Smallworld 3 Reader/Writer

FORMAT NOTES:

- This format is supported only by the **FME Smallworld Edition**. This includes Smallworld FME Translators 3.3 which you must purchase separately from GE Energy (formerly GE Smallworld).
- The Smallworld interface requires the addition of Magik code to a Smallworld application. This installation and configuration of the FME server is beyond the scope of this document; however, it is discussed in the GE Energy document *Smallworld–FME Translator User Guide*. This document is in the docs folder on the FME CD.
- The GE Smallworld format is supported only for Smallworld sessions based on Smallworld Core Spatial Technology version 3.3. For Smallworld sessions based on Smallworld Core version 4, use the GE Energy Smallworld format.

This chapter explains how the Feature Manipulation Engine (FME) reads and writes data to a Smallworld Geographic Information System (GIS).

Smallworld Quick Facts

Format Type Identifier	SWORLD
Reader/Writer	Both
Licensing Level	FME Smallworld Edition
Dependencies	See Format Notes
Dataset Type	Database
Feature Type	Class name
Typical File Extensions	N/A
Automated Translation Support	No
User-Defined Attributes	Yes
Coordinate System Support	No
Generic Color Support	No
Spatial Index	Never
Schema Required	Yes
Transaction Support	No
Geometry Type	sworld_geom_type

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

Geometry Support			
Geometry	**Supported?**	**Geometry**	**Supported?**
line	yes	z values	yes
none	yes		

Overview

FME communicates with a Smallworld GIS process through a special server executing within the Smallworld process. The communication operates using one of two methods.

The first method for the communication is for the FME to run as a Smallworld Alien Co-Process (ACP). The second method is to communicate through Smallworld, Inc.'s Transport Independent Client/Server (TICS) package. The TICS package provides an architecture-independent transport of information between Smallworld and other applications – in this case, the FME – through Transmission Control Protocol/Internet Protocol (TCP/IP) networking. This is a more flexible method, but TICS requires a special license from GE Smallworld, Inc., and the TCP/IP connection is inherently slower and less secure than the ACP method.

To communicate with a Smallworld GIS, the FME only needs to be told each of these:

- which mode will be used for the communication – ACP or TICS – and for TICS, the name, or IP address, of the computer executing the GIS
- the TCP port number on which the FME Server is executing on that computer

Currently, the FME is capable of transferring the following physical, logical and geometrical attribute types to and from Smallworld:

- character strings `(ds_charci_vec, ds_chari_vec, etc.)`
- floating point numbers `(ds_float, etc.)`
- integers `(ds_int, etc.)`
- Boolean `(ds_bool)`
- vectors of float (`ds_float`), integer (`ds_int`), boolean (`ds_bool`)
- points
- chains, with line, arc, circle, spline, and ellipse sectors
- areas
- text features
- date
- join fields: one-to-one and one-to-many, read-only

Configuration Keywords

The FME's Smallworld module is configured by a few special keyword or value pairs in the mapping file. These keywords are:

- `SWORLD_SERVER` In TICS mode, this keyword defines the name or IP address of computer running Smallworld GIS (and FME Server). In ACP mode, this keyword should have the value `ACP`.
- `SWORLD_PORT` In TICS mode, this is the TCP port on which the FME server is listening for connections. In ACP mode, this keyword should have the value `ACP`.

Reader Overview

The Smallworld reader requires no configuration in addition to the standard configuration keywords listed above. When invoked, the reader will connect to the FME Server and begin downloading features. The server will interface with the GIS user to determine what features—all, current selection, etc.—should be downloaded.

Each feature read from Smallworld will be emitted in the form described under the heading *Feature Representation*.

Writer Overview

The Smallworld writer must be given the definitions of the Smallworld objects before it may start writing them out. This is accomplished by the `SWORLD_DEF` keyword. It has the following format:

```
SWORLD_DEF <object name> \
    [<attribute name> <attribute type>] *
```

The attribute names and types must match corresponding attributes in Smallworld's definition of the object. The attribute types come from the following set:

Attribute Type	Description
`char(<len>)`	Character string with a specified maximum length.
`char`	Character string with a default maximum length (currently set to 254). This type should be used only for testing purposes, and not for production mapping files; for most cases, use the `char(<len>)` form above.
`float`	Floating point number.
`float(<len>)`	Vector of floating point numbers, with a length of `<len>` values. These are represented in the FME by a comma-separated list. This list must have exactly `<len>` values in it. If the feature being written has attributes named `<attribute name>{0}`, `<attribute_name>{1}`, etc. then these values will take precedence over the comma-separated list in `<attribute name>`.
`int`	Integer
`int(<len>)`	Vector of integers, with a length of `<len>` values. These are represented in the FME by a comma-separated list. This list must have exactly `<len>` values in it. If the feature being written has attributes named `<attribute name>{0}`, `<attribute_name>{1}`, etc. then these values will take precedence over the comma-separated list in `<attribute name>`.
`boolean`	Boolean value, represented by `1` for true or `0` for false.
`boolean(<len>)`	Vector of booleans, with a length of `<len>` values. These will be represented in the FME by a comma-separated list. This list must have exactly `<len>` values in it.

Attribute Type	Description
enum(<enum name>)	This is an enumerated attribute. In the FME, enumerated attributes simply contain a string with the symbolic value of the enumerated type. An enumerated attribute must contain a valid value when being sent to Smallworld.
enum	Alternate form of enum(<enum name>), which matches any enumerated type. The values for these attributes must be valid for the actual Smallworld attribute to which they map.
join	Simple join field. The value of this attribute is a single database join reference. Available only when reading from Smallworld.
joinm	Multiple join field available only when reading from Smallworld. The value of this attribute is a comma-separated list of zero or more database join references. If the feature being written has attributes named <attribute name>{0}, <attribute_name>{1}, etc. then these values will take precedence over the comma-separated list in <attribute name>.
date	Date and time attribute. The value of <attribute name> is in the form year, month, day: YYYYMMDD. If an attribute named <attribute name>.full is defined on the feature, it has the form year, month, day, hour, minute, second—YYYYMMDDhhmmss— and will be used instead of <attribute name>.
sworld_point	A geometrical attribute defining a point, scale, and orientation or rotation.
sworld_chain	A geometrical attribute defined by a series of linear parts or sectors.
sworld_area	A geometrical attribute defined by a series of linear parts. The area completely contained by the joined linear parts is the geometry of the attribute.
sworld_text	A text attribute containing a position, a text string, height, justification, and orientation or rotation.

A Smallworld object may contain zero or more geometrical attributes and each of these may be defined or may be left unset. The *Geometry Representation* discussion, found a little later on in this chapter, explains how FME geometry is mapped onto geometric attributes such as sworld_point, sworld_chain, sworld_text, and sworld_area.

Feature Representation

There is a one-to-one correspondence between FME features and Smallworld objects. The feature type of the FME feature is the same as the external name given to the Smallworld object definition. In addition, the sworld_table_name attribute is set to the name of the Smallworld object. Each attribute in the Smallworld object is represented by one or, for geometric attributes, several attributes on the corresponding FME feature. The names of physical attributes in FME are identical to those in Smallworld.

The FME uses the representations shown in the following table for Smallworld's physical attributes' values. Vectors, multiple joins, and dates have special representation in the feature's attributes and are discussed under the headings *Vectors and Multiple Join*

Attributes and *Date and Time Attributes* found later in this chapter. Geometric attributes are discussed under the heading *Geometry Representation*.

Physical Attribute Type	Representation (by examples)
character string	"Bourbon Street"
integer	123
float	327.44556
boolean	1
enumerated type	"toll_highway"
integer vector	123,345,445,111
float vector	193.03,123.33,123.444
single join field	7058127
multiple join field	7058127,7058190
date field	"19990304"

Vectors and Multiple Join Attributes

Smallworld fields that can contain a collection of values—integer and float vectors, and multiple join fields—are represented in two forms on the FME feature.

The first is the comma-separated list depicted in the preceding table. For vectors, this table always contains exactly as many elements as are defined for the vector in the table's schema. For multiple joins, this list contains as many elements as there are joins in the Smallworld object. If a join field has no joins, its corresponding attribute on the FME feature will be an empty string.

The second representation of multiple-value fields is as a list of attributes. When reading a vector or multiple join field, the Smallworld reader creates attributes `<attrName>{0}`, `<attrName>{1}`, etc. Each `<attrName>{n}` attribute contains a single element from the comma-separated list. When writing to Smallworld, the writer first looks for an attribute named `<attrName>{0}` on the feature being written. If this attribute exists, the writer will ignore the contents of the comma-separated value attribute and use the values of `<attrName>{0}`, `<attrName>{1}`, etc. to define the output field values instead.

Date and Time Attributes

When a date field is read by the Smallworld reader, two attributes are set in the FME feature.

The first attribute is simply the name of the attribute, as specified by `YYYYMMDD`. This is compatible with all other FME dates.

The second attribute has a suffix of `.full` and is of the form `YYYYMMDDHHMMSS`. It specifies the date and the time, with the time portion specified using the 24-hour clock.

For example, if a date field called `update_time` is read, the following attributes will be set in the retrieved FME feature:

```
update_time ='19980820'
update_time.full='19980820201543'
```

When writing to Smallworld, the writer looks for both attributes. Either may be in the form of `YYYYMMDD` or `YYYYMMDDHHMMSS`. If both attributes are specified, then the value specified in `update_time.full` is used.

Geometry Representation

There are two concepts to be conquered in understanding the representation of Smallworld geometries within the FME:

- the representation of an individual geometric attribute – point, text, chain, or area
- the handling of multiple geometric attributes on a single feature

Multiple Geometric Attributes

Smallworld objects can have very complex geometries associated with them. A single object can have any number of geometries of varying geometric types associated with it. To represent this in FME, it is necessary to use aggregates of geometry.

Aggregates contain a list of geometric components. For certain geometric attribute types – chains and areas – it is also possible for a component of the aggregate to be an aggregate itself. The handling of such structured geometry is discussed below, in the description of the representation of chain geometries. In general, however, when a feature has an aggregate geometry, each component of the aggregate relates to a single geometric attribute.

In order to correlate each component of an aggregate with a particular geometric attribute, additional attribution is given to the FME feature, as follows:

- `sworld_geometry{<n>}.sworld_name`
  Name of corresponding Smallworld geometric attribute for component `<n>` of the aggregate.
- `sworld_geometry{<n>}.sworld_type`
  Type of corresponding Smallworld attribute for component `<n>` of aggregate. Legal values are:

`sworld_chain, sworld_area, sworld_point, and sworld_text`

The components of the aggregate are numbered starting at 0. Therefore, if an FME feature is representing a Smallworld object with two defined geometric attributes, its geometry will be an aggregate with two components; each component will be the geometry for one of the geometric attributes on the Smallworld object. To identify the geometries to the FME, the feature must also contain the following attributes:

```
sworld_geometry{0}.sworld_name    attrName1
sworld_geometry{0}.sworld_type    geomType1
sworld_geometry{1}.sworld_name    attrName2
sworld_geometry{1}.sworld_type    geomType2
```

The count and ordering of the geometries within the aggregate must match the count and ordering of the `sworld_geometry{}` attribute lists exactly. There is no ordering of geometries implied by attribute naming or position within the `SWORLD_DEF` lines.

Certain geometric types require additional attributes. These are discussed in depth under the heading *Individual Geometric Attribute Representation* which directly follows this chapter. If a particular geometric attribute requires additional information, it must be provided in an attribute whose name begins with the correct `sworld_geometry{}` specification.

For example, if the first geometric attribute requires an orientation, the FME feature must contain an attribute like:

```
sworld_geometry{0}.sworld_orientation   23.44
```

In its simplest case, a Smallworld object has only a single geometric attribute or the Smallworld writer supplies only a single geometric attribute to an object. In these cases, it is possible for the FME feature to contain a single, non-aggregate geometry for the object. However, there must still be a complete accompanying set of `sworld_geometry{0}.xxx` attributes defined on the feature.

When reading data from Smallworld, the FME always returns the geometry as an aggregate. Processing this aggregate can prove to be a challenge to the most worthy of FME mapping file designers. For this reason, the `SmallworldGeometryFactory` has been developed to do most of the necessary processing. It takes care of flattening the structured geometry represented in the aggregate and keeping the geometry-related attributes in order. It even combines consecutive sectors of chain or area sector ropes – as discussed under the headings *Chain Geometries* and *Area Geometries* – into a single FME line, polygon or donut geometry.

Mapping files generated by the FME use a `SmallworldGeometryFactory` to process the data retrieved from Smallworld. The `SmallworldGeometryFactory` is fully described in the *FME Functions and Factories* manual.

Individual Geometric Attribute Representation

There are four basic geometric types handled by the FME: point, text, chain, and area. Each of these is discussed below. The aggregate referred to in the text below is described in the previous section.

Point Geometries

In addition to the basic `(x,y[,z])` location, a Smallworld point has the following attributes defined:

Attribute Name	Meaning	Default
`sworld_orientation`	Rotation of the point's symbol, expressed in degrees counterclockwise.	0.0
`sworld_scale`	Magnification factor of the symbol.	1.0

The geometric type for a point is `sworld_point`. For example, the following FME feature is used in a Smallworld GIS to represent a fire hydrant whose hose attaches on the northeast side:

Feature type: hydrant

FME geometry: Aggregate containing one point (123.4,567.8)

Attributes:

- color gold
- street Knowles Ave
- sworld_geometry{0}.sworld_name position
- sworld_geometry{0}.sworld_type sworld_point
- sworld_geometry{0}.sworld_orientation 45.0
- sworld_geometry{0}.sworld_scale 1.0

Text Geometries

Text geometry is commonly used to provide annotation in a GIS. An FME feature representing annotation has a point geometry and the following geometric attributes:

Attribute Name	Meaning	Default
`sworld_text_string`	The textual part of the annotation.	" "
`sworld_text_height`	The annotation text height scaling factor. The scaling factor of the text height defined in the Smallworld style.	1
`sworld_orientation`	The rotation of the text, in degrees counter-clockwise.	0.0
`sworld_vert_just`	Vertical justification of the text about its location point. The value is an integer in the range 1..5. See the discussion following this table for more information.	1
`sworld_horiz_just`	Vertical justification of the text about its location point. The value is an integer in the range 1..5. See the discussion following this table for more information.	1

The geometric type for a piece of annotation is `sworld_text`. Justification of text in Smallworld is performed on a five by five grid, with the lower left corner being `(1,1)`, and the upper right corner being `(5,5)`. Each horizontal or vertical justification value is a discrete position within the grid. The position within the grid is the location of the text's point geometry with respect to the text string.

Chain Geometries

Chains and areas are somewhat more complex than point and text geometries. The complication comes from two aspects of the Smallworld geometries:

- A single chain or area is comprised of one or more sectors, which join together end-to-end to make the chain.
- Each sector can be one of four different types: line string, circular arc, circle, or ellipse.

The first problem is handled once again by using FME's aggregate geometries. A chain or area's component of the feature's aggregate geometry is itself an aggregate. This can be somewhat confusing, but it is a very powerful and necessary way to represent Smallworld data.

For example, suppose a chain is a rope of three sectors: two line strings connected by an arc. This would be represented by an aggregate geometry with three parts:

- the first line string
- a circular arc, whose first point is the same as the last point of the previous line string
- the second line string, whose first point is the same as the arc's last point

Of course, there must be a way to tell the FME how to interpret each sector. Like the multiple geometric attributes, this is handled by FME's list attributes. For this example, there must be three attributes—one for each sector—defined on the FME feature. These attributes are all located below `sworld_geometry{X}`, just like all other geometric attributes.

If this chain is the second component of an aggregate geometry, the FME feature will contain the following attributes and values to describe the chain:

Attribute Name	Value
`sworld_geometry{1}.sworld_sector{0}.sworld_sector_type`	line
`sworld_geometry{1}.sworld_sector{1}.sworld_sector_type`	arc
`sworld_geometry{1}.sworld_sector{2}.sworld_sector_type`	line

The arc also requires some additional attributes to describe it. These attributes are:

Attribute Name	Value
`sworld_geometry{1}.sworld_sector{1}.sworld_radius`	403.22
`sworld_geometry{1}.sworld_sector{1}.sworld_start_angle`	32.21
`sworld_geometry{1}.sworld_sector{2}.sworld_sweep_angle`	-47.4

The building of these complex geometries in FME mapping files will require a number of `AggregateFactories`. The extraction of geometries will require a number of `DeaggregateFactories`.

As for the aggregates representing multiple geometric aggregates, it is possible that a chain or area can be represented by a single FME line geometry. In this case, the feature given to the Smallworld writer may contain a simple line geometry rather than an aggregate containing a single line.

There are six kinds of sectors which may make up a sector rope aggregate for chain and area geometries: line, arc, circle, ellipse, elliptical arc, and spline. The structure of these geometries is described below.

Line Sector

A line is a simple string of `(x,y)` or `(x,y,z)` points. A feature containing a chain or area with a line sector must have the following attribute defined:

```
sworld_geometry{<m>}.sworld_sector{<n>}.sector_type
sworld_line
```

where `<m>` is the index of the chain or area geometry, and `<n>` is the index of the line sector within the geometry.

No additional attributes are required to define a line sector.

Arc Sector

An arc sector represents a portion of a circle. The arc is defined on the FME feature as a point geometry representing the centre of the circle to which the arc belongs, with a number of attributes to describe the arc.

The following attribute is defined on an FME feature containing an arc sector:

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_sector_type
sworld_arc
```

where `<m>` is the index of the chain or area geometry, and `<n>` is the index of the line sector within the geometry.

The following attributes are also defined on arc geometries. The attribute names will appear at the same level as the sector type; for example:

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_radius)
```

Attribute Name	Meaning	Default
`sworld_radius`	The radius of the circle containing the arc.	none
`sworld_start_angle`	The angle in degrees counterclockwise from due east denoting the position on the circle of the arc's starting point.	none
`sworld_sweep_angle`	The length of the arc, measured in counter-clockwise arc degrees. This number may be negative.	none

Elliptical Arc Sector

An elliptical arc sector represents a portion of an ellipse. The arc is defined on the FME feature as a point geometry representing the centre of the circle to which the arc belongs, with a number of attributes to describe the arc. Elliptical arcs are supported in the FME version 2.2 and higher, but require a Smallworld server that supports the FME/Smallworld level 3 protocol; at the time of writing, no code released from Smallworld supports the required level 3 protocol.

The following attribute is defined on an FME feature containing an elliptical arc sector:

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_sector_type
sworld_arc
```

where <m> is the index of the chain or area geometry, and <n> is the index of the line sector within the geometry.

The following attributes are also defined on arc geometries. The attribute names will appear at the same level as the sector type; for example:

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_prim_radius)
```

Attribute Name	Meaning	Default
sworld_prim_radius	The primary radius of the ellipse containing the arc.	none
sworld_sec_radius	The secondary radius of the ellipse containing the arc.	none
sworld_orientation	The orientation of the ellipse's primary axis, measured in degrees counterclockwise from due east.	none
sworld_start_angle	The angle in degrees counterclockwise (measured relative to the orientation of the ellipse) denoting the position on the circle of the ellipse's starting point.	none
sworld_sweep_angle	The length of the arc, measured in counterclockwise arc degrees. This number may be negative.	none

Circle Sector

A circle sector represents a full circle centered on a specific geographic point. The circle is defined in the FME feature as a point geometry representing the centre of the circle, with an additional attribute to specify its radius.

The following attribute is defined on an FME feature containing an circle sector:

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_sector_type
sworld_circle
```

where <m> is the index of the chain or area geometry, and <n> is the index of the line sector within the geometry.

The following attribute is also defined on circle geometries. The attribute name will appear at the same level as the sector type, in other words,

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_radius)
```

Attribute Name	Meaning	Default
sworld_radius	The radius of the circle.	none

Ellipse Sector

An ellipse sector represents a full ellipse centered on a specific geographic point. The ellipse is defined in the FME feature as a point geometry representing the centre of the ellipse, with additional attributes to specify its radii and orientation.

The following attribute is defined on an FME feature containing an ellipse sector:

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_sector_type
sworld_ellipse
```

where <m> is the index of the chain or area geometry, and <n> is the index of the line sector within the geometry.

The following attributes are also defined on ellipse geometries. The attribute names will appear at the same level as the sector type; for example,

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_prim_radius)
```

Attribute Name	Meaning	Default
sworld_prim_radius	The primary radius of the ellipse.	none
sworld_sec_radius	The secondary radius of the ellipse.	none
sworld_orientation	The orientation of the ellipse's primary axis, measured in degrees counterclockwise from due east.	none

Spline Sector

A spline sector represents Smallworld's rational b-spline geometry. The spline is defined in the FME feature as a line string representing the spline's fit points, and a number of attributes describing the spline's control points and other required data.

The following attribute is defined on an FME feature containing a spline sector:

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_sector_type
sworld_spline
```

where <m> is the index of the chain or area geometry, and <n> is the index of the line sector within the geometry.

The following attributes are also defined on spline geometries. The attribute names will appear at the same level as the sector type; for example,

```
sworld_geometry{<m>}.sworld_sector{<n>}.sworld_poly_degree)
```

Attribute Name	Meaning
sworld_poly_degree	The degree of the polynomial used to form the spline.
sworld_spline_type	The flag that indicates the type of the spline. It is a bitwise combination of the following values: 1. CLOSED 2. PERIODIC 4. RATIONAL 8. PLANAR 16. LINEAR

Attribute Name	Meaning
sworld_start_tangent_x sworld_start_tangent_y sworld_start_tangent_z	The (x,y,z) coordinate of the point that identifies the tangent to the start of the spline.
sworld_end_tangent_x sworld_end_tangent_y sworld_end_tangent_z	The (x,y,z) coordinate of the point that identifies the tangent to the end of the spline.
sworld_num_knots	The number of knots in the spline.
sworld_knots	The knots of the spline, as a comma-separated list.
sworld_control_x sworld_control_y sworld_control_z	Comma-separated lists of the x, y, and z control point coordinates.
sworld_control_weights	The control point weights, a comma-separated list of the weight values for each control vertex.

Area Geometries

An area is a solid region contained by a linear feature. The linear feature must be closed. That is, it must have identical start and end points, and be non-intersecting. The area is everything inside of the chain.

The representation of a Smallworld area on an FME feature is the same as the representation would be for the chain that forms the area's boundary. It is a list of the same sectors described earlier under the heading *Chain Geometries*. In fact, the only difference between chains and areas in the FME features is the value of the geometry type attribute, which for areas is:

```
sworld_geometry{<geomNum>}.sworld_type      sworld_area
```