

U.S. Census Bureau TIGER/Line Reader/Writer

Overview

The Topologically Integrated Geographic Encoding and Referencing (TIGER)/Line[®] Reader module provides the Feature Manipulation Engine (FME) with access to data in the TIGER/Line format distributed by the U.S. Census Bureau.

TIGER/Line files are a digital database of geographic features, such as roads, railroads, rivers, lakes, political boundaries, census statistical boundaries, etc., that cover the entire United States. The database contains information about these features such as their location in latitude and longitude, the name, the type of feature, address ranges for most streets, the geographic relationship to other features, and other related information. TIGER/Line files are the public product created from the Census Bureau's TIGER database of geographic formation. TIGER was developed at the Census Bureau to support the mapping and related geographic activities required by the decennial census and sample survey programs.

Note that the TIGER/Line product does not include census demographic statistics. Those are sold by the Census Bureau in a separate format not directly supported by FME, but those statistics do relate back to census blocks in TIGER/Line files.

More information on the TIGER/Line file format and data product can be found on the U.S. Census web page at:

<http://www.census.gov/geo/www/tiger/>

This website contains a detailed description of the current TIGER/Line format, with an explanation of field meaning for each feature type.

TIGER/Line Quick Facts

Format Type Identifier	TIGER
Reader/Writer	Both
Licensing Level	Directory or File
Dependencies	Base
Dataset Type	None
Feature Type	Feature role
Typical File Extensions	.rt1, .bw1
Automated Translation Support	Yes
User-Defined Attributes	Yes
Coordinate System Support	Yes
Generic Color Support	No
Spatial Index	Never
Schema Required	Yes
Transaction Support	No
Geometry Type	tiger_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	yes	z values	no
none	yes		

Reader Overview

FME considers a directory of TIGER/Line data files to be a dataset. Optionally the *.rt1 file may be used as a dataset. The directory may contain TIGER/Line files for one or more regions. The regions are counties, or county equivalents. Each county consists of a series of files with a common basename, and different extensions. For instance, county 1 in state 26 (Michigan) consists of the following file set in Tiger98.

```
TGR26001.RT1
TGR26001.RT2
TGR26001.RT3
TGR26001.RT4
TGR26001.RT5
TGR26001.RT6
TGR26001.RT7
TGR26001.RT8
```

TGR26001.RT9
TGR26001.RTA
TGR26001.RTC
TGR26001.RTH
TGR26001.RTI
TGR26001.RTP
TGR26001.RTR
TGR26001.RTS
TGR26001.RTZ

Versions

The reader was implemented to read TIGER/Line 98 files, but effort has gone into also ensuring compatibility with 1990, 1992, 1995, 1997, 1999, 2000, 2002, 2003, 2004, 2005 and 2006 TIGER/Line products.

Any TIGER/Line product from the 1990s should work with the reader, with the possible loss of some version-specific information. Attribute names are generally based on TIGER 1998 field names, even when the nominal field name in TIGER documentation is different and the interpretation has changed slightly.

Reader Directives

The directive processed by the Top10NL reader are listed below. The suffix shown is prefixed by the current `<ReaderKeyword>` in a mapping file. By default, the `<ReaderKeyword>` for the Top10NL reader is `TOP10`.

DATASET

Required/Optional: *Required*

The name of a directory containing one or more TIGER/Line county file sets, or a single TIGER/Line file. The file names should all start with `TGR`, and the extensions should not be altered from the distributed form.

```
TIGER_DATASET D:\DATA\26_MI
```

IDs

Required/Optional: *Optional*

This optional specification is used to limit the available and defined TIGER/Line files read. If no IDs are specified, then all available files are read. The syntax of the IDs directive is:

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

The example below selects only the `TGR26001.*` county file set.

```
TIGER_IDS TGR26001
```

LFIELD_AS_STRING

Required/Optional: *Optional*

This optional directive determines whether or not to treat left-justified numeric fields as strings (which will preserve the leading zeros). If the directive value is "yes" then the attribute values like "09" will be read as "09" and if the directive value is "no" then "09" will be read as "9".

Note: This directive value can only be set during mapping file generation.

The example below shows how to set the directive during mapping file generation from the command prompt.

```
> fme GENERATE TIGER NULL <sourceDataset> <outputMappingFile> --  
Source_LFIELD_AS_STRING no
```

Writer Overview

Note: To use the TIGER/Line writer, you will have to create a custom mapping file. TIGER/Line writing is not supported in Workbench.

The TIGER/Line writer considers a directory of TIGER/Line data files to be a dataset. A logical TIGER file consists of a series of files with a common basename, and different extensions. Typically, a logical TIGER file represents a county or other demographic region.

The writer has a fixed schema (the same as that generated by the TIGER/Line reader), so all output must be matched to one of the predefined feature types:

- CompleteChain
- AltName
- FeatureIds
- ZipCodes
- Landmarks
- AreaLandmarks
- KeyFeatures
- Polygon
- EntityNames
- IDHistory
- PolyChainLink
- PIP
- ZipPlus4

Any attempt to write features with other feature types will result in an error.

The written features are divided into logical file sets on the basis of the required `MODULE` attribute that must appear on each feature. The `MODULE` attribute typically takes the form of `TGR<state#><county#>`; for example, `TGR26001` for county 1 in Michigan (FIPS

state code 26). Any features that lack a `MODULE` value cannot be written. If the county and state codes are unknown, you can use a `MODULE` value of `Tgr00000`.

Note:

- File organization is written in TIGER 2000 format regardless of `VERSION` setting. The actual attribute sets and underlying file format vary slightly for `Polygon` and `EntityNames` features.
 - For any counties for which features are written, all TIGER files for that county will be deleted in advance by the writer. Files for counties not written in a given run will be left untouched.
 - If some feature types are not written, the corresponding TIGER/Line files will not be written; they will not be created with zero length.
 - Consistency checking is not performed on features, for instance to ensure that TIGER/Line IDs (TLIDs) appear within the ranges indicated by a `TLIDRange` feature, or that polygon links can be resolved.
 - Geometry is not required on Polygon objects; however, to produce an output TIGER/Line dataset with reconstructable geometry, it is necessary to write appropriate `PolyChainLink` features, and corresponding `CompleteChains`.
-

Writer Directives

The directives processed by the TIGER/Line writer are listed below. The suffixes shown are prefixed by the current `<WriterKeyword>_` in a mapping file. By default, the `<WriterKeyword>` for the TIGER/Line writer is `TIGER`. Note that there is no support for an `IDs` directive in the TIGER/Line writer.

DATASET

Required/Optional: *Required*

The name of the directory into which TIGER/Line county file sets should be written.

VERSION

Required/Optional: *Optional*

The version code written to the file. The default is 1000, indicating Redistricting Census 2000. Legal `VERSION` values fall into the following ranges:

0000	TIGER/Line Precensus Files, 1990
0002	TIGER/Line Initial Voting District Codes Files, 1990
0003	TIGER/Line Files, 1990
0005	TIGER/Line Files, 1992
0021	TIGER/Line Files, 1994
0024	TIGER/Line Files, 1995
0697 to 1098	TIGER/Line Files, 1997
1298 to 0499	TIGER/Line Files, 1998
0600 to 0800	TIGER/Line Files, 1999
1000 to 1100	TIGER/Line Files, Redistricting Census 2000

Note that the actual file format is that of Redistricting Census 2000 (this affects `Polygon` and `EntityNames` features) so it is generally unwise to use version values preceding census 2000.

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.

With a few exceptions, an FME feature is created for each record of a TIGER/Line data file. Each file, that is, `.RT1`, `.RTA`, is translated to an appropriate FME feature type, with attribute names matching those in the TIGER/Line product manual.

The TIGER/Line `RT` (record type) and `VERSION` attributes are generally discarded, but the `MODULE` attribute is added to each feature. The `MODULE` attribute contains the file base name, for example, `TGR26001`, of the county from which the feature originated. For some keys, such as `LAND`, `POLYID`, and `CENID`, this `MODULE` attribute is needed to make the key unique when the `DATASET` directory consists of more than one logical file, i.e. more than one county of data.

Following subsections provide a list of feature types and their relationship to the TIGER/Line product.

CompleteChain

A `CompleteChain` is a polyline with an associated TLID (TIGER/Line ID). The `CompleteChain` features are established from a type 1 record (Complete Chain Basic Data Record) and, if available, its associated type 3 record (Complete Chain Geographic Entity Codes). As well, any type 2 records (Complete Chain Shape Coordinates) available are used to fill in intermediate Shape points on the arc. The TLID is the primary key and is unique within the entire national TIGER/Line coverage.

These features always have a line geometry.

AltName

These features are derived from the type 4 record (Index to Alternate Feature Identifiers), and relate a TLID to 1 to 4 alternate feature name numbers (the `FEAT` attribute) which are kept separately as `FeatureIds` features.

The standard reader pipeline attaches the names from the `FeatureIds` feature attributes `ALT_FEDIRS`, `ALT_FEDIRP`, `ALT_FENAME`, and `ALT_FETYPE` to the `CompleteChain` features. The `ALT_FENAME` is a list of feature names associated with the TLID on the `AltName` feature, with the alternate names separated by commas.

Note that 0, 1 or more `AltName` records may exist for a given TLID and that each `AltName` record can contain between one and four alternate names. Currently `CompleteChains` will only be updated with the alternate names from one of the related `AltName` records, potentially resulting in the loss of some alternate names. Future versions of the TIGER/Line reader will add support for multiple `AltName` records.

These features have no associated geometry.

FeatureIds

These features are derived from type 5 (Complete Chain Feature Identifiers) records. Each feature contains a feature name (`FENAME`), and its associated feature ID code (`FEAT`). The `FEAT` attribute is the primary key and is unique within the county module. `FeatureIds` have a one-to-many relationship with `AltName` features, and `KeyFeatures` features.

These features have no associated geometry.

ZipCodes

These features are derived from type 6 (Additional Address Range and ZIP Code Data) records. These features are intended to augment the ZIP Code information kept directly on `CompleteChain` features. There is a many-to-one relationship between `ZipCodes` features and `CompleteChain` features.

These features have no associated geometry.

Landmarks

These features are derived from type 7 (Landmark Features) records. They relate to point or area landmarks. For area landmarks, there is a one-to-one relationship with an `AreaLandmark` record. The `LAND` attribute is the primary key and is unique within the county module.

These features may have an associated point geometry. Landmarks associated with polygons do not have the polygon geometry attached. It needs to be collected by way of the `AreaLandmark` feature from a `Polygon` feature.

AreaLandmarks

These features are derived from type 8 (Polygons Linked to Area Landmarks) records. Each associates a `Landmark` feature (attribute `LAND`) with a `Polygon` feature (attribute `POLYID`). This feature has a many-to-many relationship with `Polygon` features.

These features have no associated geometry.

KeyFeatures

These features are derived from type 9 (Polygon Geographic Entity Codes) records. They may be associated with a `FeatureIds` feature by way of the `FEAT` attribute and a `Polygon` feature by way of the `POLYID` attribute.

These features have no associated geometry.

Polygon

These features are derived from type A (Polygon Geographic Entity Codes) records and, if available, the related type S (Polygon Additional Geographic Entity Codes) records. The `POLYID` attribute is the primary key, uniquely identifying a polygon within a county module.

These features have an associated polygon geometry, automatically collected by the standard reader pipeline. It is collected from `CompleteChain` geometries based on the type I (`PolyChainLink`) records.

EntityNames

These features are derived from type C (Geographic Entity Names) records and have no associated geometry.

IDHistory

These features are derived from type H (TIGER/Line ID History) records. They can be used to trace splitting, merging, creating and deleting of `CompleteChain` features.

These features have no associated geometry.

PolyChainLink

These features are derived from type I (Link Between Complete Chains and Polygons) records. They are normally all consumed by the standard reader pipeline while attaching `CompleteChain` geometries to `Polygon` features to establish their polygon geometries. `PolyChainLink` features have a many-to-one relationship with `Polygon` features, and a one-to-one relationship with `CompleteChain` features.

These features have no associated geometry.

PIP

These features are derived from type P (Polygon Internal Point) records. They relate to a `Polygon` feature by way of the `POLYID` attribute and can be used to establish an internal point for `Polygon` features.

These features have a point geometry.

ZipPlus4

These features are derived from type Z (ZIP+4 Codes) records. `ZipPlus4` features have a many-to-one relationship with `CompleteChain` features.

These features have no associated geometry.

Mapping File Example

```
# =====  
# Create TIGER/Line dataset  
#  
# =====  
  
GUI TITLE SHAPE to TIGER/Line Translation  
  
# =====  
# LOG_FILENAME translation.log  
# LOG_APPEND NO
```

```

# =====
# define the type of reader and writer

READER_TYPE SHAPE

WRITER_TYPE TIGER

# =====
# prompts for a directory to be used as the source of the ESRI SHAPE files.

# The dataset this mapping file was generated from was:
DEFAULT_MACRO SourceDataset D:\MarksWork\City of Sakatoon\sourceSLSN\

GUI DIRNAME SourceDataset Original Shape File Directory:

SHAPE_DATASET "$(SourceDataset)"

# =====
# prompts for a directory to be used as the destination for the TIGER directory.

GUI DIRNAME DestDataset Destination TIGER Directory:

# =====

TIGER_DATASET "$(DestDataset)"

# =====
# The main body of the mapping file starts here.
# =====

# Define the coordinate systems
MACRO sourceCoord      UTM27-13
MACRO destinationCoord LL-27

# set leading character to spaces. Use "%9s" to set the leading character to
zeros.
# required for some TIGER fields.
TCL set formatString2 "%2s"
TCL set formatString6 "%6s"
TCL set formatString9 "%9s"
TCL set formatString20 "%20s"

# Reproject the Coordinates
FACTORY_DEF * TeeFactory \
    FACTORY_NAME "Extract From/To LL" \
    INPUT FEATURE_TYPE * \
    OUTPUT FEATURE_TYPE * \
    @Reproject($(sourceCoord),$(destinationCoord))

# Set the attributes & format them with leading spaces
FACTORY_DEF * TeeFactory \
    INPUT FEATURE_TYPE * \
    OUTPUT FEATURE_TYPE CompleteChain \
    @SupplyAttributes(MODULE,Tgr00000) \
    @SupplyAttributes(TLID,@Count(TLID,0)) \

```

```

        @SupplyAttributes(tiger_type,tiger_line) \
        @SupplyAttributes(FENAME,&STNAME) \
        @SupplyAttributes(FETYPE,&STEXTEN) \
        @SupplyAttributes(FEDIRS,&STDIR) \
        @SupplyAttributes(FRLONG,@Evaluate("round(@Coordinate(x,0)*1000000)"))
\
        @SupplyAttributes(FRLAT,@Evaluate("round(@Coordinate(y,0)*1000000)"))
\
@SupplyAttributes(TOLONG,@Evaluate("round(@Coordinate(x,END)*1000000)"))\
    @SupplyAttributes(TOLAT,
@Evaluate("round(@Coordinate(y,END)*1000000)"))\
    @TCL("set FME_Attributes(FENAME) [format $formatString20
$FME_Attributes(FENAME)]")\
    @TCL("set FME_Attributes(FETYPE) [format $formatString20
$FME_Attributes(FETYPE)]")\
    @TCL("set FME_Attributes(FEDIRS) [format $formatString2
$FME_Attributes(FEDIRS)]")\
    @TCL("set FME_Attributes(FRLONG) [format $formatString9
$FME_Attributes(FRLONG)]") \
    @TCL("set FME_Attributes(FRLAT) [format $formatString9
$FME_Attributes(FRLAT)]") \
    @TCL("set FME_Attributes(TOLONG) [format $formatString9
$FME_Attributes(TOLONG)]") \
    @TCL("set FME_Attributes(TOLAT) [format $formatString9
$FME_Attributes(TOLAT)]")

# Remove redundant attributes - those not used by TIGER
FACTORY_DEF * TeeFactory \
    FACTORY_NAME "Clear-up attributes" \
    INPUT FEATURE_TYPE * \
    OUTPUT FEATURE_TYPE * \
    @RemoveAttributes(LINKNO,RCSSEG,STNAME,STEXTEN,STDIR,CLASS,PRIORITY,
CONST_YE,STATUS)

# =====

SHAPE_DEF Flame \
    SHAPE_GEOMETRY shape_arc \
    LINKNO char(254) \
    RCSSEG char(254) \
    STNAME char(254) \
    STEXTEN char(254) \
    STDIR char(254) \
    CLASS char(254) \
    PRIORITY char(254) \
    CONST_YE char(254) \
    STATUS char(254)

SHAPE *
TIGER * @Log(TIGER,10,2)

```