

# Northgate StruMap Reader/Writer

The StruMap<sup>®</sup> reader and writer modules allow the Feature Manipulation Engine (FME) to read and write Northgate Information Solutions StruMap SGF/SGX format files. StruMap SGF/SGX files use a published ASCII format.

## Overview

The StruMap reader and writer support the storage of point, line, polygon, and text geometric data in a single file. The name of the user-defined attributes are not stored with each feature, but rather in the file header where the structure of the features are defined.

The FME considers a StruMap dataset to be a single file.

## StruMap Quick Facts

Format Type Identifier	STRUMAP
Reader/Writer	Both
Licensing Level	Base
Dependencies	None
Dataset Type	File for Both
Feature Type	Feature Code
Typical File Extensions	.sgf
Automated Translation Support	Yes
User-Defined Attributes	Yes
Coordinate System Support	No
Generic Color Support	Yes
Spatial Index	Never
Schema Required	Yes
Transaction Support	No
Geometry Type	strumap_type

Geometry Support			
Geometry	Supported?	Geometry	Supported?
aggregate	no	point	yes
circles	no	polygon	yes
circular arc	no	raster	yes
donut polygon	no	solid	no
elliptical arc	no	surface	no
ellipses	no	text	yes

Geometry Support			
Geometry	Supported?	Geometry	Supported?
line	yes	z values	no
none	no		

## Reader Overview

The StruMap reader extracts features from the input file individually and passes them on to the rest of the FME for further processing. An important thing to note in the StruMap input file is that text components or child can be attached to any geometric types, whereas line symbols can only be attached to lines. The StruMap reader will not output these as a single aggregate feature, but rather as a series of features sharing the same `strumap_id`. Additionally, a `strumap_child_id` is added to each child to identify its order.

## Reader Directives

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

### DATASET

**Required/Optional:** *Required*

The value for this keyword is the directory containing the StruMap files to be read. A typical mapping file fragment specifying an input StruMap dataset looks like:

**Example:**

```
STRUMAP_DATASET /usr/data/strumap/input.sgf
```

**Workbench Parameter:** [<WorkbenchParameter>](#)

### DEF

**Required/Optional:** *Required*

Each StruMap feature must be defined before it can be read. The definition specifies the feature code of the feature, and the names and the types of all attributes. The syntax of a StruMap `DEF` line is:

```
<ReaderKeyword>_DEF <baseName> \
  [<attrName> <attrType>]+
```

The following table shows the attribute types supported.

Field Type	Description
<code>char(&lt;width&gt;)</code>	Character fields store fixed length strings. The <code>width</code> parameter controls the maximum number of characters that can be stored by the field. No padding is required for strings shorter than this width.

Field Type	Description
date	Date fields store date as character strings with the format YYYYMMDD.
double	Float fields store 64-bit floating point values. There is no ability to specify the precision and width of the field.
integer	Integer fields store 32-bit signed integers.
logical	Logical fields store TRUE/FALSE data. Data read or written from and to such fields must always have a value of either <code>true</code> or <code>false</code> .

**Workbench Parameter:** [<WorkbenchParameter>](#)

## IDs

**Required/Optional:** *Optional*

This optional specification limits StruMap features read. If no IDs are specified, then all StruMap features in the input file are read.

The syntax of the IDs keyword is:

```
<ReaderKeyword>_IDs<baseName>           \
                <baseName1>              \
                <baseNameN>
```

The basenames must match those used in DEF lines.

The example below selects only the `roads` StruMap feature for input during a translation:

```
STRUMAP_IDS roads
```

**Workbench Parameter:** [<WorkbenchParameter>](#)

## Writer Overview

The StruMap writer creates and writes all features it is given to the file specified by the DATASET keyword. Any existing StruMap files in the directory that have the same name as the output dataset will be overwritten with the new data.

Multiple geometric types can be stored in each StruMap file. Any feature whose feature type is not specified in the mapping file will not be written to file. If no `strumap_id` is present, a feature is considered a single feature in the output file (as opposed to being part of an aggregate). Features that share the same ID, which pass through the writer one after another (grouped by ID prior to entering the writer), are considered a single feature when output to file. It is an additional requirement that components of a parent feature pass through the writer before their children. Otherwise, the parent and child relationship may not be realized.

The JOIN directive is not supported by the writer; in other words, features must carry all the attribution before they are passed into the writer.

## Writer Directives

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

### DATASET

**Required/Optional:** *Required*

This specifies the output file to which the features are to be written. A typical mapping file fragment specifying an output StruMap dataset looks like:

```
STRUMAP_DATASET /usr/data/Strumap/output.sgf
```

**Workbench Parameter:** [<WorkbenchParameter>](#)

### DEF

**Required/Optional:** *Required*

Each StruMap feature type must be defined before it can be written. The definition specifies a unique feature code of the feature type, and the names and the types of all attributes. The syntax of a StruMap `DEF` line is:

```
<WriterKeyword>_DEF <baseName> \
  [<attrName> <attrType>]+
```

The attribute types supported are the same as those listed in the Reader section.

**Workbench Parameter:** [<WorkbenchParameter>](#)

### STRING\_DELIMITER

**Required/Optional:** *Optional*

This is the character added around a string to indicate the beginning and the ending of a string. For example, to indicate that a pair character `"\"` should be used to quote a string, the following is used. By default, double quotation marks are used.

```
<WriterKeyword>_STRING_DELIMITER "\"
```

## Feature Representation

StruMap features consist of geometry and attributes. In particular, all StruMap features contain a `strumap_type` attribute, which identifies the geometric type of the feature.

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.

There are five types defined for the `strumap_type` attribute:

Attribute Name	Contents
<code>strumap_type</code>	<p>The StruMap geometric type of this feature.</p> <p><b>Range:</b></p> <pre>strumap_point   strumap_line   strumap_line_symbol   strumap_polygon   strumap_displayed_text</pre> <p><b>Default:</b> No default</p> <p><b>Note:</b> <code>strumap_displayed_text</code> is used in place of both <code>strumap_text</code> and <code>strumap_ftext</code> because the latter types are not supported by the Writer.</p>
<code>strumap_id</code>	<p>The StruMap ID number of the feature. This helps the writer identify the components of each feature. Features which are meant to be part of an aggregate feature should share the same ID. For the reader, this is automatically set. For the writer, if this is not specified, then each feature is considered a single output feature when written to file.</p> <p><b>Range:</b> Positive Integers</p> <p><b>Default:</b> No default</p>
<code>strumap_child_id</code>	<p>The StruMap geometric type of this feature.</p> <p><b>Range:</b></p> <pre>strumap_point   strumap_line   strumap_line_symbol   strumap_polygon   strumap_displayed_text</pre> <p><b>Default:</b> No default</p>

## Points

**strumap\_type:** `strumap_point`

This indicates that the feature is a StruMap point. Additional attributes include:

Attribute Name	Contents
<code>strumap_angle</code>	<p>The angle of the rotated symbol.</p> <p><b>Range:</b> Real</p> <p><b>Default:</b> No default</p>

**strumap\_type:** `strumap_line_symbol`

This indicates that the feature is a StruMap line symbol. This is actually a component of a StruMap line and cannot be written to file on its own in terms for the StruMap definition; however, it can be represented by other formats.

In addition to the geometry are the following attributes:

Attribute Name	Contents
strumap_linesym_num	The line symbol number of the line symbol. <b>Range:</b> integers greater or equal to 0 <b>Default:</b> No default
strumap_linesym_angle	The rotation of the line symbol. <b>Range:</b> Any real number <b>Default:</b> No default
strumap_linesym_scale	The scale factor of the line symbol. <b>Range:</b> Any real number <b>Default:</b> No default

## Lines

**strumap\_type:** strumap\_line

This indicates that the feature is a StruMap line. In addition to the geometry are the following attributes:

Attribute Name	Contents
strumap_line_style	The line style of the line. <b>Range:</b> integers greater than or equal to 0 <b>Default:</b> No default
strumap_red	The red intensity of the line. This must be used together with <code>strumap_green</code> and <code>strumap_blue</code> in order to be used by the StruMap writer. <b>Range:</b> Integer from 1 to 255 <b>Default:</b> No default
strumap_green	The green intensity of the line. <b>Range:</b> Integer from 1 to 255 <b>Default:</b> No default
strumap_blue	The blue intensity of the line. <b>Range:</b> Integer from 1 to 255 <b>Default:</b> No default
strumap_flowdir_position	The position of the flow direction. This must be used with the <code>strumap_flowdir_easting</code> and <code>strumap_flowdir_northing</code> attributes. <b>Range:</b> 0 (not set)   1 (flow from point 1 to point 2)   2 (flow from point 2 to point 1)   3 (bidirectional flow)   4 (blocked) <b>Default:</b> No default
strumap_flowdir_x1	The x coordinate for the first point in terms for the <code>strumap_flowdir_position</code> . <b>Range:</b> Real <b>Default:</b> No default

Attribute Name	Contents
strumap_flowdir_y1	The y coordinate for the first point in terms for the strumap_flowdir_position. <b>Range:</b> Real <b>Default:</b> No default
strumap_flowdir_x2	The x coordinate for the second point in terms for the strumap_flowdir_position. <b>Range:</b> Real <b>Default:</b> No default
strumap_flowdir_y2	The y coordinate for the second point in terms for the strumap_flowdir_position. <b>Range:</b> Real <b>Default:</b> No default
strumap_mask_x{<number>}	The x coordinate of the mask. <number> is a positive integer used to indicate the order of the mask. Hence, in order for this to be valid, it has to be used in conjunction with strumap_mask_y{<number>} and strumap_mask_length{<number>} where <number> share the exact same value. As a result of this mechanism, more than one mask can be added to each line feature. <b>Range:</b> Real <b>Default:</b> No default
strumap_mask_y{<number>}	The y coordinate of the mask. See strumap_mask_x. <b>Range:</b> Real <b>Default:</b> No default
strumap_mask_length{<number>}	The length of the gap. See strumap_mask_x. <b>Range:</b> Real <b>Default:</b> No default

## Polygons

**strumap\_type:** strumap\_polygon

This indicates that the feature is a StruMap polygon. The first and last coordinates of the polygon must be the same.

Additional attributes includes:

Attribute Name	Contents
strumap_seed_x	The x coordinate of the seed for the polygon. <b>Range:</b> Real <b>Default:</b> Defaults to the x coordinate of the first point in the polygon.
strumap_seed_y	The y coordinate of the seed for the polygon. <b>Range:</b> Real <b>Default:</b> Defaults to the y coordinate of the first point in the polygon.

## Text

**strumap\_type:** strumap\_displayed\_text

StruMap displayed text features are used to specify annotation information. Each text feature has a location defined by a single point geometry, and can have its text string, style, justification, and rotation angle set independently.

The following table lists the special FME attribute names used to control the `strumap_display_text` settings.

Attribute Name	Contents
<code>strumap_attr_code</code>	A text string representing the attribute short code as defined in the rule file.
<code>strumap_attr_value</code>	A text string containing the value of the attribute.
<code>strumap_height</code>	The height of the <code>strumap_attr_value</code> in ground units.
<code>strumap_width</code>	The width of the <code>strumap_attr_value</code> in ground units.
<code>strumap_red</code>	The red color component of the <code>strumap_attr_value</code> .
<code>strumap_green</code>	The green color component of the <code>strumap_attr_value</code> .
<code>strumap_blue</code>	The blue color component of the <code>strumap_attr_value</code> .
<code>strumap_angle</code>	The angle at which the <code>strumap_attr_value</code> is displayed.
<code>strumap_position</code>	The justification of the <code>strumap_attr_value</code> . <b>Range:</b> 0 (not set)   1 (bottom left)   2 (center left)   3 (top left)   4 (bottom center)   5 (center center/ original position)   6 (top center)   7 (bottom right)   8 (center right)   9 (top right) <b>Default:</b> No default
<code>strumap_min_length</code>	The minimum length of span on which to display bubble. <b>Range:</b> Positive real numbers <b>Default:</b> 0
<code>strumap_circle</code>	Indicates that an ellipse should be drawn instead of a box. <b>Range:</b> true <b>Default:</b> false
<code>strumap_box</code>	Indicates that a box should be drawn. <b>Range:</b> true <b>Default:</b> false
<code>strumap_solid</code>	Indicates that the ellipse or the box should be filled. <b>Range:</b> true <b>Default:</b> false

<b>Attribute Name</b>	<b>Contents</b>
strumap_line	Indicates that a line should be drawn from the text to the item. <b>Range:</b> true <b>Default:</b> false
strumap_arrow	Indicates that an arrow should be drawn on a line. <b>Range:</b> true <b>Default:</b> false

