

Autodesk MapGuide (Version 6.5 and older) SDF Reader/Writer

FORMAT NOTES:

- This format is not supported by FME Base Edition.
- To use the SDF Reader/Writer, you must install either Autodesk® MapGuide® or the “SDF Toolkit to support Publish to MapGuide.” The toolkit is free and available from:
<http://usa.autodesk.com/adsk/servlet/ps/dl/item?siteID=123112&id=4307350&linkID=2475897>

The Autodesk MapGuide SDF Reader/Writer allows the Feature Manipulation Engine (FME) to read and write Autodesk MapGuide binary SDF files. The SDF file format is a binary format used with AutoDesk’s MapGuide and other World Wide Web map authoring tools.

Overview

SDF data is two-dimensional (2D), and SDF files store both geometry and attributions. A logical SDF dataset consists of one or more files in the same directory with the extension `.sdf` – this extension is added to the basename of the SDF files. Each `.sdf` file is also paired with its spatial index file, which has the extension `.sif`. The paired `.sif` file is required in order to use its `.sdf` file for processing. An optional key index file (`.kif`) may also be created and is used by MapGuide to speed access to the file.

The SDF reader and writer support the storage of *point*, *line*, and *polygon* geometric data in `.sdf` files. Output files contain only one geometry type to conform with MapGuide. The SDF format can also store features with no geometry (which are referred to as having a geometry of `none`).

SDF Quick Facts

Format Type Identifier	SDF
Reader/Writer	Both
Licensing Level	Professional
Dependencies	Autodesk MapGuide or SDF Toolkit
Dataset Type	Directory or File
Feature Type	File base name
Typical File Extensions	.sdf
Automated Translation Support	Yes
User-Defined Attributes	No
Coordinate System Support	Yes
Generic Color Support	No
Spatial Index	Always
Schema Required	Yes
Transaction Support	No
Geometry Type Attribute	sdf_type

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

Reader Overview

The SDF reader first scans the directory it is given for SDF files that have been defined in the mapping file. The SDF reader then extracts features from the files one at a time, and passes them on to the rest of the FME for further processing.

Reader Directives

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

DATASET

Required/Optional: *Required*

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

```
SDF_DATASET /usr/data/sdf/92i080
```

Workbench Parameter: [<WorkbenchParameter>](#)

IDs

Required/Optional: *Optional*

This specification is used to limit the available and defined SDF files read. The syntax of the `IDs` keyword is:

```
<ReaderKeyword>_IDs <baseName1> \
                      <baseName2> \
                      <baseNameN>
```

The basenames must match those used in `DEF` lines.

The example below selects only the `roads` SDF file for input during a translation:

```
SDF_IDS roads
```

Workbench Parameter: [<WorkbenchParameter>](#)

SPATIAL_EXTENT

Required/Optional: *Optional*

This specifies a bounding box where all features retrieved by the read function must in some part be within the bounding box area. This keyword is formatted as follows:

```
<ReaderKeyword>_SPATIAL_EXTENT <minX> <minY> <maxX> <maxY>
```

Workbench Parameter: [<WorkbenchParameter>](#)

Writer Overview

The SDF writer outputs each feature type into a separate file in order to comply with Autodesk MapGuide. Each feature has the following associations: vertices, a name, an ID, and a Universal Resource Locator (URL). The SDF Writer will output `.sdf` files along with their paired `.sif` file.

Note: The Autodesk MapGuide Component Toolkit (SDFComTk) limits the number of vertices that can be stored in each feature when writing. If the toolkit version is less than 6.0, the maximum number of vertices that can be written to a feature is 16384. If you have version 6.0 or newer, the limit is 65535 vertices.

Writer Directives

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

DATASET

Required/Optional: *Required*

The value for this keyword is the name of the created SDF directory. If a directory of this name exists, it is replaced by the new SDF. A typical mapping file fragment specifying an output SDF dataset looks like:

```
SDF_DATASET /tmp
```

Workbench Parameter: [<WorkbenchParameter>](#)

APPEND

Required/Optional: *Optional*

This directive indicates whether or not the writer will append to existing SDF files. An example use of this directive is:

```
SDF_APPEND yes
```

Workbench Parameter: [<WorkbenchParameter>](#)

DEF

Required/Optional: *Required*

The SDF writer uses SDF_DEF lines to define files to write features to. A typical mapping file fragment specifying an output SDF file looks like:

```
SDF_DEF roads
```

Workbench Parameter: [<WorkbenchParameter>](#)

CREATE_KIF

Required/Optional: *Optional*

This directive controls whether or not a key index file will be created along with the SDF file. If an existing SDF file is being appended to, this flag is ignored. An example specification is:

```
SDF_CREATE_KIF yes
```

Workbench Parameter: [<WorkbenchParameter>](#)

KEY_MAX_LEN

Required/Optional: *Optional*

This directive is only used when a key index file is being created, and sets the maximum length of the values expected to be found in the `sdf_key` attribute. For example:

```
SDF_KEY_MAX_LEN 65
```

Workbench Parameter: [<WorkbenchParameter>](#)

PRECISION

Required/Optional: *Optional*

This directive sets the number of bits of accuracy which will be stored for each coordinate output. If an existing SDF file is being appended to, it is ignored and the precision is set to pre-existing precision. An example specification is:

```
SDF_PRECISION 64
```

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), special FME feature attributes direct the SDF writer as it adds features to its output files, and are used by the SDF reader to store the characteristics of the features it reads. The most important of these is the `sdf_type` attribute, which controls the overall interpretation of the feature. The correct values for `sdf_type` are `sdf_line`, `sdf_point`, and `sdf_polygon`. The parameters specified for each of these are described in the following subsections, and the attributes common to each are given in the following table:

Attribute Name	Contents
<code>sdf_url</code>	Specifies a URL for the line, polygon or point. Required:No Default:NULL
<code>sdf_name</code>	Specifies an internal name for the line, polygon or point. Required:No Default:NULL
<code>sdf_key</code> ^a	Specifies a key for the line, polygon or point. Required:No Default:FME will generate a key value so that this column can be used as a key field. The generated key looks like this: FME_GENERATED_<feature number> where the feature number is based on how many features have been written out previous to the current feature within that file.

- a. In previous versions of FME, `sdf_key` was called `sdf_id`. This has been changed to be consistent with the MapGuide terminology. However, FME still accepts `sdf_id` in place of `sdf_key` to accommodate backwards compatibility.

Lines

sdf_type: `sdf_line`

Features with an `sdf_type` of `sdf_line` contain single-part or multi-part linear features.

Points

sdf_type: `sdf_point`

Features with an `sdf_type` of `sdf_point` contain point features. When writing, text features originating in another format are converted to points with the name attribute being the value of the text string by default.

Polygons

sdf_type: `sdf_polygon`

Features with an `sdf_type` of `sdf_polygon` contain single-part or multi-part polygonal features.