

Grafbat (OUT) Reader/Writer

FORMAT	• Dieses Format wird nicht unterstützt durch die FME Desktop Suite
NOTES	• Dieses Format benötigt eine installierte JAVA-Laufzeitumgebung

Überblick

Das CAD-System Geograf der Firma HHK Datentechnik Braunschweig benutzt als zum Austausch zwischen Schnittstellenprogrammen eine ASCII-Datei, welche die gesamte Geograf-Datenhaltung abbildet. Das so genannte GrafBat-Format ist in der Hilfedatei zum System Geograf dokumentiert. Die Standarddateierweiterung ist *.out.

Der Reader unterstützt die **GrafBat-Versionen 3 bis 8**, der Writer schreibt die Daten ausschließlich in der aktuellen **GrafBat-Version 8**.

Inzwischen kann der Reader auch die Daten aus den Geograf-**Katalogen** lesen. Unterstützt werden Artdateien, hier im ArtBat-Format ab Version 7, die Stiftdatei, die Leveldatei, beide liegen als Textdateien vor, und die binäre Farbdatei, in der die Bildschirmfarben festgelegt sind. Die Katalogdaten werden beim Lesen der Features verknüpft, sollen die Katalogdaten separat verarbeitet werden, kann dies über das gesondert verfügbare ArtBat-Reader-PlugIn erfolgen.

Neuerungen der Version 2017

Mit dem Erscheinen der FME 2017 wurde auch das GrafBat-Plugin aktualisiert. Reader und Writer basieren damit vollständig auf dem **erweiterten Geometriemodell** der FME. Insbesondere wirkt sich das auf die Verarbeitung der Geograf-Objekte aus. Für die atomaren Objektbestandteile werden alle Eigenschaften als Geometrieattribute gespeichert, die auch so der Writer wieder verarbeitet.

Bögen und Kreise werden jetzt auch bei Schraffuren und Objekten als solche in der Geometrie gespeichert, so dass die Umwandlung in Bogenpolygone nicht mehr notwendig ist. Einzig die Geograf-Klothoide wird als Bogenpolygon umgesetzt, da es hierfür keine Entsprechung im FME-Geometriemodell gibt.

Vollständig umgesetzt wurden auch die **3D-Eigenschaften** der Elemente, auch Schraffuren und Böschungen können mit Höhenwerten gelesen werden.

Die Geograf-**Beschriftungen** können jetzt in verschiedenen Varianten gelesen werden, die Beziehung zwischen Text und beschrifteten Element kann auch der Writer wieder erzeugen.

Die erweiterten Möglichkeiten, die sich mit der Einführung der Geograf-**Sachdaten-MDB** ergeben haben, werden jetzt auch vom Reader und Writer genutzt. Der Writer erzeugt auch relationale Tabellen, Sub-Tabellen und Picklisten. Dem Writer können jetzt bereits definierte Sachdatentabellen mit einer Template-Datei übergeben werden, so dass vorhandene Definitionen einfach genutzt werden können.

Bei **Geländemodellen** werden jetzt nicht mehr die einzelnen Dreiecke gelesen, der Reader erzeugt je Modell ein Mesh.

Die kleinen **Detaillösungen**, die mit den letzten Geograf-Versionen eingeführt wurden (Multisymbole, mehrfache Textbezüge) wurden in Reader und Writer integriert.

OUT Quick Facts

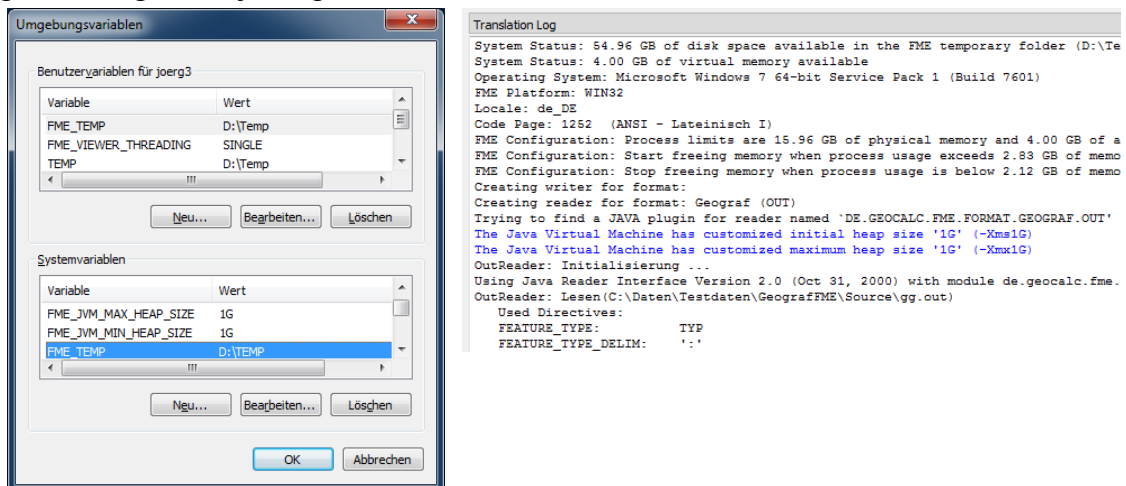
Format Type Identifier	OUT		
Reader/Writer	Both		
Dataset Type	File for Reader/Writer		
Feature Type	configurable		
Typical File Extension	.out		
Automated Translation Support	Yes		
User-Defined Attributes	Yes		
Coordinate System Support	Yes		
Generic Color Support	No		
Spatial Index	Never		
Schema Required	No		
Transaction Support	No		
Geometry Type Attribute	out_type		
Geometry Support			
Geometry	Supported	Geometry	Supported
aggregate	yes	polygon	yes
circles	yes	donut polygon	yes
circular arc	yes	line	yes
elliptical arc	no	point	yes
ellipses	no	text	yes
none	no	3D	yes

Reader Overview

Die Speicherung von Geometriedaten erfolgt in Geograf im Gegensatz zu vielen anderen Softwarelösungen punktorientiert. So haben Linien keine eigenen Koordinaten, es wird ein Verweis zum Linienanfangs- und Endpunkt gespeichert.

Komplexe Geometrien werden als Objekte gespeichert. Hierbei wird ebenfalls in einer Liste auf die atomaren Geometrieelemente verwiesen.

Der Geograf-Reader importiert die Geograf-Daten in zwei Schritten. Die OUT-Datei wird zuerst in einem Zwischenspeicher komplett geladen. Im zweiten Schritt werden die Geograf-Elemente einzeln der FME übergeben. Dabei werden die Geometrien für jedes Geograf-Feature erzeugt. So erhalten die Linien die Koordinaten der verlinkten Punkte, aus den Geograf-Objekten werden je nach Geometriertyp Linien, Flächen oder Aggregate erzeugt. Mit den Keywords kann die Geometrieerzeugung gesteuert werden. Durch das Zwischenspeichern wird im größeren Umfang Arbeitsspeicher belegt. Sollte bei größeren Projekten ein OutOfMemoryError entstehen, kann mit Hilfe den in der Systemsteuerung einstellbaren Umgebungsvariablen FME_JVM_MIN_HEAP_SIZE und FME_JVM_MAX_HEAP_SIZE der Arbeitsspeicher für die JavaRuntime reserviert werden. Mit der Einstellung beider Umgebungsvariablen auf 1G können auch sehr große Geograf-Projekte geladen werden.



Reader Keywords

Keyword Suffix	Value	Required/ Optional
DATASET	Hiermit wird die zu lesende Eingabedatei vorgegeben.	Required
FEATURE_TYPE	Mit diesem Keyword wird die Erzeugung des Feature Types gesteuert. Die Einstellung TYP erzeugt folgende Feature Types: P für Punkte L für Linien T für Texte S für Schraffuren B für Böschungen O für Objekte M für Modelle D für Dokumente G für Geländemodelle Mit der Einstellung EBENE wird der Feature Type aus der Geograf-Ebene erzeugt. Bei Einstellung ART wird die Geograf-Art benutzt. Bei dem Schalter SACH wird der FeatureType aus dem Namen des Sachdatums erzeugt. Die Einstellungen können mit dem Trennzeichen “:“ in Kombination angegeben werden. Value: TYP,EBENE,ART SACH Default: TYP Example: OUT_FEATURE_TYPE TYP ART	Optional
FEATURE_TYPE_DELIM	Wenn beim Keyword FEATURE_TYPE eine Kombination gesetzt wird, kann hiermit das vom Reader erzeugte Trennzeichen gesteuert werden. Value: beliebiges ASCII-Zeichen Default: “:“	Optional
ARC_TO_POLYLINE	Je nach Erzeugung unter Geograf generiert der Reader Bögen nach dem FME-Geometriemodell. Diese werden so auch in die Umringspolygone der Flächen übernommen. Ist dieser Schalter auf Yes gesetzt, wandelt der Reader die Bögen in Bogenpolygone. Value: Yes No Default: Yes	Optional
SPLINE_PHANTOM_POINTS	Geograf-Splines werden als <code>fme_line</code> übersetzt. Als Zwischenpunkte werden die Splinekoordinaten übernommen. Um eine annähernd gleiche Ausrundung zu erhalten, kann der Schalter auf YES gesetzt werden. Entsprechend der Angabe SPLINE_SEGMENT_POINTS werden dann weitere Zwischenpunkte interpoliert. Value: Yes No Default: No	Optional
SPLINE_SEGMENT_POINTS	Gibt die Anzahl der zu interpolierenden Zwischenpunkte bei der Splineinterpolation an. Ca. 10 Zwischenpunkte werden benötigt, um eine Darstellung analog zu Geograf zu erhalten. Value: Ganzzahlwert Default: 0	Optional
OBJECT_PARSER	Ein Geograf-Objekt ist eine Zusammenfassung von beliebigen Einzelelementen. Zum Objekt werden lediglich Referenzen auf die Einzelelemente gespeichert. In der Regel werden lediglich flächenförmige Objekte, zB. Flurstücke, erfasst. Es können aber auch beliebige Zusammenhänge zwischen Einzelelementen mit Objekten modelliert werden, so kann ein Objekt auch Einzelpunkte, Texte und Schraffuren als Bestandteile enthalten. Linien, die eine Fläche bilden sollen, speichert Geograf als Definitionsgeometrie, alle anderen Elemente werden als Ausgestaltungsgeometrie markiert. Auch linienförmige Objekte enthalten die Kanten als Ausgestaltungsgeometrie. Um diesen Umständen Rechnung zu tragen, bietet der Schalter OBJEKT_PARSER verschiedene Einlesevarianten an: NO Objekte werden nicht gelesen LINK Es wird ein <code>fme_point</code> aus der Objektkoordinate	Optional

Keyword Suffix	Value	Required/Optional
	<p>erzeugt, bzw. soweit keine Objektcoordinate vorliegt ein Objekt ohne Geometrie. Das Objekt erhält das Attribut <code>out_link_id</code>, welches als Referenzschlüssel für die verlinkten Elemente gilt. Alle Objektbestandteile erhalten die Liste <code>out_object{}.link</code>, welche die Schlüssel der Objekte, in denen es referenziert ist enthält und die Liste <code>out_object{}.linktyp</code> mit der Art der Referenzierung (Definition oder Ausgestaltung).</p> <p>Es wird ein <code>fme_point</code> aus der Objektcoordinate erzeugt, bzw. soweit keine Objektcoordinate vorliegt ein Objekt ohne Geometrie. Das Feature erhält zwei Attributlisten <code>out_link{}.link</code> und <code>out_link{}.linktyp</code>, die die Linkadresse und den Geometrietyp (Definition oder Ausgestaltung) der referenzierten Einzelelemente enthalten. Die Objektbestandteile erhalten das Attribut <code>out_link_id</code>, auf das die Objektliste verweist.</p> <p>Es wird ein Aggregat mit ausschließlich den im Objekt enthaltenen Linien erzeugt. Aus Linien der Definitionsgeometrie wird eine Fläche erzeugt, Ausgestaltungslinien werden als Einzellinien angehängt. Alle punktförmigen Bestandteile des Objekts (Punkte, Texte) und auch Schraffuren werden in einer Liste <code>out_elements{}</code> gespeichert.</p> <p>Flächenhaft ausgebildete Geografobjekte werden als Polygon gespeichert, eventuell enthaltene Ausgestaltungslinien und alle anderen Elemente gehen in der Liste <code>out_elements{}</code> auf. Linienhafte Geografobjekte werden als Aggregat aus Einzellinien gespeichert.</p> <p>Es wird ein Aggregat mit allen im Objekt enthaltenen Geometrien erzeugt. Soweit die Definitionslinien Flächen bilden, werden diese als <code>fme_polygon</code> bzw. <code>fme_donut</code> im Aggregat abgelegt.</p> <p>In den Varianten LINE, AREA und AGGREGATE werden Definitions- und Ausgestaltungselemente grundsätzlich getrennt behandelt. Die Geometrie erhält den Namen <code>out_definition</code> oder <code>out_decoration</code>. Bei der Variante AGGREGATE wird bei gemischten Geometrien ein IFMEAggregat mit dem Namen <code>out_composite</code> angelegt, welches zwei Geometriebestandteile mit den Namen <code>out_definition</code> oder <code>out_decoration</code> enthält. Alle Geometriebestandteile erhalten die Geograf-Eigenschaften als Geometrieattribute mit den Bezeichnungen, wie sie bei den Einzelelementen dokumentiert sind.</p> <p>Alle Varianten kann auch der Writer verarbeiten. Die Varianten LINK und LIST können speicherintensiv sein, da hier die Bestandteile teilweise zwischengespeichert werden.</p> <p>Value: NO LINK LIST LINE AREA AGGREGATE Default: AGGREGATE</p>	
REMOVE_OBJECT_GEOM	<p>Soweit aus den Objekt-Definitionselementen mit dem OBJECT_PARSER bereits geometrische Features erzeugt werden, ist es oft sinnvoll, die ursprünglichen Einzelteile aus dem Datensatz zu löschen. Soweit OBJECT_PARSER = NO, LINK oder LIST gesetzt ist, setzt der Reader diesen Schalter auf No.</p> <p>Value: Yes No Default: Yes</p>	Optional

Keyword Suffix	Value	Required/Optional
REMOVE_OBJECT_DATA	<p>Soweit aus den Objekt-Ausgestaltungselementen mit dem OBJECT_PARSER bereits geometrische Features bzw. Listeneinträge erzeugt werden, ist es oft sinnvoll, die ursprünglichen Einzelteile aus den Datensatz zu löschen.</p> <p>Soweit OBJECT_PARSER = NO, LINK oder LIST gesetzt ist, setzt der Reader den Schalter auf No.</p> <p>Value: Yes No</p> <p>Default: Yes</p>	Optional
ADD_OBJECT_POINT	<p>Soweit mit dem OBJECT_PARSER Aggregate erzeugt werden, kann die Objektkoordinate als <code>fine_point</code> in das Aggregate übernommen werden. Diese wird dann als Punktgeometrie mit dem Namen <code>out_snap</code> angelegt.</p>	Optional
STANDARD_TEXT_SIZE	<p>Geograf speichert die Textgröße nur im Ausnahmefall. In der Regel ist sie auf 0.0 gesetzt und wird bei der Präsentation aus der Artendatei entnommen. Mit diesem Schalter kann eine Größe vorgegeben werden, die dann für alle Texte benutzt wird. Die ungleich 0.0 gelesene oder die hiermit erzeugte wird in dem Attribut <code>out_text_size</code> abgelegt, der originale Geografwert (auch 0.0) bleibt im Attribut <code>out_size</code> erhalten.</p> <p>Value: Fließkommazahl</p> <p>Default: 2.0</p>	Optional
MULTI_LINE_TEXT	<p>Geograf kann Texte speichern, die aus mehreren Textzeilen bestehen. Wenn der Schalter auf Yes gesetzt ist, werden die Textzeilen zusätzlich in einer Liste <code>out_text_string{}</code> gespeichert.</p> <p>Value: Yes No</p> <p>Default: No</p>	Optional
MULTI_LINE_TEXT_DELMIM	<p>Mehrzeilige Texte werden auch in dem Attribut <code>out_text_string</code> gespeichert. Dabei wird ein Zeilenumbruch '\n' als Trennzeichen verwendet. Das Trennzeichen kann mit dem Schalter <code>MULTI_LINE_TEXT_DELMIM</code> gesondert gesetzt werden.</p> <p>Value: beliebiges ASCII-Zeichen</p> <p>Default: '\n'</p>	Optional
TEXT_PARSER	<p>Mit der Beschriftungsfunktion hält Geograf eine Verbindung des Beschriftungstextes mit dem beschrifteten Element vor. Der Schalter TEXT_PARSER bietet folgende Einlesevarianten an:</p> <ul style="list-style-type: none"> NO Beschriftungen werden nicht besonders gelesen LINK Ein Beschriftungstext erhält das Attribut <code>out_text_link</code>, welches den Verweis zu beschrifteten Element enthält. Dieses ist mit dem Attribut <code>out_text_id</code> gekennzeichnet. Bei Texten, die nicht Beschriftung sind, ist in <code>out_text_link</code> der Wert 0 abgelegt. LIST Die beschrifteten Elemente erhalten eine Liste <code>out_text{}</code>, die alle Attribute der Beschriftungstexte enthält. Die Texte werden als solche erzeugt, ohne dass die Verbindung zum Beschriftungselement gespeichert wird. EXCLUSIVE Wie in der Variante LIST erhält das beschriftete Element eine Liste mit den Textattributen. Die Texte werden dabei nicht als Features erzeugt. <p>Die Varianten LINK und EXCLUSIVE werden auch durch den Writer unterstützt, bei der Variante LIST würden die Texte doppelt abgelegt werden.</p> <p>Value: No LINK LIST EXCLUSIVE</p> <p>Default: LINK</p>	
TEXT_AS_3D	<p>Geograf selbst speichert die Texte zweidimensional. Bei auf Yes gesetzten Schalter übernimmt der Reader für Beschriftungen die Höhe des beschrifteten Objekts. Bei beschrifteten Linien oder Flächen wird die Höhe aus der der Textkoordinate nächstgelegenen Kante interpoliert.</p> <p>Value: Yes No</p> <p>Default: No</p>	

Keyword Suffix	Value	Required/ Optional
READ_UUID	Bei Fortführungsdaten speichert Geograf die UUID zum Element. Bei Bedarf muss die Übernahme der UUID explizit angeschaltet werden. Value: Yes No Default: No	Optional
READ_DIGIT	Geograf hält für die Punktdaten mehrere Berechnungsansätze vor. Bei Bedarf kann das Lesen angeschaltet werden. Die PunktFeatures erhalten dann eine Liste <code>out_dig{}</code> , diese enthält Koordinaten, Verbesserungen und Gewichte der Einzelbestimmungen. Value: Yes No Default: No	Optional
READ_OFFSET	Geograf speichert mit dem Rissmodul Verschiebungsvektoren zu Punkten und Texten, bei Bedarf kann das Lesen angeschalten werden. Value: Yes No Default: No	
READ_MODEL	Die Transformation von Bitmaps und Dokumenten speichert Geograf in Modellen, die als Standard nicht gelesen werden. Das Lesen der Modelle kann angeschalten werden. Value: Yes No Default: No	Optional
READ_DOCUMENT	Das Lesen der Dokumente kann angeschalten werden. Dokumente werden als Polygon mit dem Dokumentnamen gelesen. Value: Yes No Default: Yes	Optional
READ_PK0	Bei GIS-Auswertungen mit linien- und flächenhaften Objekten werden oft viele Geograf-Punkte nur dazu benötigt, um Linien oder Flächen zu definieren. Mit diesem Schalter kann das Lesen der Punkte mit Art=0 ausgeschalten werden, es wird so Rechenzeit und Speicher gespart. Value: Yes No Default: No	Optional
READ_VOID_DATA	Soweit dieser Schalter aus Yes gesetzt ist, werden auch für leere Sachdatenfelder die entsprechenden Attribute angelegt und je nach Sachdatentyp mit 0 oder "" belegt. Value: Yes No Default: No	Optional
MERGE_DATA_ATTRIBUTES	Geograf kann mehrere Sachdatenmasken pro Element speichern. Aus den Geograf-Sachdaten werden entsprechende Attributnamen angelegt, die als Präfix den Namen des Sachdatensatzes erhalten. Damit werden auch gleichlautende Attributnamen aus mehreren Sachdatenmasken kollisionsfrei übernommen. Soweit sichergestellt ist, dass nur ein Sachdatensatz pro Element angelegt ist bzw. eindeutige Attributnamen vergeben sind, kann dieser Schalter auf Yes gesetzt werden. Die Vergabe des Präfixes unterbleibt dann. Value: Yes No Default: No	Optional
DATA_ATTRIBUTE_PREFIX	Die Attributnamen der Sachdaten können ein zusätzliches Präfix erhalten. Value: string Default: ""	Optional
ART_SETTING_FILE	Hiermit wird die zusätzlich zu lesende Artendatei vorgegeben. Die Artendatei muss im ArtBat-Format der Version 7 vorliegen. Die Informationen der Artendatei werden vom Reader an die Features angehängt. Default: ""	Optional
PEN_SETTING_FILE	Hiermit wird die zusätzlich zu lesende Stiftdatei vorgegeben. Die Stiftdatei liegt als reine Textdatei typischerweise mit der Extension pen oder ins vor. Sie enthält die Einstellungen zu der Plotausgabe. Auch diese Informationen werden mit den Features verknüpft. Default: ""	Optional

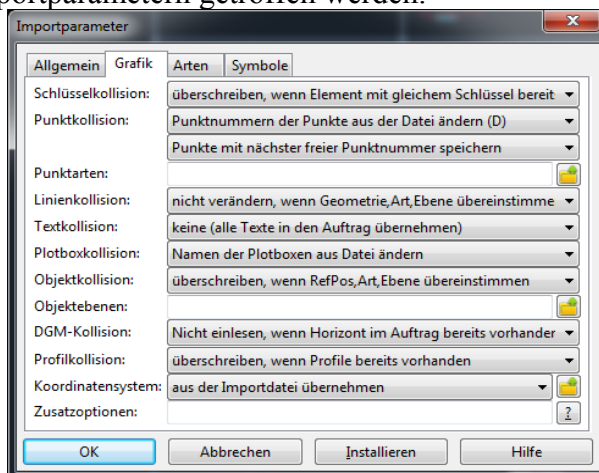
Keyword Suffix	Value	Required/ Optional
LEVEL_SETTING_FILE	Hiermit wird die zusätzlich zu lesende Leveldatei vorgegeben. Die Leveldatei liegt ebenso als Textdatei vor und enthält Informationen zur Darstellungs- bzw. Plotreihenfolge der Features. Sie liegt typischerweise mit der Extension lvl oder ins vor. Default: ""	Optional
COLOR_SETTING_FILE	Hiermit wird die zusätzlich zu lesende Farbdatei vorgegeben. Die binär vorliegende Einstellungsdatei enthält die Farbinformationen zur Bildschirmdarstellung der Features. Default: ""	Optional
ART_KEEP_ATTRIBUTES	Die sehr umfangreichen Attribute der Katalogdateien können mit diesem Parameter auf die notwendigen eingegrenzt werden. Value: string (Komma- bzw. Leerzeichen getrennte Liste der art_-Attribute) Default: ""	Optional

Writer Overview

Die Speicherung von Geometriedaten erfolgt in Geograf im Gegensatz zu vielen anderen Softwarelösungen punktorientiert. So haben z.B. Linien keine eigenen Koordinaten, es wird ein Verweis zum Linienanfangs- und Endpunkt gespeichert.

Komplexe Geometrien werden als Objekte gespeichert. Hierbei wird ebenfalls in einer Liste auf die atomaren Geometrieelemente verwiesen.

Der Geograf-Writer baut diese Verweisstruktur teilweise eigenständig auf. Die endgültige Entscheidung, welche Punkte und Kanten verschmolzen werden, kann mit den Geograf-Importparametern getroffen werden.



Da der Punkt- und Linienabgleich unter Umständen abhängig von den FeatureTypen ist, bietet der Writer die Möglichkeit, Features in verschiedene Ausgabedateien zu schreiben. Diese können dann nacheinander in Geograf eingelesen werden, mit den Kollisionsparametern kann dabei der Abgleich gesondert behandelt werden.

Die **Objektstruktur** kann aufgebaut werden, in dem die zu schreibenden Features als komplexe Aggregate erzeugt werden. Hierbei können jedem Objektbestandteil individuelle Attribute zugewiesen werden. Es können aber auch Objekte darüber generiert werden, dass ihnen Objektverweise mit Schlüsselattributen zugewiesen werden. Hierzu später mehr im Kapitel **Objekte**.

Ebenso ermöglicht der Writer die Generierung von Beschriftungen, so dass die Beziehung zwischen Text und beschrifteten Element erhalten bleibt. Hierzu mehr im Kapitel **Beschriftungen**.

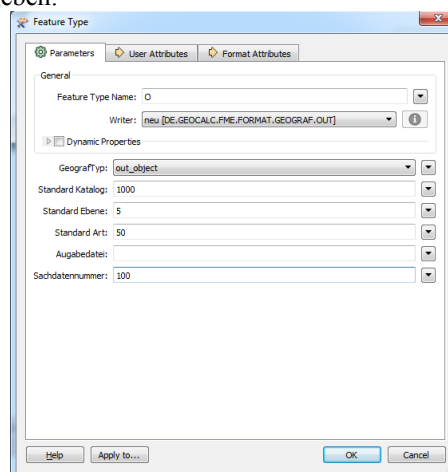
Der Writer ermöglicht auch das Schreiben von individuellen Attributen, die Geograf in Sachdatensätzen verwaltet. Geograf ermöglicht das Anlegen von mehreren Sachdatensätzen pro Feature. Auch diese können erzeugt werden, hierzu später mehr im Kapitel **Sachdaten**.

Der Writer schreibt die Grafbat-Datei sequentiell, es sollte damit eine praktisch unbegrenzte Dateigröße geschrieben werden können. Lediglich der Aufbau der Objekt- und Beschriftungsstrukturen bedingt, dass Features teilweise zwischengespeichert werden. Dazu mehr in beiden Kapiteln.

Der Writer erzeugt Koordinaten generell mit 4 Nachkommastellen, diese Stellenanzahl wird von Geograf auch für die automatische Punktverschmelzung beim Import benutzt.

Writer Keywords

Keyword Suffix	Value	Required/Optional
DATASET	Hiermit wird die zu schreibende Ausgabedatei vorgegeben.	Required Optional
ATTDEFS_OVERWRITE	Mit dem Schalter kann gesteuert werden, ob die formatspezifischen Attribute der DEF-Zeile eventuell vorhandene FeatureAttribute überschreiben. Value: Yes No Default: Yes	Optional
MERGE_COORDS	Der Writer erzeugt automatisch Geograf-Punkte an den Vertex von Linien und Flächen. Die Punktart kann dabei mit den Attributen <code>out_point_art</code> , <code>out_point_ebene</code> , <code>out_point_katalog</code> gesetzt werden. Es „gewinnt“ der erste geschriebene Punkt, später geschriebene Vertex verweisen auf diesen. Wenn je Linien- bzw. Objektart gesonderte Punktarten vergeben werden sollen, muss der Schalter auf 'No' gesetzt werden, dann entsteht aus jedem Feature ein individueller Punkt, sofern er gesondert attribuiert wird. Value: Yes No Default: Yes	
DEF	Für jeden FeatureType können die Inhalte der formatspezifischen Attribute vorgegeben werden. Diese werden in der DEF-Zeile aufgelistet. Beispiel: <pre>OUT_DEF Flurstueck \ out_type out_object \ out_art 100 \ out_ebene 1</pre> Für alle Attribute, die nicht mit dem Formatkennzeichen <code>out_</code> beginnen, wird ein Sachdatensatz erzeugt. Das Sachdatum erhält den Namen des FeatureTypes. Die Sachdatennummer wird automatisch generiert, soweit sie nicht mit dem Attribut <code>out_data_id</code> vorgegeben wird. Geograf ermöglicht eine Vergabe in dem Bereich von 1-9999. In der Workbench werden die Featureattribute über den Reiter Format Parameters vergeben:	Optional



Folgende Attributtypen sind für die User-Attribute vorgesehen:

- boolean** entspricht dem GeografAttributtyp `b=boolescher Wert`
- number(w,0)** entspricht dem GeografAttributtyp `d=Ganzzahl`
- number(w,p)** entspricht dem GeografAttributtyp `f=Gleitkommazahl`
- buffer** entspricht dem GeografAttributtyp `s=alphanumerisch (ohne Längenbegrenzung, Memofeld)`
- string** entspricht dem GeografAttributtyp `s=alphanumerisch (ohne Längenbegrenzung, Memofeld)`
- char(w)** entspricht dem GeografAttributtyp `s=alphanumerisch`

Keyword Suffix	Value	Required/Optional
----------------	-------	-------------------

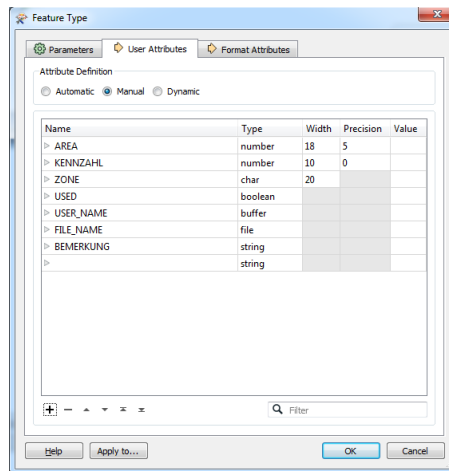
date (mit Längenbegrenzung)
entspricht dem GeografAttributtyp s=alphanumerisch
(10 Zeichen Längenbegrenzung)

file entspricht dem GeografAttributtyp n=alphanumerisch
(ohne Längenbegrenzung)

Beispiel:

```
OUT_DEF Flurstueck \
out_type out_object
out_art 100 \
out_ebene 1 \
out_data_id 1
Flur number(5,0)\
Nummer string \
```

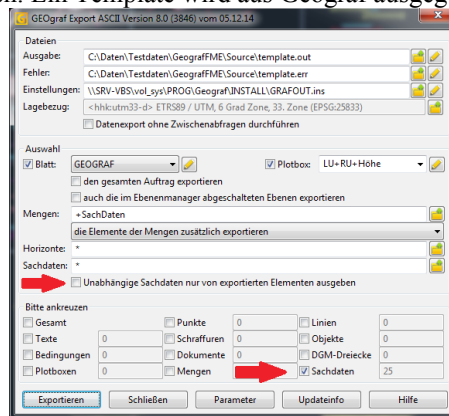
In der Workbench werden die Userattribute über den Reiter User Attributes vergeben:



Weitere Möglichkeiten der Erzeugung von Sachdaten werden im Kapitel **Sachdaten** beschrieben.

TEMPLATE_FILE

Es können dem Writer unter Geograf bereits definierte Sachdaten vorgegeben werden. Dann kann auf die Definition der User-Attribute verzichtet werden. Ein Template wird aus Geograf ausgegeben:



Default: null

APPEND_TEMPLATE_DATA

Bei dem Schreiben von Subtabellen und Picklisten sucht der Writer ein passenden Sachdatensatz aus einem vorgegebenen Template, wird ein solcher nicht gefunden, wird eine Fehlermeldung protokolliert. Beim Setzen dieses Schalters werden die Verweistabellen mit fehlenden Datenzeilen aufgefüllt.

Value: Yes | No

Default: No

Feature Representation

Alle Features

Attribute Name	Content
out_type	Der Geograf-Typ des Elements Value: out point, out line, ...
out_key	Eindeutiger Elementschlüssel, diese ist je OUT-Datei eindeutig. (nur Reader) Value: Ganzzahlwert Default: No default
out_extkey	Der externe Schlüssel des Elements Value: char(30) Default: No default
out_uuid	Beim Export aus Geograf können UUIDs erzeugt werden. Diese werden gelesen, wenn der Schalter READ_UUID gesetzt ist. Value: boolean Default: No
out_uuid_flag	Beim Export aus Geograf können UUIDs erzeugt werden. Dabei wird auch ein Flag gesetzt, dass anzeigt, ob Veränderungen am Objekt vorgenommen wurden. Dieses Flag wird unabhängig vom Schalter READ_UUID gelesen. Geograf setzt in der aktuellen Version das Flag nur, wenn die Struktur oder die Attribute des Features geändert werden (zB. Linie teilen, Art ändern). Das Verschieben von Punkten samt der anhängenden Linien bewirkt noch kein Setzen des Flags. Value: boolean Default: No
out_katalog	Der Katalog, auf der das Element liegt. Range: 0-9999 Default: 0
out_ebene	Die Ebene, auf der das Element liegt. Range: 0-9999 Default: 0
out_art	Die Art des Elements. Range: 0-9999 Default: 0
out_link_id out_object{}.link out_object{}.linktyp	Im Modus OBJECT_PARSER = LINK setzt der Reader beim Objekt das Attribut out_link_id , bei den Objektelementen die Listenattribute out_object{}.link und out_object{}.linktyp. Diese Verweisstruktur wird auch vom Writer ausgewertet, die Objekte aus den Objektteilen zusammengesetzt. Value: Integer für id und link Value: Integer für linktyp (1=Ausgestaltung, 2=Definition)
out_link_id out_link{}.link out_link{}.linktyp	Im Modus OBJECT_PARSER = LIST setzt der Reader beim Objekt die Listenattribute out_link{}.link und out_link{}.linktyp, bei den Objektbestandteilen das Attribut out_link_id. Auch diese Verweisstruktur wird vom Writer ausgewertet. Value: Integer für id und link Value: Integer für linktyp (1=Ausgestaltung, 2=Definition)
out_elements{} out_elements{}.geometry	Im Modus OBJECT_PARSER = LINE oder AREA speichert der Reader alle Elemente, die nicht in der Featuregeometrie aufgegangen sind, als Liste. Die Geometrie der Elemente wird im Attribut out_elements{}.geometry als OGC-WKT gespeichert. Auch diese Listen wertet der Writer aus.
out_text_id out_text_link	Im Modus TEXT_PARSER = LINK erhalten beschriftete Elemente das Attribut out_text_id, die Beschriftungstexte verweisen auf die ID mit dem Attribut out_text_link. Der Writer wertet diese Verlinkung aus. Value: Integer für id und link
out_text{}	Im Modus TEXT_PARSER = LIST und EXCLUSIVE erhalten beschriftete Elemente eine Liste mit den Beschriftungstexten.

Attribute Name	Content
out_data{}	In dem Listenattribut werden die Namen der zum Geograf-Element gespeicherten Sachdatensätze gespeichert. Soweit Sachdaten mit Hilfe eines Geograf-Templates geschrieben werden soll, müssen Features mit dieser Liste anzeigen, welche Sachdatensätze generiert werden sollen. Value: char(100) Default: No default
out_file	Mit dem Attribut können Features in verschiedene Grafbat-Dateien umgeleitet werden. Wenn keine Pfadangebe enthalten ist, erzeugt der Writer die Datei im Verzeichnis des DATASET-Parameters. (nur Writer)
out_comment	Mit dem Attribut können Kommentarzeilen zu den einzelnen Features in die auszugebende Grafbat-Dateien geschrieben werden. (nur Writer)
out_point_art out_point_ebene out_point_katalog out_line_art out_line_ebene out_line_katalog out_text_art out_text_ebene out_text_katalog	Beim Erzeugen von Objekten aus Aggregaten erzeugt der Writer die Attribute der Bestandteile vorrangig aus den Geometrieattributen. Wenn diese nicht gesetzt sind, greift er auf diese Attribute zurück, soweit sie in dem Aggregat-Feature gesetzt sind. Das setzen der Punktarten kann auch beim Schreiben von Linien angewendet werden. (nur Writer)

Punkte

out_type: out_point

Attribute Name	Content
out_number	Nummer des Punktes Value: char(15) Default: No default. Soweit kein Wert vorgegeben wird, erzeugt Geograf selbständig eine eindeutige Punktnummer.
out_rotation	Drehwinkel des Punktes in Grad entsprechend den FME-Konventionen Range: 0.0 – 360.0 Default: 0.0
out_size	Symbolgröße des Punktes. Im Regelfall ist diese auf 0.0 gesetzt. Range: Fließkommawert Default: 0.0
out_l_stat out_h_stat	Lagegenauigkeit Höhen Genauigkeit 0 = ungültig 1 = digitalisiert 2 = gerechnet 3 = fest Range: 0-3 Default: 2
out_date	Datum der Berechnung (nach FME-Datumskonventionen) Range: date Default: No default
out_symbol{}.oska out_symbol{}.rotation out_symbol{}.size	Mehrfachsymbole werden in dieser Liste abgelegt. Das Attribut oska beinhaltet Katalog, Ebene und Art, mit Punkt getrennt.
Attribute bei READ_OFFSET = Yes	
out_x_off out_y_off out_z_off	Verschiebevektor für die Rissansicht Range: Fließkommawert Default: 0.0
Attribute bei READ_DIGIT = Yes	
out_ansatz	Die Ansatznummer der Koordinatenberechnung Range: 0-999999999 Default: 0
out_dig_trafo	Transformationspunkt Range: 0-1 Default: 0
out_dig_dot	digitalisierter Punkt Range: 0-1 Default: 0
out_dig_aus	ungültig für Transformation Range: 0-1 Default: 0
out_dig_rand	Randpunkt (dieser Schalter wird jetzt offensichtlich dazu benutzt, um die Verzerrung einzuschalten) Range: 0-1 Default: 0
out_dig_soll	Passpunkt aus ehemals digitalisiertem Punkt Range: 0-1 Default: 0
out_dig_gew	Gewicht der digitalisierten Koordinaten Range: 0-9999 Default: 0
out_l_gew out_h_gew	Lage-, Höhengewicht Range: 0-9999 Default: 0

Attribute Name	Content
out_l_ctrl out_h_ctrl	Lage-, Höhenkontrolle Range: 0-9999 Default: 0
out_dig{}.ansatz out_dig{}.y out_dig{}.x out_dig{}.z out_dig{}.y_ver out_dig{}.x_ver out_dig{}.z_ver out_dig{}.l_gew out_dig{}.h_gew out_dig{}.l_stat out_dig{}.h_stat out_dig{}.l_ctrl out_dig{}.h_ctrl out_dig{}.dig_trafo out_dig{}.dig_aus out_dig{}.dig_gew	Die Liste enthält alle Werte eines Berechnungsansatzes.

Linien

out_type: out_line
out_type: out_polyline
out_type: out_arc
out_type: out_circle
out_type: out_clothoid

Eine Geograf-Klothoide wird immer als Bogenpolygon gespeichert, Bögen und Kreise abhängig vom Schalter ARC TO POLYLINE-

Attribute Name	Content
out_pattern_side_change	Linienmuster wird gespiegelt. 0 = nein 1 = ja Range: 0-1 Default: 0
out_pattern_start_change	Linienmuster umdrehen. 0 = nein 1 = ja Range: 0-1 Default: 0
out_size	Linienbreite in Koordinateneinheiten Range: Fließkommawert Default: 0.0
out_pattern_dist	Startwert des Linienmusters in Koordinateneinheiten Range: Fließkommawert Default: 0.0
out_color	Linienfarbe entsprechend den FME-Konventionen Range: 0.0-1.0, 0.0-1.0, 0.0-1.0 Default: <null>
out_number1	Nummer des ersten Linienpunktes (nur Reader) Value: char(15)
out_number2	Nummer des zweiten Linienpunktes (nur Reader) Value: char(15)
out_rotation out_primary_axis out_secondary_axis out_start_angle out_sweep_angle out_clothoid_param	Die Geometrieattribute von Bögen, Kreisen und Klothoiden. Default: 0.0

Splines

`out_type`: `out_spline`

Geograf-Splines werden als `fine_line` übertragen.

Für die Typen 1-4 kann eine für den gesamten Spline geltende Höhe angegeben werden, der Reader übernimmt diese in jeden Vertex. Bei den Typen 6-9 hat jeder Vertex einen individuellen z-Wert.

Bei linearen Splines werden alle Stützpunktkoordinaten übernommen. Der Schalter `SPLINE_SEGMENT_POINTS` hat auf diese Linienzüge keine Auswirkung. Alle linearen Splines erhalten das Geometrieattribut `out_spline_info = linear`.

Bei kubischen Splines wird der Linienzug aus allen Stütz- und Kontrollpunkten erzeugt, wenn der Schalter `SPLINE_PHANTOM_POINTS` auf `NO` gesetzt ist. Die kubischen Splines erhalten dann das Geometrieattribut `out_spline_info = controls`.

Ist der Schalter `SPLINE_PHANTOM_POINTS` auf `YES` gesetzt, wird der Linienzug vorerst nur aus den signifikanten Knickpunkten ($n=0, 3, 6, 9, \dots$) erzeugt. Das Geometrieattribut ist dann `out_spline_info = 0 segments`.

Ist zusätzlich `SPLINE_SEGMENT_POINTS > 0` gesetzt, werden weitere Zwischenpunkte interpoliert. Eine größere Anzahl von Zwischenpunkten ergibt einen weicheren Linienzug. Ab einem Wert von etwa 10 Segmentpunkten entspricht die Darstellung der Geograf-Grafik. Die kubischen Splines werden dann mit `out_spline_info = <n> segments` gekennzeichnet.

Soweit unter Geograf Splines durch Umwandlung vorhandener Linienzüge erzeugt werden, speichert Geograf die ursprüngliche Geometrie, stellt diese aber nicht dar. Zusätzlich zum eigentlichen Spline speichert Geograf auch die Ausgangsgeometrie in der Out-Datei und setzt Referenzen. Der Reader übernimmt die Ausgangsgeometrie nicht, damit gehen diese Informationen bei einer Geograf-Geograf-Übertragung verloren.

Wenn bei einer Transformation von Geograf nach Geograf die Geometrien der Splines nicht verändert werden, so sollte der Schalter `SPLINE_PHANTOM_POINTS` auf `NO` belassen werden. So werden die Splines originalgetreu wiederhergestellt. Wenn die Linienzüge geometrisch verändert werden sollen, ist die Wandlung in Linienzüge sinnvoll, auch hier reicht i.d.R. die Aufteilung in 10 Segmente. Der Reader speichert den ursprünglichen Splinetyp als Geometrieattribut `out_spline_info`, so stellt der Writer die Splines auch wieder her. Da hier aber die die Ausrundungen bestimmenden Stützpunkte nicht mehr vorliegen, gelingt dies nicht originalgetreu.

Wenn die Splineerzeugung beim Writer gesteuert werden soll, kann `out_spline_info` auch als Featureattribut gesetzt werden, es überschreibt ein eventuell vorhandenes Geometrieattribut. Zusätzlich zu den obig beschriebenen Varianten kann `out_spline_info` auf `'generate'` oder `'detect'` gesetzt werden.

Bei `'generate'` werden alle Vertex übertragen, beim Einlesen generiert Geograf daraus einen durch diese Stützpunkte gerundeten Spline. Dies eignet sich besonders für kantige Linienzüge, die ausgerundet werden sollen.

Liegt ein bereits feingliedrig ausgerundeter Linienzug vor, der als Spline übertragen werden soll, hilft der Modus `'detect'`. Der Writer ermittelt aus dem Linienzug die markanten Stützpunkte und erzeugt die Kontrollpunkte.

Generierte Splines werden nur 2D oder mit einheitlicher Höhe übertragen.

Attribute Name	Content
out_rotation	Werte ohne Bedeutung
out_size	
out_typ	<p>Splinetyp (nur Reader)</p> <p>1 = linear offen 2 = linear geschlossen 3 = kubisch offen 4 = kubisch geschlossen 6 = linear offen, 3D (kann Geograf im Moment nicht erzeugen) 7 = linear geschlossen, 3D 8 = kubisch offen. 3D 9 = kubisch geschlossen, 3D</p> <p>Range: 1-9 Default: No default</p>
out_spline_info	<p>Art der Linienzugerzeugung</p> <p>linear = aus linearen Spline controls = aus kubischen Spline mittels Kontrollpunkten <n> segments = aus kubischen Spline mit <n> Segmenten generate = Spline soll aus Linienpunkten erzeugt werden (nur Writer) detect = Linienzug wird analysiert und die Stütz - und Kontrollpunkte erzeugt (nur Writer)</p> <p>Range: String Default: No default</p>

Texte

out_type: out_text

out_type: out_multi_text

Multitexte werden nur erzeugt, wenn der Schalter MULTI LINE TEXT auf Yes gestellt ist.

Attribute Name	Content
out_rotation	Drehwinkel des Textes in Grad entsprechend den FME-Konventionen Range: Fließkommawert Default: 0.0
out_size	Größe des Textes in Koordinateneinheiten. Im Regelfall ist dieser auf 0.0 gesetzt. Die Größe entnimmt dann Geograf der Artendatei. Der Reader belässt diesen Wert bei 0.0, wenn keine individuelle Größe vergeben ist. Der Writer benutzt den hier vergebenen Wert, auch 0.0. Nur wenn out_size nicht vergeben ist, wird auf out_text_size zurückgegriffen. Range: Fließkommawert Default: 0.0
out_text_size	Größe des Textes in Koordinateneinheiten. Wenn mit Geograf keine individuelle Größe festgelegt ist, wird der Wert mit STANDARD_TEXT_SIZE belegt. Range: Fließkommawert Default: STANDARD_TEXT_SIZE
out_text_string	Inhalt des Textes, mehrzeilige Texte erhalten als Trennzeichen MULTI_LINE_TEXT_DELIM Value: string
out_text_string{}	Inhalt der Textzeilen bei Multitexten wird vom Reader nur bei MULTI_LINE_TEXT=Yes erzeugt Value: string
out_text_formel	Textformel bei Beschriftungstexten, mehrzeilige Texte erhalten als Trennzeichen MULTI_LINE_TEXT_DELIM Value: string
out_text_formel{}	Textformel bei Multitexten wird vom Reader MULTI_LINE_TEXT=Yes erzeugt Value: string
out_rahmen	Besondere Hervorhebung durch Textrahmen 0 = ohne Rahmen Bezug vom Zentrum 1 = Kasten Bezug von den Ecken 2 = Kreis Bezug vom Zentrum 3 = einfach unterstrichen Bezug vom Unterstrich 4 = doppelt unterstrichen, eng Bezug vom Unterstrich 5 = durchgestrichen Bezug vom Durchstrich 6 = ohne Rahmen Bezug von den Ecken 7 = doppelt unterstrichen, breit Bezug vom oberen Unterstrich 8 = kleiner Kreis Bezug vom Zentrum Range: 0-8 Default: 0
out_frei	Freistellart 0 = Kasten 1 = ohne 2 = Kreis Range: 0-2 Default: 0

Attribute Name	Content
out_position	<p>Position, auf die sich die Textkoordinate bezieht</p> <p>0 = links-unten 1 = mitte-unten 2 = rechts-unten 3 = links-mitte 4 = mitte-mitte 5 = rechts-mitte 6 = links-oben 7 = mitte-oben 8 = rechts-oben</p> <p>Hinweis: Der FME-DataInspector zeigt den Text immer rechts-oberhalb der Textkoordinaten an. Die Einfügeposition kann in Fremdsysteme übersetzt werden (z.B. ESRI, GeoMedia), dort wird der Text dann wie unter Geograf positioniert. Siehe: https://knowledge.safe.com/articles/16546/text-justification.html</p> <p>Range: 0-8 Default: 0</p>
out_x_bez out_y_bez out_bez{}.x out_bez{}.y	<p>Koordinaten des Bezugspunktes. Geograf zeichnet einen Bezugsstrich von der Textkoordinate zum Bezugspunkt. Ab Version 8 können zusätzliche Bezugsstriche gespeichert werden. Wenn weitere vorhanden sind, werden diese in den Listenattributen abgelegt.</p> <p>out_x_bez und out_y_bez sind immer belegt, wenn in Geograf kein Bezugsstrich gesetzt ist, findet sich eine Kopie der Textkoordinate. Dem Writer müssen die Werte nicht angegeben werden, es wird dann die Textkoordinate eingesetzt.</p> <p>Range: Fließkommawert Default: 0.0</p>
Attribute bei TEXT_PARSER != No	
out_rel_flag	<p>Flags für relative Texte</p> <p>Range: 0-9999 Default: 0</p>
out_x_rel out_y_rel	<p>Koordinatenoffset der Beschriftung</p> <p>Range: Fließkommawert Default: 0.0</p>
out_r_rel	<p>Winkeloffset der Beschriftung</p> <p>Range: Fließkommawert Default: 0.0</p>
Attribute bei READ_OFFSET = Yes	
out_x_off out_y_off	<p>Verschiebevektor für die Rissansicht</p> <p>Range: Fließkommawert Default: 0.0</p>
out_r_off	<p>Winkeloffset für die Rissansicht</p> <p>Range: Fließkommawert Default: 0.0</p>
Attribute bei READ_DIGIT = Yes	
out_dig_flag	<p>Digitalisierflags</p> <p>Range: 0-9 Default: 0</p>

Schraffuren

out_type: out_schraffur

Aus Geograf-Schraffuren wird eine Geometrie des Typs fme_polygon erzeugt.

Schraffuren können mit neueren Geograf-Versionen auch 3-dimensional erzeugt werden.

Attribute Name	Content
out_rotation	<p>Drehwinkel der Schraffur in Grad entsprechend den FME-Konventionen</p> <p>Range: 0.0-360.0 Grad Default: 0.0</p>

Attribute Name	Content
out_size	Schraffurabstand bei linienhaften Schraffurn in mm Ploteinheiten Range: Fließkommawert Default: 0.0
out_sub_typ	Spezifizierung des Schraffurtyps 0 = normale Schraffur (Linien) 1 = Kreuzschraffur 2 = Rasterschraffur 3 = Flächenschraffur 4 = schraffierte Abstandsfläche 5 = Saumschraffur 6 = deckende Rasterschraffur
out_color	Schraffurfarbe entsprechend den FME-Konventionen Range: 0.0-1.0, 0.0-1.0, 0.0-1.0 Default: <null>

Böschungen

out_type: out_boeschung

Aus Geograf-Böschungen werden Aggregate mit den Böschungslinien erzeugt.

Böschungen können mit neueren Geograf-Versionen auch 3-dimensional erzeugt werden.

Attribute Name	Content
out_rotation	Werte ohne Bedeutung
out_size	
out_color	Schraffurfarbe entsprechend den FME-Konventionen Range: 0.0-1.0, 0.0-1.0, 0.0-1.0 Default: <null>

Objekte

out_type: out_object

Attribute Name	Content
out_name	Objektname Range: char(80) Default: Void
out_object_typ	Objekttyp 0 = Normales Objekt 1 = Splineobjekt 2 = Beschriftung Range: 0-2 Default: 0
out_date	Datum der Objekterzeugung (nach FME-Datumskonventionen) Default: <null>
out_x out_y	Objektkoordinaten Range: Fließkommawert Default: 0.0
out_ansatz	Ansatznummer des Objektes
out_l_stat	Lagegenauigkeit 0 = ungültig 1 = digitalisiert 2 = gerechnet 3 = fest Range: 0-3 Default: 2
out_oberobjekt	Elementschlüssel des Oberobjekts Default: 0
out_sch_katalog out_sch_ebene out_sch_art out_sch_sub_typ out_sch_size out_sch_rotation out_sch_color	Wenn eine Objektschraffur gesetzt ist, werden die Schraffurwerte in diesen Attributen gespeichert.
Attribute bei READ OFFSET = Yes	
out_x_off out_y_off	Verschiebevektor in Koordinateneinheiten Range: Fließkommawert Default: 0.0

Modelle

out_type: out_model

Geograf-Modelle speichern Transformationsparameter, aus ihnen werden Geometrien des Typs `fine_point` mit den Koordinaten `out_new_x` und `out_new_y` erzeugt (nur Reader)

Attribute Name	Content
<code>out_system</code>	Systemnummer
<code>out_old_x</code>	Rechtswert des Schwerpunktes im alten System Range: Fließkommawert
<code>out_old_y</code>	Hochwert des Schwerpunktes im alten System Range: Fließkommawert
<code>out_new_x</code>	Rechtswert des Schwerpunktes im neuen System Range: Fließkommawert
<code>out_new_y</code>	Hochwert des Schwerpunktes im neuen System Range: Fließkommawert
<code>out_par_a</code> <code>out_par_b</code> <code>out_par_c</code> <code>out_par_d</code>	Transformationsparameter Range: Fließkommawert
<code>out_ansatz</code>	Ansatznummer Range: 0-9999
<code>out_par</code>	Anzahl der Transformationsparameter Range: 4 oder 6
<code>out_name</code>	Systemname Range: string

Dokumente

out_type: out_doc

Aus Geograf-Dokumenten werden Polygonfeatures erzeugt (nur Reader)

Attribute Name	Content
<code>out_name</code>	Name des Dokuments Range: char(255)
<code>out_typ</code>	Typ des Dokuments: 0 = Bitmap 1 = DWG-Datei 2 = COM-Objekt Default: 0
<code>out_pen</code>	Stift für 1Bit-Bilder Range: 0-9999
<code>out_trans</code>	Transparente Darstellung Range: 0-9
<code>out_stufe</code>	Hierarchiestufe der Darstellung Range: 0-9999
<code>out_system</code>	Nummer des Digitalisiersystems Range: 0-9999
<code>out_colorkey</code>	Farbnummer für 1Bit-Bilder Range: 0-9999

DGM

out_type: out_dgm

Aus Geograf-DGM-Modellen werden Geometrien des Typs Mesh erzeugt (nur Reader)

Attribute Name	Content
out_horizont	Horizontnummer Range: 0-9999
out_name	Name des Horizonts Range: char(255)

Katalogdaten

Soweit beim Lesen der Geometriedaten können die Geograf-Informationen zur Darstellung an die Features gehängt werden. Dazu müssen zu den Parametern:

```
ART_SETTING_FILE  
PEN_SETTING_FILE  
LEVEL_SETTING_FILE  
COLOR_SETTING_FILE
```

die entsprechenden Katalogdateien ausgewählt werden. Der Reader übernimmt die ausgewählten Informationen und fügt sie den Geometriedaten zu. Alle Attributnamen aus den Katalogdateien haben den Präfix: `art_`.

Soweit mit dem Parameter: `ART_KEEP_ATTRIBUTES` nur einzelne Attribute ausgewählt werden, so werden auch nur diese für die Features übernommen.

Die Angabe der einzelnen Katalogdateien ist optional, damit die Daten aus der Stift-, Level- oder Colordatei zugeordnet werden können, ist allerdings die Angabe der Artendatei notwendig.

Die Stiftdatei `PEN_SETTING_FILE` erzeugt die folgenden Attribute:

```
art_*_pen_size  
art_*_pen_color  
art_*_pen_mode
```

Die Stiftdatei `LEVEL_SETTING_FILE` erzeugt das Attribut:

```
art_level
```

Die Colordatei `COLOR_SETTING_FILE` erzeugt das Attribut:

```
art_*_scr_color
```


Alle Features

Attribute Name	Content
art_name	Name Value: char(55)
art_level	Level Range: 0-9999
art_layer	Nummer der Ebene Range: 0-9999
art_layer_name	Name der Ebene Value: char(55)
art_layer_pen	Stiftnummer der Ebene Range: 0-9999
art_layer_pen_size	Stärke des Ebenenstiftes Value: Fließkommawert
art_layer_pen_color	Farbe des Ebenenstiftes Value: String (RGB-Farbwerte)
art_layer_pen_mode	Stiftmodus des Abschluss einer Linie 0 = rund 1 = eckig
art_layer_color	Farbnummer der Ebene (Bildschirmfarbe) Range: 0-9999
art_layer_scr_color	Farbe der Ebene (Bildschirmfarbe) Value: String (RGB-Farbwerte)
art_group	Nummer der Ebenengruppe Range: 0-9999
art_group_name	Name der Ebenengruppe Value: char(55)
art_color	Farbnummer der Art (Bildschirmfarbe) Range: 0-9999
art_scr_color	Farbe der Ebene (Bildschirmfarbe) Value: String (RGB-Farbwerte)
art_pen	Stiftnummer der Art Range: 0-9999
art_pen_size	Stärke des Stiftes Value: Fließkommawert
art_pen_color	Farbe des Stiftes Value: String (RGB-Farbwerte)
art_pen_mode	Stiftmodus des Abschluss einer Linie 0 = rund 1 = eckig
art_symbol	Symbolnummer der Art Range: 0-9999
art_size	Größe der Art Value: Fließkommawert
art_rotation	Drehwinkel der Art Value: Fließkommawert
art_absolute	Drehwinkel ist absolut Value: boolean

Punkte

Attribute Name	Content
art_mode	Zeichnungsmodus 0 = Schraffur-Deckend 1 = Schraffur- und Linien-Deckend 2 = VollDeckend 3 = VollTransparent

Linien

Geograf ermöglicht die Vereinbarung von mehreren Strichformen je Linie. Als Standard übernimmt der Reader nur die Linienform mit der stärksten Strichstärke. Wenn in dem Parameter `ART_KEEP_ATTRIBUTES` das Listenattribut `art_form{}` explizit angegeben ist, werden alle zur Linienart vereinbarten Linienformen einschließlich der Stiftparameter übernommen.

Attribute Name	Content
art_mode	Zeichnungsmodus 0 = deckend 1 = transparent
art_displace	Verdrängung 0 = Linien mit gleicher Geometrie und kleinerer Stufe nicht verdrängen 1 = Linien mit gleicher Geometrie und kleinerer Stufe verdrängen
art_form art_form{}.form	Formnummer Range: 0-9999
art_size art_form{}.size	Abstand der Form von der Grundlinie Value: Fließkommawert
art_pen art_form{}.pen	Stiftnummer der Form Range: 0-9999
art_pen_size art_form{}.pen_size	Stärke des FormStiftes Value: Fließkommawert
art_pen_color art_form{}.pen_color	Farbe des FormStiftes Value: String (RGB-Farbwerte)
art_pen_mode art_form{}.pen_mode	Stiftmodus des Abschluss der Linienform 0 = rund 1 = eckig

Texte

Attribute Name	Content
art_slant	Textneigung in Grad Value: Fließkommawert
art_factor	Verhältnis Breite/Höhe in Prozent Value: Fließkommawert
art_position	Position, auf die sich die Textkoordinate bezieht 0 = links-unten 1 = mitte-unten 2 = rechts-unten 3 = links-mitte 4 = mitte-mitte 5 = rechts-mitte 6 = links-oben 7 = mitte-oben 8 = rechts-oben Range: 0-8 Default: 0
art_font	Name des Fonts Value: char(40)
art_arrow_size	Pfeilgröße für Bezug in mm Value: Fließkommawert
art_pos_absolute	Schwerpunkt ist fest Value: boolean
art_mode	Zeichenmodus 0 = SchraffurDeckend 1 = Schraffur- und LinienDeckend 2 = VollDeckend 3 = VollTransparent Range: 0-3 Default: 0
art_bold	TrueTypeFont fett Value: boolean
art_border	Linienart für den Textrahmen Range: 0-9999
art_line	Linienart für Bezugsstrich Range: 0-9999
art_area	Schraffurart für Texthintergrund Range: 0-9999

Schraffuren

Attribute Name	Content
art_mode	Schraffurtyp der Fläche 0 = Normale Schraffur 1 = Kreuzschraffur 2 = Rasterschraffur 3 = Flächenfüllung bzw. Saum 6 = deckende Rasterschraffur Default: 0
art_line	Linienart bzw. Punktart, je nach Schraffurtyp Range: 0-9999
art_transparent	Transparenz Value: boolean

Objekte

Attribute Name	Content
art_object	Nummer der Oberobjektart Range: 0-9999
art_point	Nummer der Standard-Punktart Range: 0-9999
art_line	Nummer der Standard-Linienart Range: 0-9999
art_text	Nummer der Standard-Textart Range: 0-9999
art_add_point	Nummer der Zusatz-Punktart Range: 0-9999
art_add_point_layer	Nummer der Ebene für Zusatzpunkte Range: 0-9999
art_add_text	Nummer der Zusatz-Textart Range: 0-9999
art_add_text_layer	Nummer der Ebene für Zusatztexte Range: 0-9999
art_mode	Zeichnungsmodus 0 = deckend 1 = transparent
art_object_name	Speicherung des Objektnamens Value: boolean
art_praefix	Präfix für Objektnamen Value: char(12)
art_area	Nummer der Anzeige-Schraffurart Range: 0-9999
art_area_d	Nummer der Standard-Schraffurart Range: 0-9999

Geometrie

Die grundlegenden Geometrieelemente unter Geograf sind Punkte, Linien und Texte. Weitere spezielle Geometrieelemente sind Splines, Flächen- und Böschungsschraffuren. Objekte sind Aggregate aus den grundlegenden Geometrieelementen, gespeichert werden neben einer Objektkoordinate lediglich Verweise zu den Bestandteilen.

Digitale Geländemodelle werden als Dreiecksnetze gespeichert.

Bilddaten und andere Dokumente werden als Dateiverweis gespeichert, Geograf speichert hierzu die Georeferenzierung und den Berechnungsansatz dazu.

In der Standardeinstellung erzeugt der Reader die folgenden **Featuretypes**, die aus der Geograf-Geometrie abgeleitet werden.

P:

`out_point`

Da Geograf aus dem Vermessungswesen stammt, ist die grundlegende Geometrie der vermessene Punkt. Zu einem Punkt können mehrere Berechnungsansätze mit verschiedenen Ergebniskoordinaten vorliegen, diese werden gelesen und in einer Liste gespeichert, wenn `READ_DIGIT` gesetzt ist.

L:

`out_line`

`out_polyline`

`out_arc`

`out_circle`

`out_clothoid`

`out_spline`

Linien werden als gerade oder bogenförmige Verbindung zwischen Punkten aufgefasst, so ist es auch datentechnisch modelliert. Erst in den letzten Versionen ist es möglich, eine Linie mit Zwischenpunkten zu versehen, die nicht zwingend einen Bezug zu einem konkret vorliegenden Punkt haben.

Gerade Verbindungen zwischen zwei Punkten erzeugt der Reader mit dem Geometrieattribut `out_type = out_line`, Linien die weitere Zwischenpunkte aufweisen, werden mit `out_polyline` markiert.

Vollkreise und Kreisbögen werden in der Standardeinstellung des Readers als solche gespeichert, die Bogenparameter werden so wie unter Geograf abgelegt, auch die 3-Punkt-Variante ist möglich.

Die Klothoide ist eine Geograf-Geometrie, für die es keine Entsprechung in der FME gibt. Die Klothoide wird immer als Bogenpolygon ausgebildet. Klothoiden werden vom Writer nicht unterstützt.

Soweit der Parameter `ARC_TO_POLYLINE` gesetzt ist, werden Kreise und Bögen als Bogenpolygone gelesen, mit Ausnahme des Splines erhalten dann alle linienhaften Elemente das Attribut `out_type = out_line`.

Mit Ausnahme des Splines haben die genannten Typen im Geograf-Datenmodell eine einheitliche Grundlage, der Reader erzeugt damit auch einen identischen Satz von Formatattributen.

Splines entstehen unter Geograf insbesondere bei der Erzeugung von Höhenlinien. Der Reader erzeugt ein Linienpolygon mit den tatsächlich gespeicherten Stützpunkten. Um eine näherungsweise ähnliche Ausrundung, wie unter Geograf zu erreichen, kann der Parameter `SPLINE_PHANTOM_POINTS` gesetzt werden. Auch wenn unter Geograf polygonale Splines erzeugt werden können, sollte dieser Geometrietyp nicht mit den anderen Linien vermischt werden. Splines können nur bedingt Sachdaten aufnehmen (siehe Typ S).

Der Writer erzeugt aus bogenförmigen Geometrien automatisch die entsprechenden Geograf-Elemente, ohne dass der `out_type` gesetzt ist. Aus Linien

werden automatisch Einzellinien erzeugt, Sachdaten dann auf die Einzellinien kopiert. Wenn Polylinien oder Splines erzeugt werden sollen, muss das Attribut `out_type` explizit gesetzt sein oder der GeografTyp beim Ausgabe Feature Type angegeben werden.

T:

`out_text`
`out_multitext`

Texte besitzen eigene Koordinaten, sie werden unter Geograf oft als Elementbeschriftungen erzeugt, besitzen dann einen Bezug zu einem Geometrieelement. Reader und Writer stellen eine Beziehung zwischen Text und beschrifteten Element her, mit dem Parameter `TEXT_PARSER` kann dies gesteuert werden.

Bei mehrzeiligen Texten erzeugt der Reader das Attribut `out_text_string` mit allen Textzeilen, getrennt mit dem `MULTI_LINE_TEXT_DELIM`, als Standard ist hier der Zeilenumbruch '\n' voreingestellt. Die Textgröße wird mit der Zeilenanzahl multipliziert.

Eine besondere Bedeutung im Geograf Textstring hat der Unterstrich '_'. Wenn dieser im Text enthalten ist, erzeugt Geograf bei der Präsentation an dieser Stelle einen Bruchstrich mit Zeilenumbruch, wie bei Flurstücksnummern üblich. Der Reader erzeugt an dieser Stelle einen Zeilenumbruch und stellt der zweiten Zeile ein '↵' voran. Der Writer setzt dieser Zeichen wieder zu einem Unterstrich um.

Soweit der Schalter `MULTI_LINE_TEXT` gesetzt ist, erzeugt der Reader neben dem Attribut `out_text_string` eine zusätzliche Liste `out_text_string{}`, die die einzelnen Textzeilen enthält. Der Text wird dann mit `out_type = out_multitext` gekennzeichnet.

S:

`out_schraffur`

Schraffuren sind unter Geograf immer flächenförmige Features, auch Saum- oder Linienschraffuren sind über ihren Umring definiert. Dementsprechend setzt der Reader sie als Polygon um. Für den Writer muss das Attribut `out_type` explizit gesetzt sein oder der GeografTyp beim Ausgabe Feature Type angegeben werden. Mit den neueren Geograf-Versionen können auch 3d-Schraffurflächen erzeugt werden.

Soweit Linien- oder Kreuzschraffuren als Einzellinien gelesen werden sollen, müssen diese unter Geograf in Böschungsschraffuren umgewandelt werden.

Unter Geograf können in der aktuellen Version für Schraffuren wie auch für Böschungen und Splines keine Sachdaten vergeben werden, auch wenn es das Datenmodell vorsieht. Mit dem Writer können Sachdaten für Schraffuren erzeugt werden, diese werden von Geograf als Information angezeigt, sind aber nicht veränderbar und können nicht exportiert werden.

B:

`out_boeschung`

Datentechnisch sind Geograf-Böschungen Aggregate von Einzellinien, die jeweils nur aus zwei Vertex bestehen. Es können beliebige Formen erzeugt werden. Die Geograf-Werkzeuge bieten Werkzeuge für die Erzeugung von Böschungs-, Treppen oder Mauerschraffuren an, es können aber auch flächenhafte Linien- oder Kreuzschraffuren in die Einzellinien aufgelöst werden. Mit neueren Geograf-Versionen können auch 3-dimensionale Schraffurlinien erzeugt werden.

Für den Writer muss das Attribut `out_type` explizit gesetzt sein.

Bezüglich der Sachdaten gilt selbiges, wie bei Schraffuren.

O:

`out_objekt`

Das Geograf-Objekt besitzt als Geometrie lediglich eine 2-dimensionales Koordinatenpaar, das den Fangpunkt repräsentiert. Auf die Geometrie wird mittels einer Referenzliste verwiesen.

Mit dem Parameter `OBJECT_PARSER` bietet der Reader verschiedene Varianten der Verarbeitung dieser Referenzliste. Alle Varianten werden auch vom Writer unterstützt. (siehe Kapitel Objekte).

Zum Schreiben von Objekten muss das Attribut `out_type` explizit gesetzt sein.

G:

`out_dgm`

Aus digitalen Geländemodellen erzeugt der Reader jeweils ein Mesh. Der Writer unterstützt das Erzeugen von DGM (noch) nicht, aus Surface- und Solid-Geometrien werden die Kanten extrahiert und als Geograflinien in die Ausgabedatei geschrieben.

D:

`out_doc`

Aus unter Geograf georeferenzierten Dokumenten extrahiert der Reader das umschließende Polygon. Der Writer unterstützt das Schreiben von Dokumenten (noch) nicht.

M:

`out_model`

In Modellen werden unter Geograf die Berechnungsansätze zur Georeferenzierung von Dokumenten gespeichert. Der Reader erzeugt daraus einen Point mit den Koordinaten im Zielsystem. Der Writer unterstützt das Schreiben von Modellen (noch) nicht.

Linienpunkte

Linien besitzen unter Geograf keine eigenen Koordinaten, sie sind als Verbindung zwischen gesondert gespeicherten Punkten definiert.

Der Reader übernimmt diese Verknüpfung nur soweit, dass bei Linien die Punktnummer des Anfangs- und Endpunktes als Attribut gesetzt wird.

Der Writer erzeugt beim Schreiben von Linien aus den Linienvertex neue Punkte ohne Punktnummer und mit Art, Ebene und Katalog gleich 0. Die Linien erhalten dann einen Verweis auf diese Punkte.

Bereits geschriebene Punkte werden per Default eingefangen (verschmolzen). Mit dem Writer-Parameter `MERGE_COORDS` kann dieses Verhalten abgeschaltet werden, dann erzeugt jeder Linienvertex einen neuen Punkt.

Zum Schreiben der Linien können die Featureattribute `out_point_art`, `out_point_ebene` und `out_point_katalog` gesetzt werden, dann werden den Linienpunkten diese Werte gegeben. Bei Punkten, auf die mehrere Linien treffen, setzt beim Standard `MERGE_COORDS=Yes` die erste geschriebene Linie die Punktwerte. Soweit beim aufeinandertreffen von Linien Prioritäten gelten sollen, müssen die Linien vorher sortiert werden und über ein einziges Ausgabefeature geleitet werden.

Bei `MERGE_COORDS=No` erzeugt jede Linie ihren eigenen Punkt, es liegen also beim Aufeinandertreffen von Linien mehrere Punkte übereinander.

Geograf verschmilzt beim Import der Out-Datei Punkte mit identischer Art, Punkte mit gesetzter Art überschreiben ausschließlich Punkte der Art=0.

<p>MERGE_COORDS=YES Art der Linienpunkte gleich Linienart: out_point_art = out_art Sortierung: 1. Punkte 2. Linien nach out_art aufsteigend</p>	<p>MERGE_COORDS=YES Art der Linienpunkte gleich Linienart: out_point_art = out_art Sortierung: 1. Linien nach out_art aufsteigend 2. Punkte</p>	<p>MERGE_COORDS=No Art der Linienpunkte gleich Linienart: out_point_art = out_art Sortierung: ohne</p>

Wenn die Linienpunkte individuelle Arten erhalten sollen, müssen diese aus den Linien extrahiert und gesondert attribuiert werden. Diese Punkte überschreiben dann immer die mit Art=0 erzeugten Linienpunkte.

Objekte

Objekte unter Geograf sind Zusammenfassungen von atomaren Geometrieelementen. Das Geograf-Objekt besitzt als Geometrie lediglich eine Referenzkoordinate, auf die Geometriebestandteile wird mittels einer ID-Liste verwiesen.

Geometriebestandteile, die eine Fläche bilden sollen, werden als **Definitionsgeometrie**, andere Bestandteile als **Ausgestaltungsgeometrie** in der Liste aufgeführt. In der aktuellen Grafbat-Version führt das Geograf-Objekt kein Attribut, welches die gewollte geometrische Ausprägung des Objektes beschreibt.

Reader

Mit dem Reader-Keyword OBJECT_PARSER wird die Geometrieerzeugung der Objekte gesteuert.

OBJECT_PARSER = NO

Die Objektstrukturen werden nicht übernommen.

OBJECT_PARSER = LINK

Für das Geograf-Objekt wird ein punktförmiges Feature aus der Objektkoordinate erzeugt. Soweit Geograf zum Objekt keine Koordinate gespeichert hat, wird das Feature ohne Geometrie erzeugt. Das Objekt erhält ein Attribut `out_link_id`, welches als Referenzschlüssel für die Objektbestandteile dient.

Die Elemente, die Objektbestandteil sind, erhalten eine Liste `out_object{}.link`, welche die Schlüssel der Objekte enthält. Weiterhin wird eine Liste `out_object{}.linktyp` mit der Art der Verlinkung (1=Ausgestaltung, 2=Definition) erzeugt. Mit ListExploder und FeatureMerger können die Elemente zu den Objekten zusammengesetzt werden.

OBJECT_PARSER = LIST

Für das Geograf-Objekt wird ein punktförmiges Feature aus der Objektkoordinate erzeugt. Soweit Geograf zum Objekt keine Koordinate gespeichert hat, wird das Feature ohne Geometrie erzeugt.

Es werden die Listen `out_link{}.link` und `out_link{}.linktyp` erzeugt, die für jeden Geometriebestandteil den Referenzschlüssel (`out_link_id`) und den Geometriotyp (1=Ausgestaltung, 2=Definition) angeben. Die Objektbestandteile können sie mit dem ListBasedFeatureMerger zu Geometrien zusammengesetzt werden.

OBJECT_PARSER = LINE

Alle linienförmigen Geometriebestandteile werden zu einem Feature zusammengesetzt. Aus Linien der Geograf-Definitionsgeometrie werden Flächen erzeugt, alle anderen Linien werden zu einem Aggregat zusammengesetzt. Alle anderen Geometrien werden als Elementliste `out_elements{}` gespeichert.

OBJECT_PARSER = AREA

Wie bei dem Modus LINE werden auch hier nur die linienförmigen Objektbestandteile zur Geometrieerzeugung benutzt. Alle anderen Geometrien werden als Elementliste `out_elements{}` gespeichert.

Aus flächenförmigen Geograf-Objekten werden Flächen erzeugt, dabei werden Linien der Ausgestaltungsgeometrie ignoriert und der Liste `out_elements{}` zugefügt. Linienförmige Objekte, die keine Definitionsgeometrie besitzen, werden als Aggregat der Einzellinien erzeugt.

OBJECT_PARSER = AGGREGATE

Alle Geometriebestandteile werden zu einem Aggregat zusammengesetzt. Aus der Geograf-Definitionsgeometrie wird eine Fläche, u.U. eine Multifläche er-

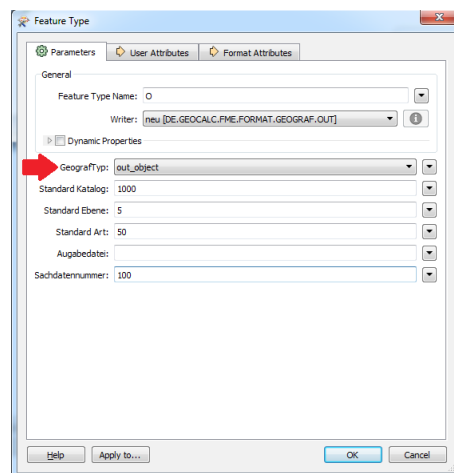
zeugt, sie erhält den Geometrienamen `out_definition`. Aus den Ausgestaltungselementen wird ein Aggregat erzeugt, welches alle Geometriearten enthalten kann. Das Aggregat erhält den Namen `out_decoration`. Wenn das Geograf-Objekt Definitions- und Ausgestaltungsgeometrie enthält, wird der Geometrienamen auf `out_composite` gesetzt, sonst je nach Inhalt auf `out_definition` oder `out_decoration`.

Mit `ADD_OBJECT_POINT` kann die Objektkoordinate als Point-Geometrie hinzugefügt werden, dann erhält dieser weitere Geometriebestandteil den Namen `out_snap`.

Alle Geometriebestandteile erhalten die obig beschriebenen Featureattribute als Geometrieattribute. Diese können mit dem `GeometryPropertyExtractor` ausgelesen werden. Soweit aus den Geograf-Objekten Geometrien erzeugt werden (`OBJECT_PARSER=LINE|AREA|AGGREGATE`), werden in der Regel die verarbeiteten Objektbestandteile nicht zusätzlich benötigt. Mit den Reader-Keywords `REMOVE_OBJECT_GEOM` und `REMOVE_OBJECT_DATA` kann gesteuert werden, ob die Definitions- und Ausgestaltungsgeometrie zusätzlich als Einzelelemente gelesen werden soll.

Writer

DerWriter unterstützt alle beim Reader genannten Varianten der Speicherung der Objektdaten. Zum Schreiben von Objekten muss das Attribut `out_type` explizit gesetzt sein, entweder als gesetztes Featureattribut oder über die Parameter des Ausgabefeatures.



Die Varianten `LINK` und `LIST` bieten sich sicher nur an, wenn die Eingangsdaten bereits solch strukturierte Listen aufweisen. Beide Varianten erfordern einigen Speicheraufwand, da die Objektinhalte zwischengespeichert werden müssen, das Objekt erst in die Grafbat-Datei geschrieben werden kann, wenn alle Objektbestandteile vorliegen.

In der Regel liegen Features als homogene Geometrien vor, so wie sie vom Reader in den Varianten `LINE` und `AREA` erzeugt werden. Diese können ausgegeben werden, ohne dass die Geometrieattribute gesetzt sind. Die Objektinhalte, in der Regel die Umringslinien erhalten dann alle die `Art = 0`. Die Objektkanten werden dabei grundsätzlich als Einzellinien ausgegeben. Wenn die Umringslinien und deren Punkte Arten mitgegeben werden sollen, kann dies für das gesamte Objekt geltend mit den Attributen

`out_point_art`, `out_point_ebene`, `out_point_katalog` und
`out_line_art`, `out_line_ebene`, `out_line_katalog`

geschehen. Nur wenn die Arten der Objektbestandteile differenziert oder Definitions- und Ausgestaltungsgeometrie gemischt werden soll, ist die Vergabe von Geometrieattributen erforderlich.

Einzelne Ausgestaltungselemente, typischerweise ein Objektsymbol oder -text können dem Objekt auch als Liste `out_elements{}` mitgegeben werden.

Soweit Objekte mit sehr differenzierten Inhalten ausgegeben werden sollen, so wie sie zum Beispiel bei Daten nach den „Baufachlichen Richtlinien (BFR)“ erforderlich sind, müssen die Geometriebestandteile mit Geometrieattributen übergeben werden. Dies ist auch dann erforderlich, wenn besondere Geometrien (Polyline, Schraffur, Spline) Objektbestandteil werden sollen. Wenn dem zu schreibenden Feature die Attribute

```
out_point_art, out_point_ebene, out_point_katalog,  
out_line_art, out_line_ebene, out_line_katalog, und  
out_text_art, out_text_ebene, out_text_katalog
```

mitgegeben werden, erhalten alle Geometriebestandteile diese Geografarten, für die keine individuellen Arten vergeben sind. Aus Flächen erzeugt der Writer grundsätzlich Definitionsgeometrie. Wenn in Aggregaten einzelne enthaltene Flächen lediglich als Ausgestaltungsgeometrie erzeugt werden sollen, muss für diese Geometriebestandteile der Geometrienname `out_decoration` vergeben werden. Wie bei allen anderen Ausgabevarianten wird auch hier eine Liste `out_elements{}` ausgewertet.

Sachdaten

Geograf ermöglicht die Erfassung von Sachdaten zu beliebigen Elementen. Die Attribute der Sachdatensätze und ihr Typ wurden in älteren Geograf-Versionen in der Datei <auftrag>.dat vereinbart, neuere Versionen verwalten die Sachdaten in einer MDB.

Zu jedem Element können auch mehrere Sachdatensätze vereinbart werden, diese Sachdatensätze können auch gleiche Attributnamen erhalten. Als Attributtypen sind zulässig: boolean, char(width), string, buffer, file, date, number(width, precision).

Reader

Mit den Standardeinstellungen erzeugt der Reader die unter Geograf vereinbarten Attributnamen mit dem vorangestellten Sachdatensatznamen, zB:

```
Flurstueck.Flur
Flurstueck.Nummer
```

Damit werden die Attributnamen des FME-Features eindeutig, auch wenn sich Attributnamen in verschiedenen Sachdatensätzen gleichen, zB:

```
Flurstueck.Flur
Flurstueck.Nummer
Eigentum.Nummer
Eigentum.Name
```

Wenn sicher gestellt ist, dass nur ein Sachdatensatz pro Element erfasst wurde oder die Attributnamen in den Sachdatensätzen eindeutig sind, kann mit dem Reader-Keyword `MERGE_DATA_ATTRIBUTES` die Erzeugung der Attribute auf den eigentlichen unter Geograf vereinbarten Attributnamen reduziert werden:

```
Flur
Nummer
```

Mit dem Keyword `DATA_ATTRIBUTE_PREFIX` kann ein zusätzlicher Präfix vereinbart werden, zB:

```
DATA_ATTRIBUTE_PREFIX OUT_
OUT_Flurstueck.Flur
OUT_Flurstueck.Nummer
```

oder:

```
DATA_ATTRIBUTE_PREFIX OUT_
MERGE_DATA_ATTRIBUTES Yes
OUT_Flur
OUT_Nummer
```

Die vereinbarten Attribute müssen unter Geograf nicht zwingend gefüllt werden, leere Attribute erzeugt der Reader mit den Standardeinstellungen nicht. Mit dem Keyword `READ_VOID_DATA` kann vereinbart werden, dass für leere Sachdaten die Attribute mit Defaultwerten erzeugt werden, String-Attribute erhalten einen leeren String "", numerische Attribute werden auf 0 gesetzt.

Es werden auch referenzierte Sachdaten (Tabellentypen S = SubTyp, P = Picklist) übernommen. Der Reader verkettet dann die Tabellen- und Attributnamen. Typische Wertetabellen enthalten die Attribute: Wert, Bezeichnung. Bei einer Leitung mit den Attributen Id, DN und einer Referenz auf eine Materialartentabelle entstehen z.b.: folgende Attribute:

```
Leitung.Id = 0815
Leitung.DN = 250
Leitung.Mat.Material.Wert = STZ
Leitung.Mat.Material.Bezeichnung = Steinzeug
```

Aus dem Geograf-Tabellentyp M (relational), hiermit können mehrere Datensätze der selben Tabelle an ein Geograf-Element gehängt werden, entstehen Listenattribute:

```
Bewohner.Art{0} = 2
Bewohner.Art{1} = 1
```

```
Bewohner.Nummer{0}          = 815  
Bewohner.Nummer{1}          = 91
```

Für jedes Feature, welches Sachdatensätze enthält, wird ein Listenattribut `out_data{}` angelegt, welches die verwendeten Tabellennamen enthält.

```
Feature: 0  
out_data{0}          'Flurstueck'  
out_data{1}          'Eigentum'
```

Writer

Zu schreibende Sachdatensätze werden in den DEF-Zeilen des Mappingfiles vereinbart.
Beispiel:

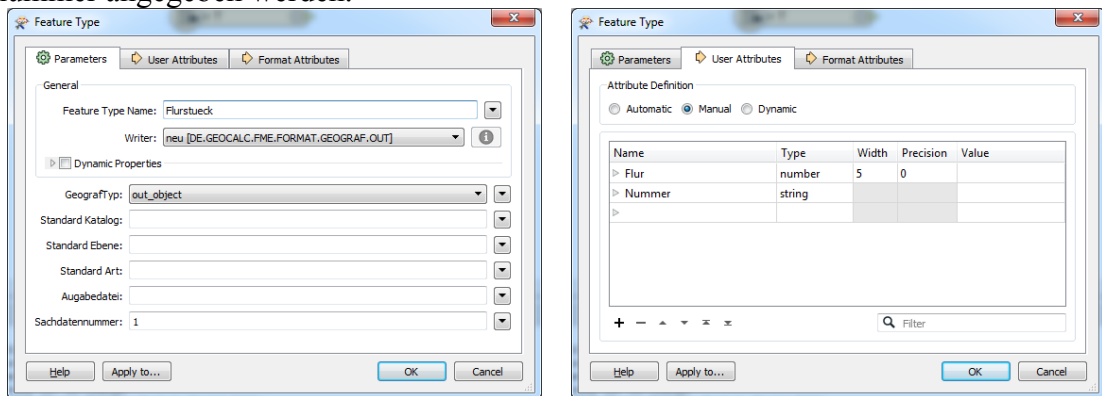
```
OUT_DEF Flurstueck \  
    out_data_id 1 \  
    Flur         number(5) \  
    Nummer      string    \  
    \
```

Sachdatensätze unter Geograf besitzen einen eindeutigen Namen und eine eindeutige Nummer. Der Sachdatensatzname entspricht dem Feature Type, die Sachdatensatznummer kann mit dem Attribut `out_data_id` vereinbart werden, möglich sind Werte von 1-9999. Unterbleibt eine Festsetzung, erzeugt der Writer automatisch eine Nummer.

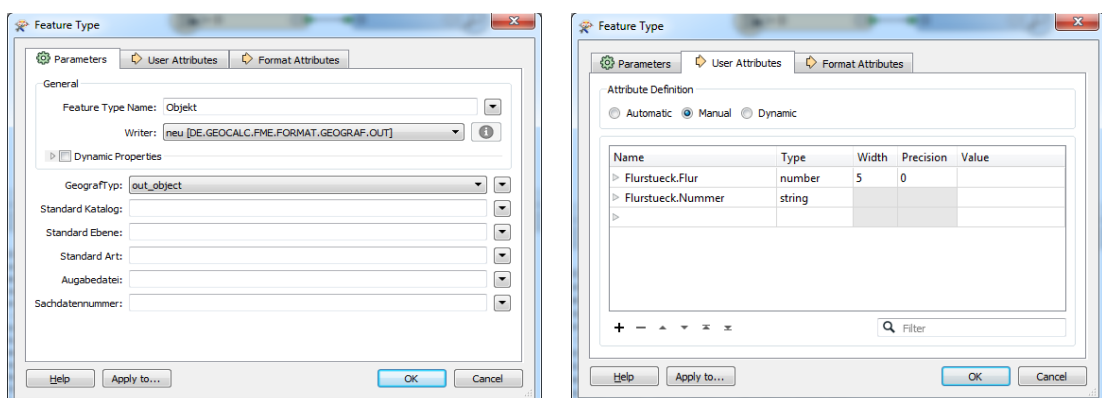
Auch in der Workbench können Sachdatensätze im FEATURE TYPE DIALOG vereinbart werden.

Einfache Sachdaten:

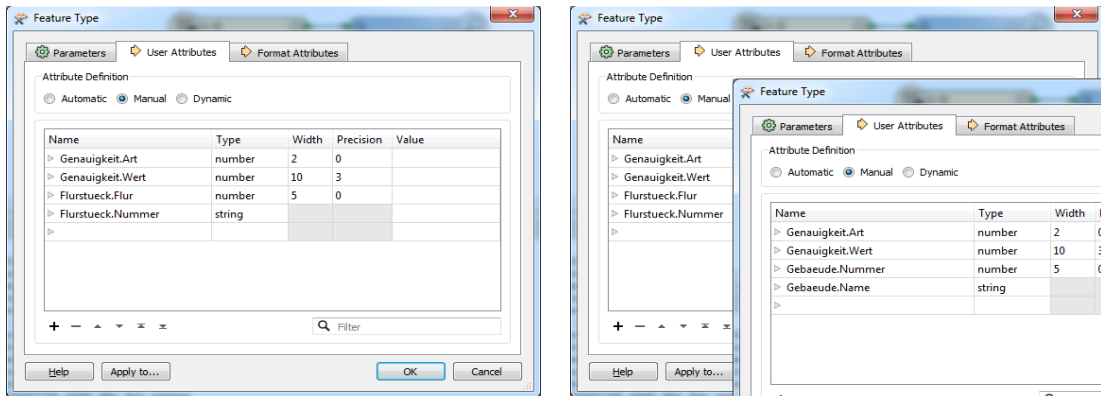
In der Regel enthält ein Feature einen Satz an Attributen. Diese werden unter dem Reiter USER ATTRIBUTES vorgegeben. Der Writer erzeugt dann eine Tabelle mit dem Namen des FeatureTypes. Die interne Geografnummer der Tabelle kann mit Sachdatennummer angegeben werden.



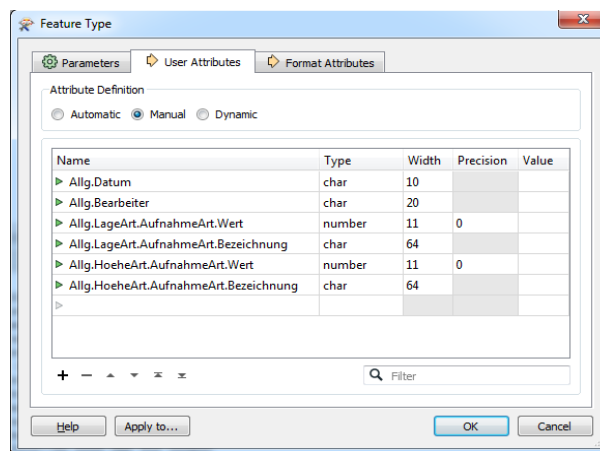
Es können auch vom FeatureType abweichende Tabellennamen angegeben werden. Diese werden dann dem jeweiligen Attribut mit Punkt getrennt vorangestellt.



Damit können für ein Geograf-Element auch mehrere Sachdatensätze vereinbart werden oder Tabellen für mehrere AusgabeFeatures verwendet werden.



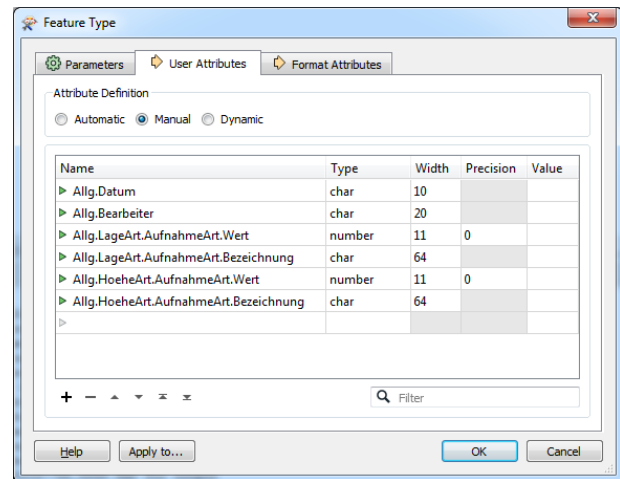
Auf diese Art und Weise können auch SubTabellen oder Picklisten vereinbart werden.



In obigen Beispiel werden für das Feature die Sachdatentabelle 'Allg' vereinbart, die die Spalten 'Datum', 'Bearbeiter', 'LageArt' und 'HoeheArt' erhält. In den Attributen 'LageArt' und 'HoeheArt' wird eine Referenz zu der Tabelle 'AufnahmeArt' gespeichert, die mit den Spalten 'Wert' und 'Bezeichnung' angelegt wird. Die referenzierten Tabellen werden als Pickliste angelegt, wenn nicht mehr als zwei Attribute vereinbart werden. Anderenfalls werden Subtabellen angelegt. Subtabelle und Pickliste haben unetr Geograf selben technischen Hintergrund, lediglich die Anzeige im Baum der Elementinfo unterscheidet sich.

Subtabellen und Picklisten werden beim Schreiben automatisch aufgefüllt, identische Datenzeilen werden nur einmal angelegt.

Soweit mehrere AusgabeFeatures mit identischen Sachdatentabellen angelegt werden sollen, brauch die Tabelle nur einmalig angelegt zu werden. Dies kann auch über ein AusgabeFeature geschehen, über welches keine Features ausgegeben werden.



Den zu schreibenden Features, muss mit dem Listenattribut `out_data{}` mitgeteilt werden, welche Sachdatensätze benutzt werden sollen. Für das folgende Feature werden die Sachdatensätze 'Allg' und 'Flurstueck' verwendet:

```
Feature: 0
out_data{0}      'Allg'
out_data{1}      'Flurstueck'
```

Soweit in Geograf die Sachdatentabellen bereits definiert sind, kann die Definition der Sachdaten aus einem Grafbat-Template gelesen werden. Auf die Definition der Feature-Attributes kann dann komplett verzichtet werden, den Features muss dann lediglich mit einer Liste `out_data{}` mitgeteilt werden, welche Sachdatensätze angelegt werden sollen (siehe Writer-Keywords `TEMPLATE_FILE` und `APPEND_TEMPLATE_DATA`).

Geograf Mapping File Example 1

```
#
GUI TITLE OUT to FFS Translation
#
LOG_FILENAME C:\daten\testdaten\gg\GGtranslation.log
#
READER_TYPE de.geocalc.fme.format.geograf.Out
OUT_DATASET C:\daten\testdaten\gg\gg.out
OUT_FEATURE_TYP Typ|Ebene|Art
OUT_ARC_TO_POLYLINE yes
OUT_OBJECT_PARSER Aggregate
OUT_REMOVE_OBJECT_GEOM yes
#
WRITER_TYPE FFS
FFS_DATASET C:\daten\testdaten\gg\gg.ffi
#

FACTORY_DEF * TeeFactory \
  FACTORY_NAME "Übertrage Elemente" \
  INPUT FEATURE_TYPE * \
  OUTPUT FEATURE_TYPE * \
  @Transform(OUT,FFS)
```

Geograf Mapping File Example 2

```
#
GUI TITLE MDB to GG Translation
#
LOG_FILENAME translation.log
READER_TYPE GEODATABASE_MDB
READER_KEYWORD MDB
MDB_DATASET C:\daten\testdaten\esri.mdb
#
WRITER_TYPE de.geocalc.fme.format.geograf.Out
WRITER_KEYWORD OUT
OUT_DATASET daten.out

# =====
OUT_DEF schacht \
  out_data_id 1
  NAME string \
  NUMMER number(5)\
  HOEHE number(10,3)

FACTORY_DEF * TeeFactory \
  FACTORY_NAME "Übertrage Elemente" \
  INPUT FEATURE_TYPE schacht \
  OUTPUT FEATURE_TYPE * \
  @SupplyAttributes(NAME, &SD_NAME)
```

Versionsinfo

- 07.02.2014 Integration der GrafBat-Formate bis Version 7
- 17.02.2014 Erste Elemente des GrafBat-Formates 8 übernommen
- 10.10.2015 Die Sachdaten aus GEOgraf >= V5 werden übernommen (FTable, FInfo)
- 28.10.2015 Fehler bei der Auswertung bogenförmiger Schraffuren beseitigt
- 05.12.2015 Der Reader erzeugte Fehlermeldungen bei einigen unbenutzten Datentypen, Berichtigt
- 17.01.2017 Objektschraffuren werden als Attributwerte übernommen
Polylinien und Bögen aus drei Punkten werden gelesen
Umlaute in den Metafiles wurden entfernt
- 01.03.2017 Version 2 des GEOgraf Reader- und Writer-Plugins
- 25.02.2018 **Der Writer hatte Probleme mit <null>-belegten Attributen**
Der Writer übernahm die Default-Ebene und -Art nicht aus der FeatureDefinition
Wenn ein Template benutzt wird:
Es können jetzt auch mehrere Sachdatensätze je Feature gefüllt werden, ohne dass diese mit den detaillierten Attributnamen vereinbart werden müssen. Die zu benutzenden Sachdatensätze sind in der FeatureDefinition unter Sachdatennummer(n) anzugeben.
- 18.06.2018 Die Interpolation von Splines wurde überarbeitet. Geograf speichert versteckt Linienzüge, die in Splines umgewandelt wurden. Diese werden nicht gelesen. Auch wurde die Splineinterpolation überarbeitet. Es werden auch 3D-Splines erzeugt.
Ebenso wurde das Schreiben von Splines überarbeitet. Es ist jetzt auch möglich, kubische Splines zu erzeugen.