of a raster.
RasterPaletteResolver	Resolves the palettes of the selected bands of the input raster features by using the band cell values to look up the corresponding palette values, which then replace the original band cell values in the raster.

RasterPropertiesExtractor	Extracts the geometry properties of a raster feature and exposes them as attributes.
RasterPyramider	Creates a series of pyramid levels for each input raster feature by specifying either the smallest pyramid level size or the number of pyramid levels to generate.
RasterReader	Reads and outputs raster features from the specified format and dataset.
RasterReplacer	Replaces the feature's geometry with the geometry held in the Raster Blob Attribute. The blob is decoded according to the selected raster format.
RasterResampler	Resamples an input raster using the desired dimensions, the desired cell size in ground units, or a percentage of the size. 
RasterRGBCreator	Creates a feature with a raster of the specified size with an RGB value and sends it into the workspace for processing.
RasterRotationApplier	Applies the raster rotation angle on the input raster properties to the rest of the raster properties and data values.
RasterSelector	Selects specific bands and palettes of a raster for subsequent transformer operations.
RasterSingularCellValue Calculator	Performs an arithmetic operation on two operands: the cell values of a raster and a numeric value.
RasterSubsetter	Reduces a raster to a subset of its original size. This is essentially a clipping operation using pixel bounds instead of ground coordinates.
RasterTiler	Splits each input raster into a series of tiles by specifying either a tile size or a number of tiles. 
VectorOnRasterOverlayer	Overlays vector features onto a single raster feature by drawing them onto the resulting output raster. The properties of the output raster are identical to that of the input raster.
WebMapTiler	Creates a series of image tiles that can be used by Microsoft Virtual Earth®. This is done by resampling rasters to various different resolutions and then splitting them into tiles.

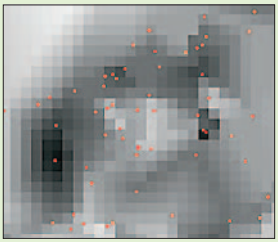
STRINGS – These transformers operate on character strings held in FME attributes. Transformers are provided for searching, replacing, changing case, and extracting character encodings from strings held in FME attributes.	
AttributeClassifier	Tests if the contents of the source attribute are entirely of a particular character classification, and routes the feature accordingly.
AttributeSplitter	Splits a selected attribute into a list attribute. Each item in the list contains a single token split from the list. For example, you can use this transformer to separate an attribute that has a comma-separated value list into its component pieces
AttributeTrimmer	Removes leading and trailing trim characters from the selected attributes.
AttributeValueMapper	Looks up and assigns attribute values based on other attributes, and stores the looked-up value in a new attribute.
CaseChanger	Changes the case of text attributes to UPPERCASE, lowercase, Title case, or Full Title Case.
CharacterCodeExtractor	Extracts the integral character code of the first character in the source string attribute, and adds its integer value in the character set to the feature as another attribute. This can be used to obtain the ASCII code of any character, including non-printable ones.
CharacterEncoder	Sets the result attribute to the character whose numeric code was contained in the source code attribute (or the entered integer).
GOIDGenerator	Calculates a Geographic Object IDentifier (GOID) for each incoming feature, and adds it as a new attribute. The GOID is a unique 128-bit number that incorporates the position of a feature with other numbers. The result is a unique value that may be used to distinguish features from each other.
HexDecoder	Decodes the given attribute into a new attribute, by converting its encoded hexadecimal value into a new ASCII string. Each hexadecimal couplet in the input attribute is converted into a single byte in the output string.
HexEncoder	Encodes the given attribute into a new attribute, by converting its ASCII value into a hexadecimal string.
NullAttributeReplacer	Checks all the selected attributes and sets them to the value given in the Default Value parameter if they were null (if they had no value).

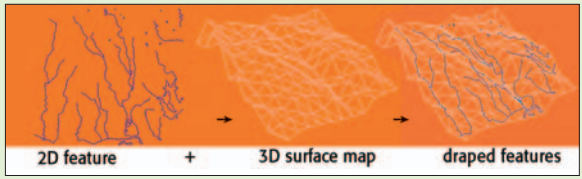
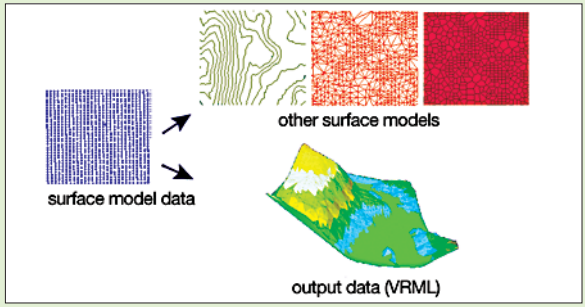
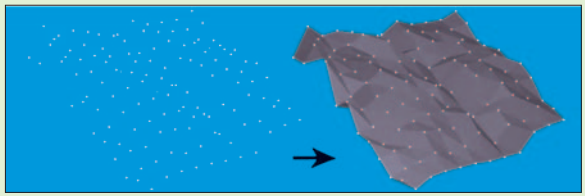
StringConcatenator	Concatenates the values of any number of attributes and constants, and stores the result in a new attribute. It complements Workbench's fanout capability by allowing you to fan out by more than one attribute simultaneously.
	`7445 132nd Street`
	`Surrey` → `7445 132nd Street, Surrey, BC`
	`BC`

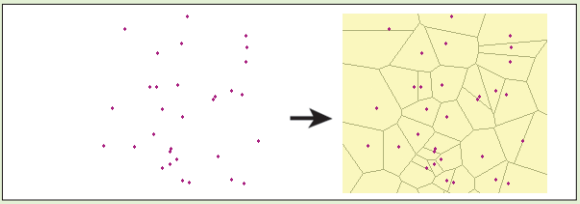
StringFormatter	Reformats the data held in each specified attribute according to the Tcl *format* command, which is similar to the C *printf* function. Attribute values can be formatted into strings, characters, or numbers.
StringLengthCalculator	Calculates the length of the string in Source Attribute.
StringPadder	Pads the given attributes with spaces, either on the right or left side.
StringPairReplacer	Replaces characters in the value contained in the source attribute based on the replacement key-value pairs.
StringReplacer	Replaces substrings matching a string or regular expression in the string contained in the source attribute.
StringSearcher	Performs a regular expression match on the value of the specified attribute.

SubstringExtractor	Extracts a substring from the source attribute.
TimeStamper	Adds a time stamp to a feature as a new attribute. The format of the time stamp is set as a parameter of the transformer.
UUIDGenerator	Calculates a Universally Unique IDentifier (UUID) for each incoming feature, and adds it as a new attribute. An example UUID looks like: 7672aac8-fa0b-464c-b0b8-3efa9ae9cd86

STYLERS – These transformers are used to prepare features for output to particular formats by providing a convenient interface for setting color and other display characteristics.	
NEW **DGNStyler**	Prepares features for output to Bentley Microstation Design V7/V8 by providing a convenient interface to set a variety of format-specific attributes.
NEW **DWGStyler**	Prepares features for output to AutoCAD DWG/DXF by providing a convenient interface to set a variety of format-specific attributes.
NEW **MapInfoStyler**	Prepares features for output to Mapinfo MIF/MID or MapInfo TAB by providing a convenient interface to set a variety of format-specific attributes.

SURFACES – These transformers create, use, and output surfaces. They operate on data that defines a surface through the z coordinate, and then either outputs this surface in a variety of ways or applies the surface to other data.	
AppearanceAdder	Adds appearance style(s) to the front, back or both sides of surfaces. You can also set the texture coordinates of the surfaces.
AppearanceExtractor	Extracts appearance style(s) from the front and/or back side of the surfaces.
AppearanceRemover	Removes appearances from the front and/or back side of surface features. The surface will then inherit its appearance from its parent, if a parent surface exists.
AppearanceStyler	Creates an appearance style that's applied to a surface later, using the Appearance Adder transformer, for example.
ContourGenerator	Generates contours from the underlying surface, which is defined by the input POINTS, 3D_LINES, and BREAKLINES.
DEMGenerator	Generates a Digital Elevation Model (DEM) as a regularly spaced set of output DEM points from the input POINTS, BREAKLINES, and 3D_LINES.
RasterDEMGenerator	Generates a DEM represented as Raster from the input POINTS, BREAKLINES, and 3D_LINES. 
SectorGenerator	Inputs point features with the following data contained in the chosen attributes: site name, sector name (optional), azimuth, and radius. Within points having the same site name, points have their geometry changed to polygons approximating sector shapes.

NEW **SurfaceSplitter**	Splits a double-sided input surface geometry into two single-sided surfaces – one equal to the front side of the input surface and one equal to the back side of the input surface.
SurfaceDraper	Interpolates z coordinates and adds them to DRAPED_FEATURES based on the underlying surface defined by the POINTS, 3D_LINES, and BREAKLINES. 
SurfaceModeller	Builds and queries surfaces, drapes features, or changes the surface representation between Digital Elevation Model (DEM), Triangulated Irregular Network (TIN), and CONTOURS. You can go from any representation to any other representation. This transformer is useful when you need to get several things from a defined surface. 
TINGenerator	Generates a Delaunay surface model and outputs the defining Triangulated Irregular Network (TIN). The TIN is output as both Triangles and TIN Edges. 
VoronoiCellGenerator	Generates a Voronoi diagram that represents the closest points around point locations. The diagram is such that the Voronoi cells only radiate out from the cell at a distance specified by the radius parameter.

VoronoiDiagrammer	Generates a Voronoi diagram or Thiessen polygon from the input POINTS. A Voronoi diagram is a set of polygons that represent proximity information about a set of input points. Each polygon in the diagram defines the area of space that is closest to a particular input point. 

WEB SERVICES – These transformers access web services using the HTTP protocol.	
Decelerator	Slows down the flow of features through the workspace.
DirectTweeter	Sends a direct Twitter™ message from Workbench.
GeoRSSFeatureExtractor	Constructs GeoRSS documents from the input features and stores them in the specified attribute for the features that are output by the GeoRSS port.
GeoRSSFeatureReplacer	Constructs features out of GeoRSS documents and URLs that are stored in a specified attribute of the input features. The features from the GeoRSS document and URL can be output with the attributes from the original feature and merged.
HTTPDeleter	Deletes a target by performing an HTTP DELETE operation on the specified URL, storing the results in the specified target attribute.
HTTPFetcher	Fetches a target by performing an HTTP GET operation on the specified URL, storing the results in the specified target attribute.
HTTPFileUploader	Uploads a file's contents by performing a HTTP PUT or HTTP POST operation on the specified URL, storing the results in the specified target attribute.
HTTPUploader	Uploads a message by performing an HTTP PUT or HTTP POST operation on the specified URL, storing the results in the specified target attribute.
ImageFetcher	Fetches an image by performing an HTTP GET operation on the specified URL, and then returning the resulting data as the geometry of a raster feature.
JSONExploder	Extracts portions of JSON (JavaScript Object Notation) formatted text into new FME features.
JSONExtractor	Extracts portions of JSON formatted text into feature attributes.
ProxixGeocoder	Geocodes addresses using a Proxix Geospatial Enterprise Real-Time (GSERT) server.
Tweeter	Sends a Twitter status update from Workbench.
TweetSearcher	Runs a search for Twitter entries that contain the given query.
TwitterStatusFetcher	Retrieves the Twitter status updates for a particular user.
URLDecoder	Decodes a string from its URL-encoded form and stores the result in an attribute.

URLEncoder	Converts a string value to its URL-encoded form and stores the result in an attribute.
WebCharter	Creates a URL that can be used to obtain a chart of the specified data as a PNG image from the Google™ Chart API. One URL is created for each feature that enters the transformer. Use of the Google Chart API is subject to the Terms of Service for the API.
WhiteStarLeaseBuilder	Posts a query to a WhiteStar Legal2Map™ WebServices (W3) server to obtain points or polygons that match a list of legal land descriptions.

WORKFLOW – These transformers run workspaces either locally or on an FME Server. To use these transformers, you may need access to an FME Server.	
FMEServerJobSubmitter	Submits FME Spatial ETL jobs to be run on an FME Server. A job consists of a workspace (housed within a repository on an FME Server) together with values for each of its published parameters.
FMEServerJobWaiter	Waits until submitted FME Spatial ETL jobs are completely processed by an FME Server. The list of jobs to wait for is identified by the job IDs of the input features. When a job that the transformer is waiting for is completed, it outputs the corresponding feature immediately.
FMEServerWorkspaceRunner	Submits FME Spatial ETL jobs to be run on an FME Server, and downloads the resulting data to a specified location. You can optionally upload files used for the job, and download results locally when the FME Server job is complete.
WorkspaceRunner	Runs another FME Workbench workspace on the local computer by spawning a new FME process. This transformer is useful for batch processing, especially in conjunction with the Directory and File Reader.

XML – These transformers work with XML data by mapping XML elements into FME features, using stylesheets to convert XML documents, and querying collections of XML data.	
XMLFeatureMapper	Constructs features from XML documents via xfMaps.
NEW NEW **XMLFormatter**	Provides various options for formatting and cleaning up XML documents.
NEW **XMLFragmenter**	Maps elements from an XML document into XML fragments.
NEW **XMLNamespaceDeclarer**	Declares missing namespaces in XML documents by matching prefixes from another sample XML file whose namespaces are fully declared.
XMLTemplater	Populates an XML template with feature attribute values.
NEW **XMLValidator**	Validates the syntax or schema of an XML file or text.
XQueryExploder	Extracts portions of XML text using XQuery expressions into new FME features.
XQueryExtractor	Uses XQuery expressions to extract portions of XML text into feature attributes.
XQueryUpdater	Provides updates to an XML document using XQuery Update expressions.
XSLTProcessor	Uses an eXtensible Stylesheet Language (XSL) stylesheet to convert an XML document. Common output formats include text, RSS, SVG, and CSV.

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Suite 2017, 7445 - 132 Street, Surrey, BC, Canada V3W 1J8
Tel: 604-501-9985 E-mail: sales@safe.com

www.fmeusercentral.com

Printed in Canada

1102-01-A