

Danish UFO Reader/Writer

FORMAT NOTES:

This format is not supported by FME Base Edition.

The Danish UFO Reader and Writer modules provide the Feature Manipulation Engine (FME) with the ability to read and write UFO files. The UFO is a published ASCII format used by the National Survey and Cadastre of Denmark.

Overview

The UFO format provides facilities for carrying a variety of metadata together with the actual feature data. Metadata can even be associated with individual vertices. The UFO reader and writer provide a complete set of facilities for reading and writing this metadata, however, custom mapping files are required to populate the metadata fields. A custom mapping file is provided with FME to do UFO to UFO translation in a lossless manner.

UFO files store both feature geometry and attribution. UFO feature coordinates are always measured in centimetres.

A UFO file has the following file name extension:

File Name Extension	Contents
.ufo	UFO format file

The extension is added to the basename of the UFO file.

UFO Quick Facts

Format Type Identifier	UFO
Reader/Writer	Both
Licensing Level	Professional
Dependencies	None
Dataset Type	File
Feature Type	Feature role
Typical File Extensions	.ufo
Automated Translation Support	Yes
User-Defined Attributes	No
Coordinate System Support	No
Generic Color Support	No
Spatial Index	Never
Schema Required	Yes
Transaction Support	No
Geometry Type	ufo_type

Geometry Support			
Geometry	Supported?	Geometry	Supported?
aggregate	yes	point	yes
circles	no	polygon	yes
circular arc	no	raster	no
donut polygon	yes	solid	no
elliptical arc	no	surface	no
ellipses	no	text	yes
line	yes	z values	yes
none	yes		

Reader Overview

The UFO reader module produces FME features for all data held in the UFO file. The UFO reader extracts data from the file one row at a time, producing FME features from the file before passing them on to the rest of the FME for further processing. The features are produced in the order they are read from the source file. The various metadata features are emitted first, followed by the feature data. When the file is exhausted, the UFO reader terminates.

Reader Directives

The directives listed below are processed by the UFO reader. The suffixes listed are prefixed by the current <ReaderKeyword> in a mapping file. By default, the <ReaderKeyword> for the UFO reader is UFO.

DATASET

Required/Optional: *Required*

This is the name of an UFO file. The extension for UFO files is `.ufo`.

An example of the `DATASET` keyword in use is:

```
UFO_DATASET /usr/data/ufo/input.ufo
```

[Workbench Parameter: <WorkbenchParameter>](#)

Writer Overview

The UFO writer creates and writes feature data to an UFO file specified by the `DATASET` keyword. Existing UFO files with the same name as the specified file are overwritten with the new feature data.

The UFO writer process two kinds of features: regular features and meta features. Regular features are features such as points and lines, which are commonly seen in FME formats. Meta features are features specifically created for the UFO writer so that it can receive information for the origin for objects, label types, and the other metadata types stored in Sections 2 through 5 of the format. Note that the order in which the meta features are received by the writer is extremely important. They must be received in the order listed in the meta feature section under the Feature Representation section.

Writer Directives

The directives listed below are processed by the UFO writer. The suffixes shown are prefixed by the current `<WriterKeyword>` in a mapping file. By default, the `<WriterKeyword>` for the UFO writer is `UFO`.

DATASET

Required/Optional: *Required*

The `DATASET` directive operates in the same manner as it does for the UFO reader.

[Workbench Parameter: <WorkbenchParameter>](#)

DEF

Required/Optional: *Required*

SOURCE_IS_UFO

The translation from UFO to UFO is handled differently from a translation from other formats to UFO. In such a situation, the input data stream will contain additional information that the UFO writer can make use of. This directive is used in the "ufo2ufo.fme" mapping file provided in the FME installation directory.

An example of the `SOURCE_IS_UFO` keyword in use is:

```
UFO_SOURCE_IS_UFO yes
```

[Workbench Parameter: <WorkbenchParameter>](#)

Feature Representation

In addition to the generic FME feature attributes that FME Workbench adds to all features (see *About Feature Attributes* on page 7), this format adds the format-specific attributes described in this section.

Each UFO element, regardless of its geometry type, shares a number of other parameters, as described in the following tables. Subsequent sections will describe parameters specific to each of the supported element types.

Attribute Name	Contents
ufo_type	The UFO geometric type of this entity. Range: ufo_point ufo_line ufo_polygon ufo_text ufo_none Default: No default

The main object line:

Attribute Name	Contents
ufo_obj_code	An integer that expresses the object code. Range: 32 bit integer Default: 0
ufo_db	A text string which indicates the database the object connects to. Range: Max. of 80 characters Default: no_database
ufo_key	The unique identification in a database expressed as a whole number. 0 means that the object has not yet been assigned a unique identification. Range: 32 bit integer Default: 0
ufo_obj_origin	A whole number which indicates the object origin group the object belongs to. Range: 32 bit integer Default: 0

The object reference:

Attribute Name	Contents
ufo_obj_ref{<number>}.group	A integer that refers to an object reference type group. Note: The <number> refers to the group of object reference description. Directives that belong to the same description should have the same number. The <number> is a positive integer which starts at 0. Range: 32 bit integer Default: 0

Attribute Name	Contents
ufo_obj_ref {<number>}.database	A text string which indicates the database being referred to. Range: max. 80 characters Default: no_database
ufo_obj_ref {<number>}.key	A whole number which indicates the object being referred to in the database. Range: 32 bit integer Default: 0

The label description:

Attribute Name	Contents
ufo_label {<number>}.ufo_text_type	A character which indicates if the label has a full description or a free text description. The allowable range for the string are "L" and "F" corresponding to "full" and "free". Note: The <number> refers to the group of label description. Directives that belong to the same description should have the same number. The <number> is a positive integer which starts at 0. Range: L or F Default: No default.
ufo_label{<number>}.ufo_group	A whole number that refers to a label type group. Range: 32 bit integer Default: 0
ufo_label{<number>}.ufo_northing	An integer which refers to the north-coordinate of the label. Range: 32 bit integer Default: 0
ufo_label{<number>}.ufo_easting	An integer which refers to the east-coordinate of the label. Range: 32 bit integer Default: 0
ufo_label{<number>}.ufo_height	An integer which refers to the height-component of the label. Range: 32 bit integer Default: 0
ufo_label{<number>}.ufo_pt_origins	A whole number which refers to the point origin group. Range: 32 bit integer Default: 0
ufo_label{<number>}.ufo_font	A whole number code for the label text font. Range: 32 bit integer Default: 0

Attribute Name	Contents
ufo_label{<number>} .ufo_caps	Specifies the capitalization of the label text. Range: 32 bit integer 0 = Mix of lower and uppercase letters, 1 = Only upper case letters. Default: 0
ufo_label{<number>} .ufo_text_size	A whole number that refers to the height of the label. Range: 32 bit integer Default: 0
ufo_label{<number>} .ufo_color	A whole number that refers to the color of the label. Range: 32 bit integer Default: 0
ufo_label{<number>} .ufo_justification	A code in DSFL that indicates the justification of the label. Range: 32 bit integer 1 = TL (top left), 2 = TM (top middle), 3 = TR (top right), 4 = ML (middle left), 5 = MM (middle middle), 6 = MR (middle right), 7 = BL (bottom left), 8 = BM (bottom middle), 9 = BR (bottom right) Default: 3
ufo_label{<number>} .ufo_orientation	A whole number that indicates the orientation of the label. Range: 32 bit integer Default: 0
ufo_label{<number>} .ufo_spacing	A whole number that indicates how many extra spaces are needed. Range: 32 bit integer Default: 0
ufo_label{<number>} .ufo_number	A whole number used in SNOSOR to indicate the place/appellative-flag. Range: 32 bit integer 0 = appellative, 1 = place name. Default: 0
ufo_label{<number>} .ufo_text_string	The actual text for the label. Range: max. 80 characters Default: no_label

Points

ufo_type: ufo_point

A multi-point feature in UFO turns into an aggregate point feature in FME. In this case, the feature will have the attribute `ufo_pt_origins{<number>}` for each point with the `<number>` being the coordinate index. The value of this attribute is a comma separated value list of point origin numbers with the first one being the point origin number for

the first point and the second one being the point origin number of the second point and so on.

Attribute Name	Contents
ufo_pt_origins	This is the point origin number that refers to the meta feature ufo_meta_pt_origin's ufo_group_id. Range: 32 bit integer Default: 0

Lines

ufo_type: ufo_line

A multi-line feature in UFO turns into an aggregate line feature in FME. In this case, the feature will have the attribute `ufo_pt_origins{<number>}` for each line with the `<number>` being the line index. The value of this attribute is a comma separated value of point origin numbers with the first one being the point origin number for the first point and the second one being the point origin number of the second point. Each comma-separated value list contains the point origin number for a line.

Attribute Name	Contents
ufo_pt_origins	This a comma-separated string that holds the list of <code>ufo_pt_origins</code> as describe for Points. If all points of the line has the same value for <code>ufo_pt_origins</code> then this string will only hold that single value. Range: 32 bit integer Default: 0

Polygons

ufo_type: ufo_polygon

A polygon with more than one line is turned into a donut feature in FME. There is also a multi-polygon feature in UFO that turns into an aggregate polygon feature. In both cases, the feature will have the attribute `ufo_pt_origins{<number>}` for each polygon, with `<number>` being the polygon index. The order of the polygons are important.

The value of `ufo_pt_origins{<number>}` is a comma separated value of point origin numbers with the first one being the point origin number for the first point and the second one being the point origin number of the second point. Each comma-separated value list contains the point origin number for a polygon.

Attribute Name	Contents
ufo_pt_origins	This a comma-separated string that holds the list of <code>ufo_pt_origins</code> as describe for Points. If all points of the polygon has the same value for <code>ufo_pt_origins</code> then this string will only hold that single value. Range: 32 bit integer Default: 0

Text

ufo_type: ufo_text

Text features are not native UFO features. They are created during translation from UFO to other formats, in order to convert the text information which can be part of the UFO feature label descriptions (Here, we call this UFO feature the original feature). The text feature is a clone of the original feature with the addition of the attributes `ufo_text_size`, `ufo_text_string` and `ufo_orientation` which are listed in the label description part of the Feature Representation Section, but do not contain the `ufo_label{<number>}` prefix. The coordinates for the text features are the first set of coordinates of the original feature if it is not available through the original features label description.

Meta Features

ufo_type: ufo_none

Note that the meta features must be created in the mapping file in the exact order as it is listed here. Otherwise, an error is output. In addition, all regular feature must be created before the meta features in order for the UFO Writer to function correctly.

Header

This is the header feature of the ufo file. The feature type of this feature is `ufo_meta_header`. The supported attributes are as followed:

Attribute Name	Contents
<code>ufo_coord_sys</code>	A text string which describes the coordinate system used. Range: Text string with max length of 25 Default: <code>ufo_coord_sys</code>

Origin and Definitions of Precision for Points

This is the Origin and Definitions of Precision for Points feature of the ufo file. The feature type of this feature is `ufo_meta_pt_origin`. The supported attributes are as followed:

Attribute Name	Contents
<code>ufo_group_id</code>	A unique number for KMS which expresses a number for the group. Range: Integer Default: 0
<code>ufo_attr{<number>}</code>	Text strings which contain information belonging to the group. <number> is a positive integer which expresses the line number of the text strings. Range: Consecutive integer starting from 0 Default: 0

Label Types

This is the Label Types feature of the ufo file. The feature type of this feature is `ufo_meta_label`. The supported attributes are exactly the same as those supported by Origin And Definitions of Precision For Points feature.

Object Reference Types

This is the Object Reference Types feature of the ufo file. The feature type of this feature is `ufo_meta_ref`. The supported attributes are exactly the same as those supported by Origin And Definitions of Precision For Points feature.

Origin for objects

This is the Origin for Object feature of the ufo file. The feature type of this feature is `ufo_meta_obj_origin`. The supported attributes are exactly the same as those supported by Origin And Definitions of Precision For Points feature.

