

GeoTask Server Reader/Writer

FORMAT NOTES:

- This format is not supported by FME Base Edition.

The GeoTask Server reader and writer module enables the Feature Manipulation Engine (FME) to retrieve geographic and tabular data, and to integrate data from any FME source onto the GeoTask Server.

The GeoTask Server is an object extension to the relational database management system (RDBMS) IBM® DB2® Universal Database (DB2). It allows the management of geographic data directly on the Structured Query Language (SQL) interface designed for the latest open standards in Geographic Information Systems (GIS).

With its Open Database Connectivity (ODBC) and Java DataBase Connectivity (JDBC) interfaces, and with the SQL query facilities within FME mapping files, the GeoTask Server can be used as the geographic data warehouse for client/server and web-based geodata applications.

The advantage of the GeoTask approach over traditional GIS data storage is that geographic data management is fully integrated into the mainstream data management technology based on relational database systems.

GeoTask Server Quick Facts

Format Type Identifier	GEOTASK
Reader/Writer	Both
Licensing Level	Professional
Dependencies	None
Dataset Type	Database
Feature Type	Table name
Typical File Extensions	Not applicable
Automated Translation Support	Yes
User-Defined Attributes	Yes
Coordinate System Support	No
Generic Color Support	No
Spatial Index	Always
Schema Required	Yes
Transaction Support	No
Geometry Type	gt_type

Geometry Support			
Geometry	Supported?	Geometry	Supported?
aggregate	no	point	yes
circles	no	polygon	yes

Geometry Support			
Geometry	Supported?	Geometry	Supported?
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		

Overview

The GeoTask Server provides a seamless data model for the normalized storage of geographic data. Geographic features are represented as rows in relational tables with one or several geometric attributes. The GeoTask Server imposes geometric constraints on feature geometries in order to respect the geometry model of the OpenGIS simple features specifications.

With an additional *OpenGISTestFactory*¹, features can be tested for their compliance with the rules and, if rejected, be corrected with FME factories or in the system of origin.

GeoTask Server features are stored in tables of the RDBMS. If tables with the names of FME features exist in the database schema, the writer inserts features into the respective table. If tables for the FME features do not exist before the FME session, the translator creates all needed tables with their triggers and fills them accordingly. This allows you to create entire database schemas with only one FME session. This makes it particularly interesting for the transfer and representation of complex data formats like INTERLIS or SAIF. The reader module reads one table at a time and allows the specification of SQL *WHERE* clauses to select specific features based on attribute or geometric constraints, including topological queries.

The GeoTask Server accepts point, linestring, polygon, and geometry collection types (complex or aggregate geometry) as geometry types with two-dimensional (2D) and three-dimensional (3D) coordinates. For attribute types, all types the RDBMS are supported. If features without geometry are passed to the GeoTask Server, it serves as an interface to an ordinary relational table.

FME GeoTask Highlights

The GeoTask Server reader and writer modules make the FME a client application to the GeoTask Server. In batch mode, the FME can be used as a middleware component to connect desktop GIS software packages to the GeoTask Server.

The GeoTask modules deliver the following capabilities:

- **Creation of geographically-enabled relational tables on the fly:** Single tables and entire database schemas are created automatically by the FME to accommodate all feature types specified in an FME mapping file.

1. *OpenGISTestFactory* is not included in the FME, however it is available through GeoTask AG in Switzerland..

- **Verification of geometry:** Geometric attributes not following the OpenGIS simple features specifications are rejected by the translator or can be filtered out using the *OpenGISTestFactory*.
- **Transaction support:** Table inserts can be stopped on insertion of erroneous data.
- **Attribute query support:** Features can be retrieved according to attribute constraints following the SQL `WHERE` clause syntax.
- **Spatial query support:** Features can be retrieved according to attribute constraints following the OpenGIS spatial functions syntax.

Tip:

- Applied in conjunction with the FME Multi-Reader, the GeoTask Server can be used as a geographic data warehouse to integrate data from heterogeneous sources with relational tables.
 - The combination of SQL and OpenGIS query syntax allows for precise and efficient selection of highly specific features from large sets of data.
-

Reader Overview

The GeoTask Server reader module is a thin-client¹ application responsible for retrieving features from GeoTask Server tables. Database, schema, and table names need to be specified as well as a valid user identification (ID) and a password to connect to the RDBMS. The database system handles the SQL `WHERE` condition and passes only valid features back to the FME. SQL `WHERE` clauses can be composed using any combination of SQL 3 and OpenGIS selection criteria.

Reader Directives

The suffixes listed are prefixed by the current `<ReaderKeyword>` in a mapping file. By default, the `<ReaderKeyword>` for the GeoTask Server reader is `GEOTASK`.

SERVER

Required/Optional: *Required*

This statement identifies an ODBC database on a GeoTask Server machine to which a connection can be established.

Parameter	Contents
<code><ODBC server></code>	The ODBC database name or its alias. ODBC connections are specified by an administrator with the Client Configuration Assistant of DB2.

Example:

```
GEOTASK_SERVER dbgeo1
```

1. Thin-client is a term used to refer to programs that communicate with a server somewhere. The server does all of the work; therefore, the client program is a small, or thin, one.

USERID**Required/Optional:** *Required*

Specify the user ID for the connection to the database.

Tip: Do not use user IDs longer than eight characters on DB2.

Parameter	Contents
<user id>	User ID required to connect to the database which is usually the same as that used to log on to the network.

Example:

GEOTASK_USERID userjoe

PASSWORD**Required/Optional:** *Required*

This is needed to specify the user ID password that allows connection to the database.

Parameter	Contents
<password>	The password for the user ID. Usually this is the same password required to log on to the workstation or to the network.

SCHEMA**Required/Optional:** *Required*

This statement identifies the database schema to work with. Do not use user IDs longer than eight characters on DB2.

Parameter	Contents
<database schema name>	The database schema where you want to access tables.

Example:

GEOTASK_SCHEMA bigproject

TABLES**Required/Optional:** *Required*

This statement identifies the database tables to transfer from the GeoTask Server.

Parameter	Contents
-----------	----------

<table(s)>	The name of a specific table or a list of table names separated by ":".
------------	---

Example:

```
GEOTASK_TABLES mytable1 : mytable2
```

SQL_STATEMENT**Required/Optional:** *Optional*

This statement provides you with the capability to retrieve specific rows of a table.

Parameter	Contents
<table name>	The name of the table where the selection statement is applied.
<where clause>	Any SQL WHERE clause statement compliant with SQL 3 and/or the OpenGIS Simple Feature Specifications for SQL, Rev. 1.0.

Example:

```
GEOTASK_SQL_STATEMENT districts (pop > 1000) AND (within(perimeter,
polygonfromtext('POLYGON (0 0, 10 0, 5 10)')))
```

DEF**Required/Optional:** *Required*

GeoTask Server tables must be defined before they can be read. The definition specifies the table name, the GeoTask Server geometry type, and the names and types of the attributes. The syntax of a GEOTASK DEF line is:

```
<ReaderKeyword>_DEF <tableName> \
    GT_GEOMETRY <geometryType> \
    [<attrName> <attrType>]+
```

The following table shows the geometry types supported:

Geometry Type	Description
gt_none	Database table without geometry.
gt_point	Database table with point geometry.
gt_polyline	Database table with polyline geometry according to the definition of the linestring type of the OpenGIS simple features specifications.
gt_polygon	Database table with polygon geometry according to the definition of the polygon type of the OpenGIS simple features specifications.

This table shows the attribute types supported:

Field Type	Description
<code>char(<width>)</code>	Character fields (strings) with a maximum of width characters.
<code>number (<width>,<decimal>)</code>	Number fields store fixed point numeric values. Width is the maximum number of fields including the decimal point. Decimal is the number of fields after the decimal point.
<code>float</code>	Float fields store floating point numbers of four byte precision.
<code>double</code>	Double fields store floating point numbers of eight byte precision.
<code>integer</code>	Integer fields store 32 bit signed integers.
<code>smallint</code>	Small integer fields store 16 bit signed integers.
<code>boolean</code>	Boolean fields store TRUE/FALSE data.
<code>date</code>	Date field stores dates in the SQL date data type. Only FME strings in the format YYYYMMDD can be used for the date format.

The following mapping file fragment specifies a GeoTask Server table with point geometry - GEOTASK_SCHEMA bigproject:

```
GEOTASK_DEF address \
  GT_GEOMETRYgt_point \
  adr_idinteger \
  str_code number(4,0) \
  house_nr char(5)
```

Tip: Remember: table and attribute names are limited to 18 characters.

Writer Overview

Writer Directives

This section lists the directives processed by the GeoTask Writer. The suffixes shown are prefixed by the current `<WriterKeyword>` in a mapping file. By default, the `<WriterKeyword>` for the GeoTask writer is GEOTASK.

SERVER

Required/Optional: *Required*

The database accessed with ODBC.

USER_ID**Required/Optional:** *Required*

The User ID to connect to the database.

PASSWORD**Required/Optional:** *Required*

The password for the user account.

SCHEMA**Required/Optional:** *Required*

The desired database schema.

ON_ERROR**Required/Optional:** *Required*

This statement tells the database whether the FME should abort or continue the transfer if an error occurs, and whether the transfer should be committed or rolled back after an error was detected. This parameter is useful if you want to test a file for additional errors before stopping translation yet without committing features translated after the first error. For technical reasons, transactions are committed after every 50 features.

Parameter	Contents
<procedure to follow in case of an error>	Either <code>ABORT</code> to stop translation or <code>CONTINUE</code> to continue translation, and either <code>ROLLBACK</code> to rollback all following features even if they are correct or <code>COMMIT</code> to commit all good features.

Example:

Here are examples of all four possibilities:

```

GEOTASK_ON_ERROR ABORT : ROLLBACK
GEOTASK_ON_ERROR CONTINUE : ROLLBACK
GEOTASK_ON_ERROR ABORT : COMMIT
GEOTASK_ON_ERROR CONTINUE : COMMIT

```

WRITER_MODE**Required/Optional:** *Optional*

Note: For more information on this directive, see the chapter *Database Writer Mode* on page 19.

This statement enables a user to decide to insert into, delete from, or update data in a database table. The current version of the GeoTask Server writer module only supports the `INSERT` mode.

Default value: *INSERT*

Example:

GEOTASK_WRITER_MODE INSERT

SRID

This statement specifies the default Spatial Reference System ID (coordinate system) for the coordinates written.

Parameter	Contents
SRID	A spatial reference ID or AUTO

Example:

GEOTASK_SRID Auto

DEF_REG

This parameter specifies whether or not to register tables in metadata on write. This only needs to be done for new tables that are not already in the database.

Parameter	Contents
DEF_REG	YES or NO The default is NO.

Example:

GEOTASK_DEF_REG YES

PRECISION

This parameter specifies the default precision for coordinates. the default value is nine (9) significant figures.

Parameter	Contents
PRECISION	An integer number. The default is 9.

Example:

GEOTASK_PRECISION 12

STR_TYP

This parameter specifies whether or not to use strong geometry types for writing to GeoTask.

Parameter	Contents
-----------	----------

STR_TYP	YES or NO. The default is YES.
---------	-----------------------------------

Example:

```
GEOTASK_STR_TYP YES
```

INDEX

The following parameter specifies whether or not to create a spatial index on the table(s). This provides extent information in GeoTask.

Parameter	Contents
WRITER_MODE	YES or NO. The default is NO.

Example:

```
GEOTASK_INDEX NO
```

DEF_FSET

The following parameter specifies whether or not to declare a default feature set for the table(s) written out.

Parameter	Contents
DEF_FSET	YES or NO The default is NO.

Example:

```
GEOTASK_DEF_FSET NO
```

DEF_LAYER

This parameter specifies whether or not to create a default layer for the table(s) in the default legend.

Parameter	Contents
DEF_LAYER	YES or NO The default is NO.

Example:

```
GEOTASK_DEF_LAYER NO
```

DEF_DFORM

This parameter specifies whether or not to create a default display form for the table(s).

Parameter	Contents
DEF_DFORM	YES or NO The default is NO.

Example:

GEOTASK_DEF_DFORM NO

DEF_EFORM

This parameter specifies whether or not to create a default editing form for the table(s).

Parameter	Contents
DEF_EFORM	YES or NO The default is NO.

Example:

GEOTASK_DEF_EFORM NO

DEF_QFORM

This parameter specifies whether or not to create a default query form for the table(s).

Parameter	Contents
DEF_QFORM	YES or NO The default is NO.

Example:

GEOTASK_DEF_QFORM NO

LEGEND

This parameter specifies the default GeoTask legend ID for the table(s).

Parameter	Contents
LEGEND	Any positive integer value. The default is 1.

Example:

GEOTASK_LEGEND 3

STYLE

This parameter specifies the default GeoTask style ID for the table(s).

Parameter	Contents
STYLE	Any positive integer value. The default is 1.

Example:

GEOTASK_STYLE 2

LOCALE

This parameter specifies the locale to use for generated texts. The default value is English (en), but German (de) and French(fr) are also supported.

Parameter	Contents
LOCALE	en, de or fr The default is en.

Example:

GEOTASK_LOCALE de

DEF

For attribute and geometry types supported by the GeoTask Server writer, please refer to the subsections found under *Reader Overview*.

