

# Grafbat (OUT) Reader/Writer

## Version 2018

In diesem Dokument sind die Neuerungen des aktuellen Jahrgangs **rot** markiert.

---

<b>FORMAT</b>	• Dieses Format wird nicht unterstützt durch die FME Desktop Suite
<b>NOTES</b>	• 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 9**, der Writer schreibt die Daten ausschließlich in der aktuellen **GrafBat-Version 9**.

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.

## Neuerungen der Version 2018

Mit der Geograf-Version 9 wurde das Datenmodell erweitert, insbesondere durch die 3D-Funktionalität von weiteren Grafikelementen. Dem wird mit der aktuellen Reader- und Writer-Version Rechnung getragen.

Die Funktionalität zu Dokumenten wurde erweitert. Die Referenzen auf Rasterdaten werden ausgewertet und die Bilddaten gelesen und jetzt auch geschrieben.

Reader und Writer verarbeiten inzwischen Geograf-Mengen, ebenso werden Plotboxen gelesen und geschrieben.

Wesentliche Umbauten gab es auch beim Speichermanagement, der Speicherbedarf konnte bei großen Datensätzen um ca. die Hälfte reduziert werden, mehr Informationen dazu unter Reader Overview.

Viele kleine Detailverbesserungen tragen dazu bei, dass kaum noch Informationsverluste beim Reimport nach Geograf auftreten.

## 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			
<b>Geometry</b>	<b>Supported</b>	<b>Geometry</b>	<b>Supported</b>
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

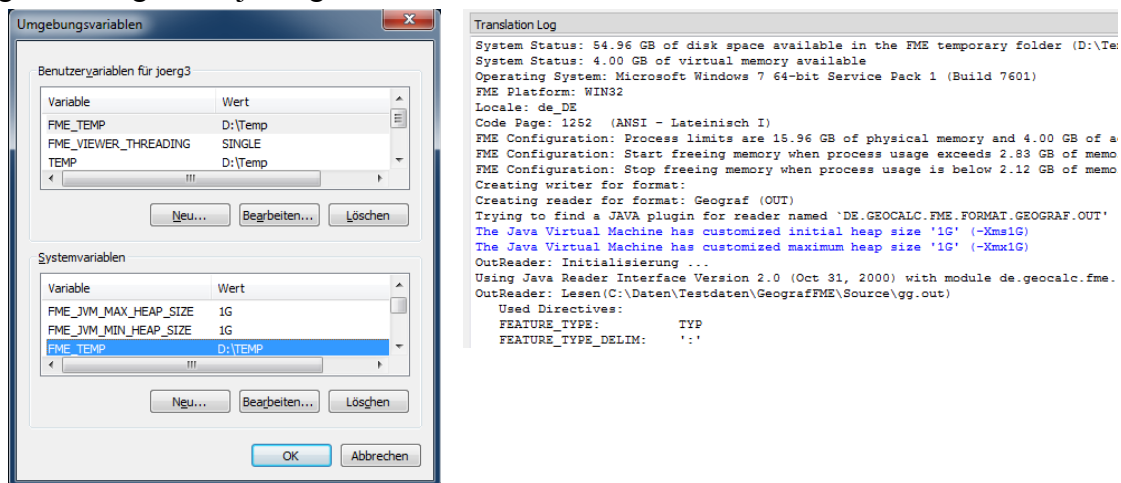
## Reader Overview

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.



Auf der Referenzinstallation wurden für die 32 bit-Version folgende Einstellungen getroffen:

`FME_JVM_MIN_HEAP_SIZE` = 128m

`FME_JVM_MAX_HEAP_SIZE` = 512m

Die Workbench arbeitet trotz Warnmeldung auch mit größer eingestellten `MAX_HEAP` (1800m), der FME-Data Inspector bricht dann aber ab. Mit der Einstellung von 512m konnte ein einfach strukturierter Datensatz mit 4 Mio Geograf-Elementen (250 MB Datei) verarbeitet werden.

Für die 64 bit Referenzinstallation wurden folgende Einstellungen getroffen:

`FME_JVM_MIN_HEAP_SIZE` = 128m

`FME_JVM_MAX_HEAP_SIZE` = 16G (gesamter verfügbarer Speicher)

Ein Datensatz mit 17 Mio Elementen (1,5 GB Datei) wurde so umgesetzt, der Reader belegte dabei ca. 5 GB Arbeitsspeicher.

# 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: <b>P</b> für Punkte <b>L</b> für Linien <b>T</b> für Texte <b>S</b> für Schraffuren <b>B</b> für Böschungen <b>O</b> für Objekte <b>M</b> für Modelle <b>D</b> für Dokumente <b>G</b> für Geländemodelle <b>X</b> für Plotboxen 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, <b>mehrere Sachdaten gehen mit Punkt getrennt in den FeatureType ein. Die Namen von relationale Tabellen fließen nicht in den FeatureType ein.</b> Die Einstellungen können mit dem Trennzeichen “:“ in Kombination angegeben werden. <b>Value:</b> TYP,EBENE,ART SACH <b>Default:</b> TYP <b>Example:</b> 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. <b>Value:</b> beliebiges ASCII-Zeichen <b>Default:</b> “:“	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. <b>Value:</b> Yes   No <b>Default:</b> Yes	Optional
SPLINE_PHANTOM_POINTS	Geograf-Splines werden als fine_line übersetzt. Als Zwischenpunkte werden die Splinekoordinaten übernommen. Um eine annähernd gleiche Ausrundung zu erhalten, kann der Schalter auf YES gesetzt werden. <b>Entsprechend der Angabe SPLINE_SEGMENT_POINTS werden dann weitere Zwischenpunkte interpoliert.</b> <b>Value:</b> Yes   No <b>Default:</b> No	Optional
SPLINE_SEGMENT_POINTS	Gibt die Anzahl der zu interpolierenden Zwischenpunkte bei der Splineinterpolation an. <b>Ca. 10 Zwischenpunkte werden benötigt, um eine Darstellung analog zu Geograf zu erhalten.</b> <b>Value:</b> Ganzzahlwert <b>Default:</b> 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.	Optional

Keyword Suffix	Value	Required/ Optional
	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 erzeugt, bzw. soweit keine Objektkoordinate 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).
	LIST	Es wird ein <code>fme_point</code> aus der Objektkoordinate erzeugt, bzw. soweit keine Objektkoordinate 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 Geometriotyp (Definition oder Ausgestaltung) der referenzierten Einzelelemente enthalten. Die Objektbestandteile erhalten das Attribut <code>out_link_id</code> , auf das die Objektliste verweist.
	LINE	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 bzw. mit <code>out_object{}.link</code> verlinkt.
	AREA	Flächenhaft ausgebildete Geografobjekte werden als Polygon gespeichert, eventuell enthaltene Ausgestaltungslinien und alle anderen Elemente gehen in der Liste <code>out_elements{}</code> auf oder werden mit <code>out_object{}.link</code> verlinkt. Linienhafte Geografobjekte werden als Aggregat aus Einzellinien gespeichert.
	AGGREGATE	Es wird ein Aggregate 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.
	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.	
	Alle Varianten kann auch der Writer verarbeiten. Die Varianten LINK und LIST können speicherintensiv sein, da hier die Bestandteile teilweise zwischengespeichert werden.	
	<b>Value:</b>	NO   LINK   LIST   LINE   AREA   AGGREGATE
	<b>Default:</b>	AGGREGATE
CREATE_OBJECT_PATHS	Die Linienzüge der Ausgestaltungsgeometrie werden als Einzelsegmente in einem Aggregat zusammengefasst. Wenn dieser Schalter auf Yes gesetzt ist, werden die Linienzüge soweit möglich als Path gebildet. Dabei werden wenn nötig auch Linienanfang und -ende getauscht. An Nodes haben Linien mit gleicher Art Vorrang.	

Keyword Suffix	Value	Required/Optional
	<p>Bei der Geograf-nach Geograf-Transformation sollte der Schalter auf No gesetzt sein, da Polylinien mit Bogenelementen dabei ihren Zusammenhang verlieren.</p> <p><b>Value:</b> Yes   No <b>Default:</b> No</p>	
REMOVE_OBJECT_GEOM	<p>Soweit aus den Objekt-<b>Definitionselementen</b> mit dem OBJECT_PARSER bereits geometrische Features erzeugt werden, ist es oft sinnvoll, die ursprünglichen Einzelteile aus dem Datensatz zu löschen.</p> <p>Elemente, die hiermit gelöscht werden und aufgrund der Einstellung OBJECT_PARSER nicht in der Objektgeometrie aufgehen, werden im Object mit der Liste <code>out_elements{}</code> gespeichert.</p> <p>Soweit OBJECT_PARSER = NO, LINK oder LIST gesetzt ist, setzt der Reader diesen Schalter auf No.</p> <p><b>Value:</b> Yes   No <b>Default:</b> Yes</p>	Optional
REMOVE_OBJECT_DATA	<p>Soweit aus den Objekt-<b>Ausgestaltungselementen</b> 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>Elemente, die hiermit gelöscht werden und aufgrund der Einstellung OBJECT_PARSER nicht in der Objektgeometrie aufgehen, werden im Object mit der Liste <code>out_elements{}</code> gespeichert.</p> <p>Soweit OBJECT_PARSER = NO, LINK oder LIST gesetzt ist, setzt der Reader den Schalter auf No.</p> <p><b>Value:</b> Yes   No <b>Default:</b> 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 Attribute <code>out_size</code> erhalten.</p> <p><b>Value:</b> Fließkommazahl <b>Default:</b> 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><b>Value:</b> Yes   No <b>Default:</b> No</p>	Optional
MULTI_LINE_TEXT_DELIM	<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_DELIM</code> gesondert gesetzt werden.</p> <p><b>Value:</b> beliebiges ASCII-Zeichen <b>Default:</b> '\n'</p>	Optional

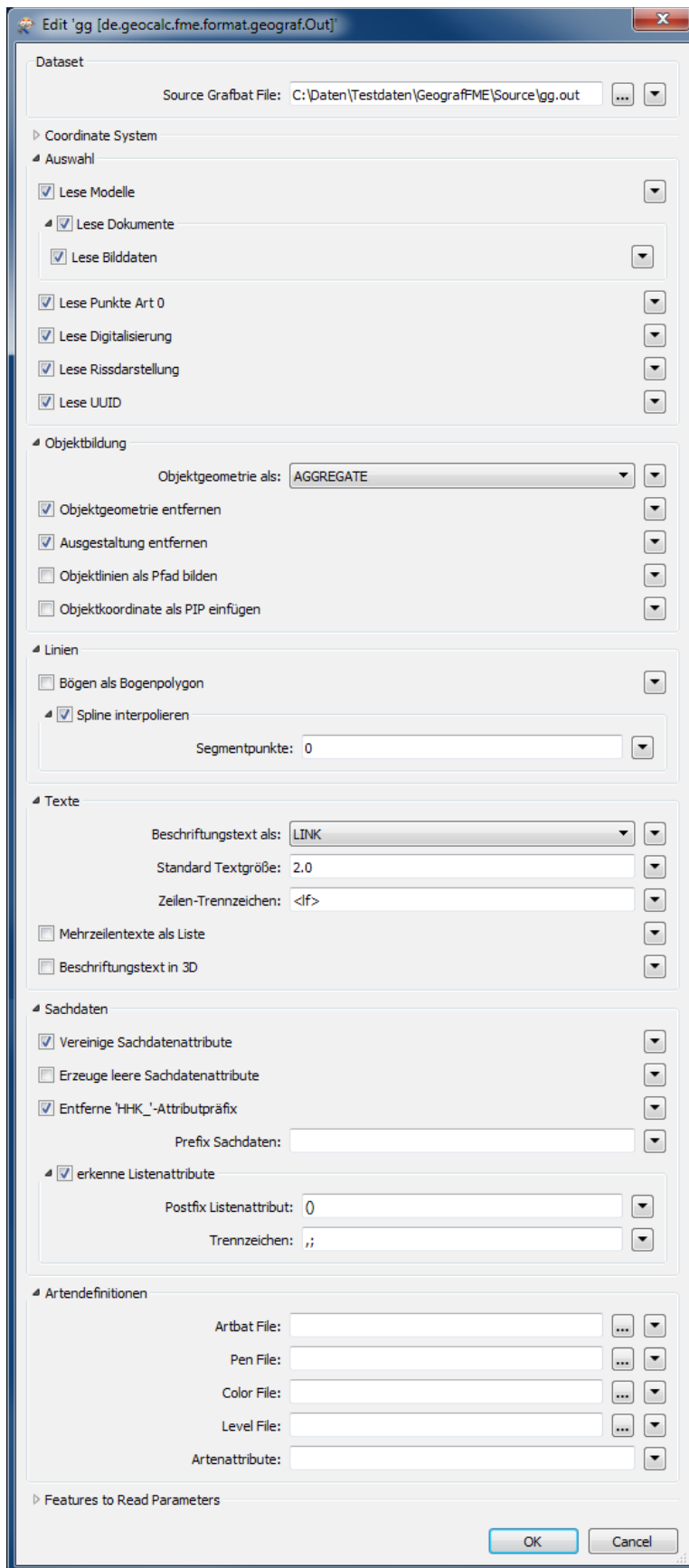
Keyword Suffix	Value	Required/Optional
TEXT_PARSER	<p>Mit der Beschriftungsfunktion hält Geograf eine Verbindung des Beschriftungstextes mit dem beschrifteten Element vor. Der Schalter TEXT_PARSER bietet folgende Einlesevarianten an:</p> <p>NO            Beschriftungen werden nicht besonders gelesen</p> <p>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.</p> <p>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.</p> <p>EXCLUSIVE    Wie in der Variante LIST erhält das beschriftete Element eine Liste mit den Textattributen. Die Texte werden dabei nicht als Features erzeugt.</p> <p>Die Varianten LINK und EXCLUSIVE werden auch durch den Writer unterstützt, bei der Variante LIST würden die Texte doppelt abgelegt werden.</p> <p><b>Value:</b>     No   LINK   LIST   EXCLUSIVE  <b>Default:</b>    LINK</p>	
TEXT_AS_3D	<p>Geograf selbst speichert die Texte inzwischen dreidimensional, die Texthöhe wird beim Beschriften automatisch aus der beschrifteten Geometrie erzeugt. In älteren Datenbeständen liegen die Texte ohne Höhe vor. Bei auf Yes gesetzten Schalter erzeugt der Reader für Beschriftungen die Höhe aus dem beschrifteten Objekt, wenn in dem Text selbst keine Höhe gesetzt ist. Bei beschrifteten Linien oder Flächen wird die Höhe aus der der Textkoordinate nächstgelegenen Kante interpoliert. Der Textbezug hat Vorrang.</p> <p><b>Value:</b>     Yes   No  <b>Default:</b>    No</p>	
READ_UUID	<p>Soweit die UUID aus Geograf ausgegeben wird, erzeugt der Reader ein entsprechendes Attribut. Werden die Elemente über den Writer ausgegeben, müssen diese eindeutig sein. Kommen identische UUIDs in einer Datei vor, übernimmt Geograf nur das letzte Element mit dieser ID.</p> <p><b>Value:</b>     Yes   No  <b>Default:</b>    Yes</p>	Optional
READ_DIGIT	<p>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.</p> <p><b>Value:</b>     Yes   No  <b>Default:</b>    No</p>	Optional
READ_OFFSET	<p>Geograf speichert mit dem Rissmodul Verschiebungsvektoren zu Punkten und Texten, bei Bedarf kann das Lesen angeschaltet werden.</p> <p><b>Value:</b>     Yes   No  <b>Default:</b>    No</p>	
READ_MODEL	<p>Die Transformation von Bitmaps und Dokumenten speichert Geograf in Modellen, die als Standard nicht gelesen werden. Das Lesen der Modelle kann angeschaltet werden.</p> <p><b>Value:</b>     Yes   No  <b>Default:</b>    No</p>	Optional
READ_DOCUMENT	<p>Das Lesen der Dokumente kann angeschaltet werden. Dokumente werden als Polygon mit dem Dokumentnamen gelesen.</p> <p><b>Value:</b>     Yes   No  <b>Default:</b>    Yes</p>	Optional



Keyword Suffix	Value	Required/Optional
READ_RASTER	<p>Für gelesene Dokumente, die auf eine Bitmap verweisen, werden die Bilddaten gelesen und ein FME-Raster-Feature erzeugt.</p> <p><b>Value:</b> Yes   No <b>Default:</b> Yes</p>	
READ_PK0	<p>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 ausgeschaltet werden, es wird so Rechenzeit und Speicher gespart.</p> <p><b>Value:</b> Yes   No <b>Default:</b> No</p>	Optional
READ_VOID_DATA	<p>Soweit dieser Schalter auf No gesetzt ist, werden für leere Sachdatenfelder die Attribute nicht angelegt. Bei 'Null' werden leere Attribute angelegt. Bei 'Default' werden die Attribute je nach Sachdatentyp mit 0 oder "" belegt.</p> <p><b>Value:</b> No,Null,Default <b>Default:</b> Default</p>	Optional
MERGE_DATA_ATTRIBUTES	<p>Geograf kann mehrere Sachdatenmasken pro Element speichern. Aus den Geograf-Sachdaten werden entsprechende Attributnamen angelegt, die als Präfix den Namen der Sachdatentabelle erhalten. Damit werden auch gleichlautende Attributnamen aus mehreren Sachdatenmasken kollisionsfrei übernommen.</p> <p>Werden Subtabellen oder Picklisten verwendet, werden diese Tabellen- und Attributnamen ebenfalls an den FME-Attributnamen mit Punkt getrennt angehängt.</p> <p>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. Für Picklisten wird hier dann nur die erste Tabellenspalte als Attribut übernommen, dort sollte der eigentliche Schlüsselwert gespeichert werden. Ebenso wird in diesem Modus der FeatureType nur aus dem letzten zum Element gespeicherten Sachdatensatz erzeugt. Beim Anlegen der Tabellen ist dann darauf zu achten, dass allgemeine Sachdatensätze zuerst, der spezielle zuletzt anzulegen ist.</p> <p>Bei einer Geograf zu Geograf-Übertragung werden in diesem Modus mehrere Sachdatensätze zu einem zusammengefasst und Picklisten nicht automatisch angelegt.</p> <p><b>Value:</b> Yes   No <b>Default:</b> No</p>	Optional
DATA_ATTRIBUTE_PREFIX	<p>Die Attributnamen der Sachdaten können ein zusätzliches Präfix erhalten.</p> <p><b>Value:</b> string <b>Default:</b> ""</p>	Optional
RESOLVE_HHK_NAMES	<p>Geograf erzeugt beim Einlesen von Sachdaten einen Präfix 'HHK_', wenn Attributnamen reservierten Bezeichnungen der benutzten Datenbank entsprechen. Dieser Präfix wird durch den Reader entfernt, wenn der Schalter auf Yes gesetzt ist.</p> <p><b>Value:</b> Yes   No <b>Default:</b> No</p>	
READ_DATA_LIST	<p>Wenn diese Schalter gesetzt ist, werden mit DATA_LIST_POSTFIX und DATA_LIST_DELIM besonders gekennzeichnete Sachdatenattribute als Liste aufgeklöst.</p>	
DATA_LIST_POSTFIX	<p>Hiermit können Erweiterungen von Attributnamen vereinbart werden, die der Reader als Liste interpretiert. Der Attributinhalt wird dann mittels der Trennzeichen DATA_LIST_DELIM zerlegt und in einer Liste gespeichert.</p> <p><b>Value:</b> string <b>Default:</b> () (runde Klammern)</p> <p>Hinweis: die geschweiften Klammern können unter Geograf nicht vereinbart werden, Grafbat ignoriert diese Sachdatensätze.</p>	

Keyword Suffix	Value	Required/ Optional
DATA_LIST_DELIM	Sachdatenattribute, die mit der Kennung DATA_LIST_POSTFIX enden, werden mittels der hier angegebenen Trennzeichen zerlegt und in einer Liste gespeichert. <b>Value:</b> string <b>Default:</b> ; (Semikolon)	
ART_SETTING_FILE	Hiermit wird die zusätzlich zu lesende Artdatei vorgegeben. Die Artdatei muss im ArtBat-Format der Version 7 vorliegen. Die Informationen der Artdatei werden vom Reader an die Features angehängt. <b>Default:</b> ""	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. <b>Default:</b> ""	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. <b>Default:</b> ""	Optional
COLOR_SETTING_FILE	Hiermit wird die zusätzlich zu lesende Farbdatei vorgegeben. Die binär vorliegende Einstellungsdatei enthält die Farbinformationen zur Bildschirmdarstellung der Features. <b>Default:</b> ""	Optional
ART_KEEP_ATTRIBUTES	Die sehr umfangreichen Attribute der Katalogdateien können mit diesem Parameter auf die notwendigen eingegrenzt werden. <b>Value:</b> string (Komma- bzw. Leerzeichen getrennte Liste der art_-Attribute) <b>Default:</b> ""	Optional

Das Metafile de.geocalc.fme.format.geograf.Out generiert folgenden Parameterdialog:



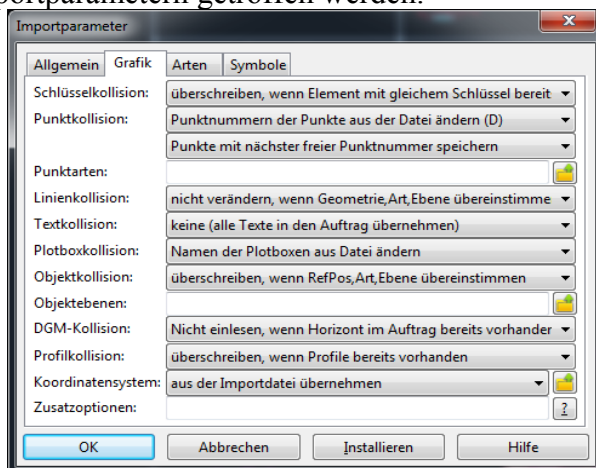
- READ\_MODEL
- READ\_DOCUMENT
- READ\_RASTER
- READ\_PK0
- READ\_DIGIT
- READ\_OFFSET
- READ\_UUID
- OBJECT\_PARSER
- REMOVE\_OBJECT\_GEOM
- REMOVE\_OBJECT\_DATA
- CREATE\_OBJECT\_PATHS
- ADD\_OBJECT\_POINT
- ARC\_TO\_POLYLINE
- SPLINE\_PHANTOM\_POINTS
- SPLINE\_SEGMENT\_POINTS
- TEXT\_PARSER
- STANDARD\_TEXT\_SIZE
- MULTI\_LINE\_TEXT\_DELIM
- MULTI\_LINE\_TEXT
- TEXT\_AS\_3D
- MERGE\_DATA\_ATTRIBUTES
- READ\_VOID\_DATA
- RESOLVE\_HHK\_NAMES
- DATA\_ATTRIBUTE\_PREFIX
- READ\_DATA\_LIST
- DATA\_LIST\_POSTFIX
- DATA LIST DELIM
- ART\_SETTING\_FILE
- PEN\_SETTING\_FILE
- COLOR\_SETTING\_FILE
- LEVEL\_SETTING\_FILE
- ART\_KEEP\_ATTRIBUTES

## 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**.

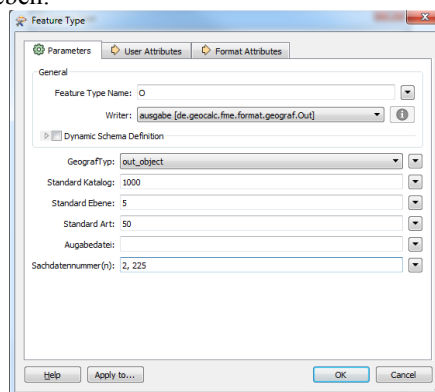
Der Writer ermöglicht auch das Schreiben von individuellen Attributen, die Geograf in Sachdatensätzen verwaltet. Geograf ermöglicht das Anlegen von mehreren Sachdatensätzen pro Feature. Auch diese können erzeugt werden, hierzu später mehr im Kapitel **Sachdaten**.

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
ATTDEFS_OVERWRITE	Mit dem Schalter kann gesteuert werden, ob die formatspezifischen Attribute der DEF-Zeile eventuell vorhandene FeatureAttribute überschreiben. <b>Value:</b> Yes   No <b>Default:</b> 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. <b>Value:</b> Yes   No <b>Default:</b> 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> <p>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. <b>Soweit ein Template bei der Ausgabe verwendet wird, brauchen die Sachdatenattribute nicht definiert werden. Unter <code>out_data_id</code> müssen dann die Nummern der Sachdatentabellen angegeben werden, die zu dem Feature erzeugt werden sollen.</b> In der Workbench werden die Featureattribute über den Reiter Format Parameters vergeben:</p>	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)

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

**char(w)** entspricht dem GeografAttributtyp s=alphanumerisch (mit Längenbegrenzung)

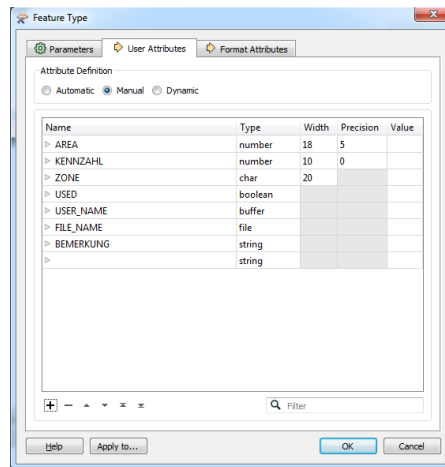
**date** entspricht dem GeografAttributtyp s=alphanumerisch (10 Zeichen Längenbegrenzung)

**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:



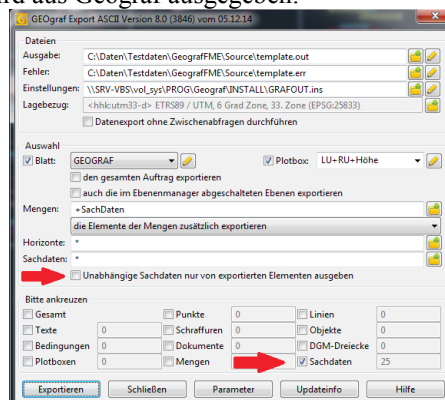
Weitere Möglichkeiten der Erzeugung von Sachdaten werden im Kapitel **Sachdaten** beschrieben.

TEMPLATE\_FILE

Es können dem Writer unter Geograf bereits definierte Sachdaten vorgegeben werden. Dann kann auf die Definition der User-Attribute verzichtet werden.

**Insbesondere bei mehreren oder komplexen Sachdatensätzen mit Subtabellen und Picklisten ist dies eine einfache Möglichkeit, feste Datenschemata einzuhalten.**

Ein Template wird aus Geograf ausgegeben:



APPEND\_TEMPLATE\_DATA

**Default:** null

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.

Keyword Suffix	Value	Required/ Optional
	<b>Value:</b> Yes   No <b>Default:</b> No	
CREATE_VOID_DATA	<p>Der Writer legt Sachdatensätze nur dann an, wenn mindestens ein Attribut in dem Feature vorhanden ist. Wird dieser Schalter auf Yes gesetzt, werden die in der Featuredefinition oder im Template vereinbarten Sachdaten auch als leerer Datensatz angelegt.</p> <b>Value:</b> Yes   No <b>Default:</b> No	
DATA_LIST_POSTFIX	<p>Der Writer erzeugt aus Listenattributen ein einzelnes Geografattribut, der Attributname wird dann mit dieser Kennung erweitert. Die Daten aus der FeatureList werden dann, separiert mit DATA_LIST_DELIM, als ein einzelner Attributwert geschrieben.</p> <b>Value:</b> string <b>Default:</b> () (runde Klammern)	
DATA_LIST_DELIM	<p>Das Trennzeichen, mit dem Listenattribute zu einem Geograf-Attributwert zusammengefügt werden.</p> <b>Value:</b> char(1) <b>Default:</b> ';' (Semikolon)	

# Feature Representation

## Alle Features

Attribute Name	Content
out_type	Der Geograf-Typ des Elements <b>Value:</b> out point, out line, ...
out_key	Eindeutiger Elementschlüssel, diese ist je OUT-Datei eindeutig. (nur Reader) <b>Value:</b> Ganzzahlwert <b>Default:</b> No default
out_extkey	Der externe Schlüssel des Elements <b>Value:</b> char(30) <b>Default:</b> No default
out_uuid	Beim Export aus Geograf können UUIDs erzeugt werden. Diese werden gelesen, wenn der Schalter READ_UUID gesetzt ist. <b>Value:</b> boolean <b>Default:</b> 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. <b>Value:</b> boolean <b>Default:</b> No
out_katalog	Der Katalog, auf der das Element liegt. <b>Range:</b> 0-9999 <b>Default:</b> 0
out_ebene	Die Ebene, auf der das Element liegt. <b>Range:</b> 0-9999 <b>Default:</b> 0
out_art	Die Art des Elements. <b>Range:</b> 0-9999 <b>Default:</b> 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. <b>In den Modi OBJECT_PARSER = LINE, AREA oder AGGREGATE wird diese Struktur für die Elemente erzeugt, die nicht mit REMOVE_OBJECT_GEOM oder REMOVE_OBJECT_DATA gelöscht werden.</b> <b>Value:</b> Integer für id und link <b>Value:</b> 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. <b>Value:</b> Integer für id und link <b>Value:</b> 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 <b>und mit REMOVE_OBJECT_GEOM oder REMOVE_OBJECT_DATA gelöscht werden</b> als Liste. Die Geometrie der Elemente wird im Attribut out_elements{}.geometry als OGC-WKT gespeichert. Auch diese Listen wertet der Writer aus.



Attribute Name	Content
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. <b>Value:</b> Integer für id und link
out_text{}	Im Modus TEXT_PARSER = LIST und EXCLUSIVE erhalten beschriftete Elemente eine Liste mit den Beschriftungstexten.
out_data{} 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. <b>Soweit je Feature nur ein Sachdatensatz erzeugt werden soll, reicht das Setzen des Attributes out_data (nur Writer).</b> <b>Wenn bei der Writer-Featuredefinition die Sachdatennummer(n) angegeben werden, ist dieses individuelle Listenattribut nicht notwendig.</b> <b>Value:</b> char(100) <b>Default:</b> No default
out_menge{} out_menge	In dem Listenattribut werden die Namen der Mengen gespeichert, in denen das Geograf-Element enthalten ist. Die Menge selbst mit Ihren Eigenschaften werden nicht gespeichert. Beim Writer reicht das Setzen des Attributes out_menge, das Feature wird dann dieser Menge zugeordnet.
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 Objektbestandteile vorrangig aus den Geometrieattributen. Wenn diese nicht gesetzt sind, greift er auf die hier aufgeführten Attribute zurück, soweit sie in dem Aggregat-Feature gesetzt sind. Das setzen der Punktarten kann auch beim Schreiben von Linien und DGM angewendet werden. <b>Soweit Ebene und Katalog hiermit nicht gesetzt werden, leitet der Writer diese Werte aus dem Aggregat ab. Die Art muss explizit gesetzt werden, sonst verbleibt sie auf 0.</b> (nur Writer)
out_error_name out_error_msg	Der Reader erzeugt diese optionalen Attribute, wenn beim Lesen der Geografdaten Fehler am Feature aufgetreten sind. Dies geschieht zum Beispiel dann, wenn unter Geograf fehlerhafte Objekte gebildet wurden, bei denen die Definitionsgeometrie keine Fläche ergibt. Trifft der Writer diese optionalen Attribute an, werden die so markierten Features zusätzlich in eine Fehlerstapeldatei abgelegt, die im Geograf-Fehlermanager abgefragt werden kann.

## Punkte

**out\_type:** out\_point

Attribute Name	Content
out_number	<p>Nummer des Punktes, eindeutig im Projekt Es ist entweder out_number oder out_name belegt <b>Value:</b> char(15) <b>Default:</b> No default. Soweit kein Wert vorgegeben wird, erzeugt Geograf selbständig eine eindeutige Punktnummer.</p>
out_name	<p>Name des Punktes, wenn die Punktnummer nicht als eindeutig gekennzeichnet ist, kann mehrfach in dem Projekt vorkommen Es ist entweder out_number oder out_name durch den Reader belegt. Der Writer wertet vorrangig das Attribut out_number aus. <b>Value:</b> char(15) <b>Default:</b> No default.</p>
out_rotation	<p>Drehwinkel des Punktes in Grad entsprechend den FME-Konventionen <b>Range:</b> 0.0 – 360.0 <b>Default:</b> 0.0</p>
out_size	<p>Symbolgröße des Punktes. Im Regelfall ist diese auf 0.0 gesetzt. <b>Range:</b> Fließkommawert <b>Default:</b> 0.0</p>
out_size_prop	<p>Eigenschaften der Symbolgröße Wenn unter Geograf die Symbolgröße individuell eingestellt wird, ist dieses Attribut ungleich 'A' belegt. A = Größe aus Artdefinition, out_size ist mit 0.0 belegt M = Größe in Meter F = Größe als Faktor zur Artdefinition Den Werten 'M' und 'F' kann getrennt mit einem Doppelpunkt ein Fließkommawert folgen. Dieser gibt die zweite Größe an, wenn für X- und Y-Richtung unterschiedliche individuelle Größenangaben eingestellt wurden. <b>Range:</b> String <b>Default:</b> A</p>
out_l_stat out_h_stat	<p>Lagegenauigkeit Höhengenauigkeit 0 = ungültig 1 = digitalisiert 2 = gerechnet 3 = fest <b>Range:</b> 0-3 <b>Default:</b> 2</p>
out_date	<p>Datum der Berechnung (nach FME-Datumskonventionen) <b>Range:</b> date <b>Default:</b> No default</p>
out_symbol{}.out_oska out_symbol{}.out_rotation out_symbol{}.out_size out_symbol{}.out_size_prop	<p>Mehrfachsymbole werden in dieser Liste abgelegt. Das Attribut oska beinhaltet Katalog, Ebene und Art, mit Punkt getrennt.</p>
Attribute bei <b>READ OFFSET</b> = Yes	
out_x_off out_y_off out_z_off	<p>Verschiebevektor für die Rissansicht <b>Range:</b> Fließkommawert <b>Default:</b> 0.0</p>
out_color	<p>Symbolfarbe entsprechend den FME-Konventionen <b>Range:</b> 0.0-1.0, 0.0-1.0, 0.0-1.0 <b>Default:</b> &lt;null&gt;</p>
Attribute bei <b>READ DIGIT</b> = Yes	
out_ansatz	<p>Die Ansatznummer der Koordinatenberechnung <b>Range:</b> 0-999999999 <b>Default:</b> 0</p>

Attribute Name	Content
out_dig_trafo	Tranformationspunkt <b>Range:</b> 0-1 <b>Default:</b> 0
out_dig_dot	digitalisierter Punkt <b>Range:</b> 0-1 <b>Default:</b> 0
out_dig_aus	ungültig für Transformation <b>Range:</b> 0-1 <b>Default:</b> 0
out_dig_rand	Randpunkt (dieser Schalter wird jetzt offensichtlich dazu benutzt, um die Verzerrung einzuschalten) <b>Range:</b> 0-1 <b>Default:</b> 0
out_dig_soll	Passpunkt aus ehemalig digitalisiertem Punkt <b>Range:</b> 0-1 <b>Default:</b> 0
out_dig_gew	Gewicht der digitalisierten Koordinaten <b>Range:</b> 0-9999 <b>Default:</b> 0
out_l_gew out_h_gew	Lage-, Höhengewicht <b>Range:</b> 0-9999 <b>Default:</b> 0
out_l_ctrl out_h_ctrl	Lage-, Höhenkontrolle <b>Range:</b> 0-9999 <b>Default:</b> 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\_pointline (nur Writer)  
**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.

Geograf speichert intern die Polylinien auf unterschiedliche Art und Weise. Die Zwischenpunkte können als eigene Position gespeichert werden, sie bestehen dann unabhängig von Punkten. Die Zwischenpunkte können aber auch als Referenz auf einen bestehenden Punkt gespeichert werden, die Lage des Vertex wird dann durch diesen Punkt bestimmt.

Der Reader erzeugt alle Linien, die neben dem Anfangs- und Endpunkt auch Zwischenpunkte aufweisen, als out\_polyline. Der Writer erzeugt aus diesen wiederum Geograf-Polylinien, wobei die Zwischenpunkte als Referenz gespeichert werden, wenn an dieser Position vorher ein out\_point geschrieben wurde. Anderenfalls wird der Vertex als eigene Position gespeichert. Wird eine Linie als out\_pointline ausgegeben, werden immer Referenzen auf Punkte geschrieben. Sollte an dieser Position noch kein Punkt erzeugt worden sein, entsteht hier ein neuer mit den Attributen out\_point\_katalog, out\_point\_ebene, out\_point\_art.

Attribute Name	Content
out_pattern_side_change	Linienmuster wird gespiegelt. <b>Value:</b> boolean <b>Default:</b> No
out_pattern_start_change	Linienmuster umdrehen. <b>Value:</b> boolean <b>Default:</b> No
out_size	Linienbreite in Koordinateneinheiten <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
out_pattern_dist	Startwert des Linienmusters in Koordinateneinheiten <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
out_color	Linienfarbe entsprechend den FME-Konventionen <b>Range:</b> 0.0-1.0, 0.0-1.0, 0.0-1.0 <b>Default:</b> <null>
out_number1	Nummer des ersten Linienpunktes (nur Reader) <b>Value:</b> char(15)
out_number2	Nummer des zweiten Linienpunktes (nur Reader) <b>Value:</b> 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. <b>Default:</b> 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 (2,5D) übertragen.

Attribute Name	Content
out_rotation	Werte ohne Bedeutung
out_size	
out_typ	Splinetyp ( <b>nur Reader</b> ) 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 <b>Range:</b> 1-9 <b>Default:</b> No default
out_spline_info	Art der Linienzugerzeugung 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) <b>Range:</b> String <b>Default:</b> No default

## Texte

**out\_type:** out\_text

**out\_type:** out\_multi\_text

Multitexte werden nur erzeugt, wenn der Schalter MULTI\_LINE\_TEXT auf Yes gestellt ist.

Mit der Version 8 können Geograf-Texte auch dreidimensional gespeichert werden. Beim beschriften von 3D-Geometrien geschieht dieses automatisch. Ältere Daten enthalten keinen Z-Wert. Dieser kann durch den Reader aus den beschrifteten Elementen generiert werden, dazu ist der Schalter TEXT AS 3D auf 'Yes' zu setzen.

Attribute Name	Content
out_rotation	Drehwinkel des Textes in Grad entsprechend den FME-Konventionen <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
out_size	Größe des Textes in Koordinateneinheiten. Im Regelfall ist dieser auf 0.0 gesetzt. Die Größe entnimmt dann Geograf der Artdatei. 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. <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
out_size_prop	Eigenschaften der Symbolgröße Wenn unter Geograf die Symbolgröße individuell eingestellt wird, ist dieses Attribut ungleich 'A' belegt. A = Größe aus Artdefinition, out_size ist mit 0.0 belegt M = Größe in Meter F = Größe als Faktor zur Artdefinition <b>Range:</b> String <b>Default:</b> A
out_text_size	Größe des Textes in Koordinateneinheiten. Wenn mit Geograf keine individuelle Größe festgelegt ist, wird der Wert mit STANDARD_TEXT_SIZE belegt. <b>Range:</b> Fließkommawert <b>Default:</b> STANDARD_TEXT_SIZE
out_text_string	Inhalt des Textes, mehrzeilige Texte erhalten als Trennzeichen MULTI_LINE_TEXT_DELIM <b>Value:</b> string
out_text_string{}	Inhalt der Textzeilen bei Multitexten wird vom Reader nur bei MULTI_LINE_TEXT=Yes erzeugt <b>Value:</b> string
out_text_formel	Textformel bei Beschriftungstexten, mehrzeilige Texte erhalten als Trennzeichen MULTI_LINE_TEXT_DELIM <b>Value:</b> string
out_text_formel{}	Textformel bei Multitexten wird vom Reader MULTI_LINE_TEXT=Yes erzeugt <b>Value:</b> string

Attribute Name	Content																		
out_rahmen	<p>Besondere Hervorhebung durch Textrahmen</p> <table> <tr><td>0 = ohne Rahmen</td><td>Bezug vom Zentrum</td></tr> <tr><td>1 = Kasten</td><td>Bezug von den Ecken</td></tr> <tr><td>2 = Kreis</td><td>Bezug vom Zentrum</td></tr> <tr><td>3 = einfach unterstrichen</td><td>Bezug vom Unterstrich</td></tr> <tr><td>4 = doppelt unterstrichen, eng</td><td>Bezug vom Unterstrich</td></tr> <tr><td>5 = durchgestrichen</td><td>Bezug vom Durchstrich</td></tr> <tr><td>6 = ohne Rahmen</td><td>Bezug von den Ecken</td></tr> <tr><td>7 = doppelt unterstrichen, breit</td><td>Bezug vom oberen Unterstrich</td></tr> <tr><td>8 = kleiner Kreis</td><td>Bezug vom Zentrum</td></tr> </table> <p>Den Rahmen 5 gibt es seit einigen Grafbat-Versionen nicht mehr. Ist dieser gesetzt, erzeugt Geograf daraus einen Rahmen 0 und setzt das Attribut crossout auf 'Yes'</p> <p><b>Range:</b> 0-8 <b>Default:</b> 0</p>	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
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																		
out_crossout	<p>Text ist durchgestrichen</p> <p><b>Value:</b> boolean</p>																		
out_frei	<p>Freistellart</p> <table> <tr><td>0 = Kasten</td></tr> <tr><td>1 = ohne</td></tr> <tr><td>2 = Kreis</td></tr> </table> <p><b>Range:</b> 0-2 <b>Default:</b> 0</p>	0 = Kasten	1 = ohne	2 = Kreis															
0 = Kasten																			
1 = ohne																			
2 = Kreis																			
out_position	<p>Position, auf die sich die Textkoordinate bezieht</p> <table> <tr><td>0 = links-unten</td></tr> <tr><td>1 = mitte-unten</td></tr> <tr><td>2 = rechts-unten</td></tr> <tr><td>3 = links-mitte</td></tr> <tr><td>4 = mitte-mitte</td></tr> <tr><td>5 = rechts-mitte</td></tr> <tr><td>6 = links-oben</td></tr> <tr><td>7 = mitte-oben</td></tr> <tr><td>8 = rechts-oben</td></tr> </table> <p>Hinweis: Der FME-DataInspector zeigt den Text immer rechts-obenhalb der Textkoordinaten an. Die Einfügeposition kann in Fremdsysteme übersetzt werden (z.B. ESRI, GeoMedia), dort wird der Text dann wie unter Geograf positioniert.</p> <p>Siehe: <a href="https://knowledge.safe.com/articles/16546/text-justification.html">https://knowledge.safe.com/articles/16546/text-justification.html</a></p> <p><b>Range:</b> 0-8 <b>Default:</b> 0</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									
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																			
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><b>Range:</b> Fließkommawert <b>Default:</b> 0.0</p>																		
out_color	<p>Textfarbe entsprechend den FME-Konventionen</p> <p><b>Range:</b> 0.0-1.0, 0.0-1.0, 0.0-1.0 <b>Default:</b> &lt;null&gt;</p>																		
out_rel_flag	<p>Attribute bei <b>TEXT_PARSER</b> != No</p> <p>Flags für relative Texte</p> <p><b>Range:</b> 0-9999 <b>Default:</b> 0</p>																		
out_x_rel out_y_rel	<p>Koordinatenoffset der Beschriftung</p> <p><b>Range:</b> Fließkommawert <b>Default:</b> 0.0</p>																		



Attribute Name	Content
out_r_rel	Winkeloffset der Beschriftung <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
	Attribute bei <b>READ OFFSET</b> = Yes
out_x_off out_y_off	Verschiebevektor für die Rissansicht <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
out_r_off	Winkeloffset für die Rissansicht <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
	Attribute bei <b>READ DIGIT</b> = Yes
out_dig_flag	Digitalisierflags <b>Range:</b> 0-9 <b>Default:</b> 0

## Schraffuren

**out\_type:** out\_schraffur

Aus Geograf-Schraffuren wird eine Geometrie des Typs fine\_polygon erzeugt. **Nur beim Schraffurtyp 8 = 'geschlossene Saumschraffur' entsteht ein offener Linienzug.**

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

Attribute Name	Content
out_rotation	Drehwinkel der Schraffur in Grad entsprechend den FME-Konventionen <b>Range:</b> 0.0-360.0 Grad <b>Default:</b> 0.0
out_size	Schraffurabstand bei linienhaften Schraffuren in mm Ploteinheiten <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
out_size_prop	Eigenschaften des Schraffurabstandes Je nach Schraffurtyp ist dieses Feld belegt: A = ohne Abstand M = Größe in Meter M:0.00 = der zweite Abstand bei out_sub_typ = 10 <b>Range:</b> String <b>Default:</b> A
out_sub_typ	Spezifizierung des Schraffurtyps 0 = normale Schraffur (Linien) 1 = Kreuzschraffur 2 = Rasterschraffur 3 = Flächenschraffur 4 = schraffierte Abstandsfläche 5 = geschlossene Saumschraffur 6 = deckende Rasterschraffur 7 = Schraffur via Schraffurart 8 = offene Saumschraffur 10 = unregelmäßige Rasterschraffur 11 = Linienumring <b>Default:</b> 0
out_color	Schraffurfarbe entsprechend den FME-Konventionen <b>Range:</b> 0.0-1.0, 0.0-1.0, 0.0-1.0 <b>Default:</b> <null>
out_name	Schraffurname <b>Range:</b> char(80) <b>Default:</b> Void

## 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 <b>Range:</b> 0.0-1.0, 0.0-1.0, 0.0-1.0 <b>Default:</b> <null>

## Objekte

**out\_type:** out\_object

Attribute Name	Content
out_name	Objektname <b>Range:</b> char(80) <b>Default:</b> Void
out_object_typ	Objekttyp 0 = Normales Objekt 1 = Splineobjekt 2 = Beschriftung <b>Range:</b> 0-2 <b>Default:</b> 0
out_date	Datum der Objekterzeugung (nach FME-Datumskonventionen) <b>Default:</b> <null>
out_x out_y	Objektkoordinaten <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
out_ansatz	Ansatznummer des Objektes
out_l_stat	Lagegenauigkeit 0 = ungültig 1 = digitalisiert 2 = gerechnet 3 = fest <b>Range:</b> 0-3 <b>Default:</b> 2
out_oberobjekt	Elementschlüssel des Oberobjekts <b>Default:</b> 0
out_link_id	Oberobjekte, auf die verwiesen wird, erhalten diesen Schlüssel. Der Reader legt hier eine Kopie des Wertes aus out_key ab. Das Unterobjekt verweist mit out_oberobjekt auf diesen Schlüssel. Beim Writer können beliebige Ganzzahlwerte gesetzt werden, diese werden dann in die Elementschlüssel der Out-Datei umgesetzt. <b>Default:</b> 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. Der Writer erzeugt eine Objektschraffur, wenn out_sch_art gesetzt ist. Katalog und Ebene werden vom Objekt abgeleitet, wenn sie nicht explizit gesetzt sind.
Attribute bei <b>READ OFFSET</b> = Yes	
out_x_off out_y_off	Verschiebevektor in Koordinateneinheiten <b>Range:</b> Fließkommawert <b>Default:</b> 0.0
out_object_geom	Auflistung der Inhalte der Objektgeometrie (nur Reader) Der String gibt Auskunft über die Geometrietypen des Objektinhalts. 1. Zeichen: A = Fläche 2. Zeichen: L = Linie 3. Zeichen: S = Schraffur 4. Zeichen: B = Böschung 5. Zeichen: T = Text 6. Zeichen: P = Punkt Geometrietypen, die nur einmalig im Objekt vorkommen, werden mit Kleinbuchstaben, mehrmalige mit Großbuchstaben markiert. Ist ein Typ nicht enthalten, ist das Zeichen '0' (Null). <b>Range:</b> char(6) <b>Default:</b> 000000

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

## Dokumente

`out_type`: `out_raster`

`out_type`: `out_doc`

Aus Geograf-Dokumenten werden Polygonfeatures erzeugt. **(nur Reader)**

Für Dokumente des Typs 0 = Bitmaps werden die Rasterdaten soweit möglich aus der referenzierten Bilddatei Reader-intern mit der Funktion `@File()` nachgeladen, die Geometrie in ein `fme_raster` umgewandelt. Die Funktion `@File()` lädt auch die Georeferenzierung der Bitmaps aus eventuell vorliegenden world-files. Diese wird durch die Georeferenzierung aus der Out-Datei überschrieben.

Die Georeferenzierung der Daten werden im `fme_raster` oder `fme_polygon` abgelegt. Für das Polygon gilt die Vereinbarung, dass die erste Koordinaten als links-oben abgelegt und interpretiert wird, die weiteren folgen im Uhrzeigersinn.

Der Writer verarbeitet die Typen `out_raster` und `out_doc` unterschiedlich. Die Bilddaten aus `out_raster` werden als Bitmapdatei erzeugt und im Verzeichnis der Ausgabedatei abgelegt. Eventuell vorhandene Bilddateien werden dabei überschrieben. Der Dateityp wird aus `out_name` extrahiert, ist keine Dateierweiterung angegeben, wird eine PNG-Datei erzeugt. Die Referenz auf die Bilddatei wird in der Out-Datei auf die so erzeugte Bilddatei ohne Pfadangabe gesetzt.

Die Ausgabe als `out_doc` geschieht immer nur als Referenz auf die ursprüngliche Bilddatei, Bilddaten werden nicht geschrieben. Die Referenz wird mit kompletter Pfadangabe gesetzt, die Georeferenzierung wird aus dem Polygon mit 4 Eckpunkten abgeleitet. Wird ein Feature ausgegeben, dass nicht aus 4 Eckpunkten besteht, erzeugt der Writer eine Bounding-Box und gibt diese als Georeferenzierung aus.

Attribute Name	Content
<code>out_name</code>	Name des Dokuments <b>Range:</b> <code>char(255)</code>
<code>out_typ</code>	Typ des Dokuments: 0 = Bitmap 1 = DWG/DXF-Datei 2 = COM-Objekt 3 = WMS-Bitmap 4 = DGN-Datei 5 = Plot-Datei 6 = PDF-Datei <b>Default:</b> 0
<code>out_pen</code>	Stift für 1Bit-Bilder <b>Range:</b> 0-9999
<code>out_trans</code>	Transparente Darstellung <b>Range:</b> 0-9
<code>out_stufe</code>	Hierarchiestufe der Darstellung <b>Range:</b> 0-9999
<code>out_system</code>	Nummer des Digitalisiersystems <b>Range:</b> 0-9999
<code>out_colorkey</code>	Farbnummer für 1Bit-Bilder <b>Range:</b> 0-9999
<code>out_doc_lux</code> <code>out_doc_luy</code> <code>out_doc_rox</code> <code>out_doc_roy</code> <b>oder:</b> <code>out_doc_width</code> <code>out_doc_height</code>	Wenn das Dokument nicht aus Bilddaten besteht oder diese nicht gelesen werden konnten, werden die Dokument-Koordinaten vom Reader in diesen Attributen gespeichert und vom Writer ausgewertet. Die Angaben unterscheiden sich je nach Typ des Dokuments. Bei CAD-Dateien werden die Eckkoordinaten im (mathematischen) Koordinatensystem dieser Daten angegeben, bei den anderen Dateitypen sind die Daten im Bildkoordinatensystem angegeben (x' nach rechts, y' nach unten). Bei den anderen Dateitypen genügt auch die Angabe der Breite und Höhe (in Einheiten des Dokuments).

## DGM

**out\_type:** out\_dgm

Aus Geograf-DGM-Modellen werden Geometrien des Typs Mesh erzeugt.

Der Writer kann aus allen Oberflächen DGMs erzeugen. Es werden automatisch Horizonte angelegt, die mit 1 beginnend durchnummeriert werden. Ist das Attribut `out_horizont` vergeben, wird diese Nummer benutzt. So können auch mehrere Features auf denselben Horizont gelegt werden.

Geograf kann nur mit Oberflächen rechnen, deren Dreiecke sich im 2D-Raum nicht überlagern. Diese Oberflächengestalt kann durch den Writer nicht sicher gestellt werden.

Volumenkörper (Solids) werden nicht als DGM übertragen.

Wie bei Objekten kann die Art der enthaltenen Dreieckspunkte mit den Attributen `out_point_art`, `out_point_ebene`, `out_point_katalog` gesteuert werden.

Attribute Name	Content
<code>out_horizont</code>	Horizontnummer <b>Range:</b> 0-9997
<code>out_name</code>	Name des Horizonts <b>Range:</b> char(255)

## Plotboxen

**out\_type:** out\_box

Aus Geograf-Dokumenten werden Polygonfeatures erzeugt. Wie bei Dokumenten gilt hier die Vereinbarung, dass die erste Koordinaten als links-oben abgelegt und interpretiert wird, die weiteren folgen im Uhrzeigersinn.

Attribute Name	Content
out_name	Name der Plotbox <b>Range:</b> char(50)



## Fehlermeldungen

**out\_type:** out\_error (nur Writer)

Der Writer kann Features als Fehlermeldungen in eine Geograf-Stapeldatei ausgeben. Die so erzeugte Datei kann in den Geograf-Fehlermanager geladen werden.

Attribute Name	Content
out_file oder stp_file	Name der Fehlerdatei Wird dieser nicht angegeben, wird eine Fehlerdatei mit dem Namen der Out-Datei, erweitert um '.stp' angelegt. <b>Range:</b> string
out_x, out_y oder stp_x, stp_y	Koordinaten der Fehlermeldung Sind diese nicht angegeben, wird eine Koordinate aus der Featuregeometrie erzeugt. <b>Range:</b> Fließkommawert
out_error_name oder out_name oder stp_name	Name des Fehlers <b>Range:</b> string
out_error_msg oder out_msg oder stp_msg	Fehlermeldung <b>Range:</b> string

# 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 Arten

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

## Punktarten

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

## Linienarten

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 <b>Range:</b> 0-9999
art_size art_form{}.size	Abstand der Form von der Grundlinie <b>Value:</b> Fließkommawert
art_pen art_form{}.pen	Stiftnummer der Form <b>Range:</b> 0-9999
art_pen_size art_form{}.pen_size	Stärke des FormStiftes <b>Value:</b> Fließkommawert
art_pen_color art_form{}.pen_color	Farbe des FormStiftes <b>Value:</b> String (RGB-Farbwerte)
art_pen_mode art_form{}.pen_mode	Stiftmodus des Abschluss der Linienform 0 = rund 1 = eckig

## Textarten

Attribute Name	Content
art_slant	Textneigung in Grad <b>Value:</b> Fließkommawert
art_factor	Verhältnis Breite/Höhe in Prozent <b>Value:</b> 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 <b>Range:</b> 0-8 <b>Default:</b> 0
art_font	Name des Fonts <b>Value:</b> char(40)
art_arrow_size	Pfeilgröße für Bezug in mm <b>Value:</b> Fließkommawert
art_pos_absolute	Schwerpunkt ist fest <b>Value:</b> boolean
art_mode	Zeichenmodus 0 = SchraffurDeckend 1 = Schraffur- und LinienDeckend 2 = VollDeckend 3 = VollTransparent <b>Range:</b> 0-3 <b>Default:</b> 0
art_bold	TrueTypeFont fett <b>Value:</b> boolean
art_border	Linienart für den Textrahmen <b>Range:</b> 0-9999
art_line	Linienart für Bezugsstrich <b>Range:</b> 0-9999
art_area	Schraffurart für Texthintergrund <b>Range:</b> 0-9999

## Schraffurarten

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

## Objektarten

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

`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, **sie werden nach Geograf als Polylinie übergeben.**

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-dimensionale 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 kann aus beliebigen Oberflächen DGM erzeugen. Flächen, die nicht als Dreiecksnetz vorliegen, werden dann trianguliert.**

**D:**

`out_doc`

`out_raster`

Aus unter Geograf georeferenzierten Dokumenten extrahiert der Reader das umschließende Polygon. **Wenn Bilddaten verfügbar sind, werden diese gelesen und geschrieben.**

**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.

**X:**

`out_box`

**Plotboxen werden als Polygon gelesen.**

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

# 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.

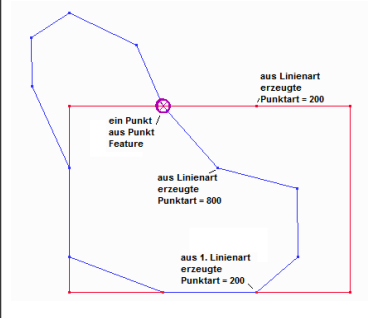
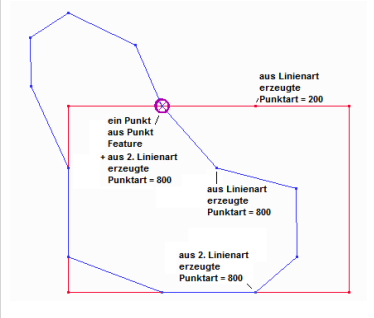
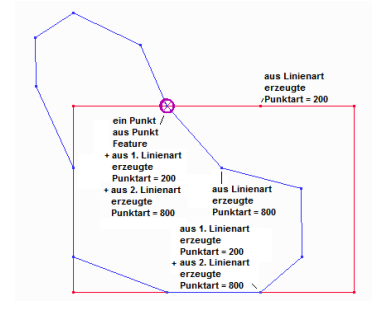
Bereits geschriebene Punkte werden per Default eingefangen (verschmolzen). Mit dem Writer-Parameter `MERGE_COORDS` kann dieses Verhalten abgeschaltet werden, dann erzeugt jeder Linienvertex einen neuen Punkt.

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. **Sind Ebene und Katalog für die Punkte nicht explizit gesetzt, werden die Werte der Linie benutzt.**

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><code>MERGE_COORDS=YES</code>                      Art der Linienpunkte gleich Linienart:  <code>out_point_art = out_art</code>                      Sortierung:                      1. Punkte                      2. Linien nach <code>out_art</code> aufsteigend</p>	<p><code>MERGE_COORDS=YES</code>                      Art der Linienpunkte gleich Linienart:  <code>out_point_art = out_art</code>                      Sortierung:                      1. Linien nach <code>out_art</code> aufsteigend                      2. Punkte</p>	<p><code>MERGE_COORDS=NO</code>                      Art der Linienpunkte gleich Linienart:  <code>out_point_art = out_art</code>                      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 Objektbestandteile werden als Elementliste `out_elements{}` gespeichert, wenn diese aufgrund des Schalters REMOVE\_OBJECT\_DATA gelöscht werden. Die damit nicht gelöschten Daten erhalten die Listen `out_object{}.link` und `out_object{}.linktyp`, womit auf das Objektattribut `out_link_id` verwiesen wird. Bei Objekten ohne Liniengeometrie wird ein `fme_point` aus der Objektcoordinate erzeugt.**

OBJECT\_PARSER = AREA

Wie bei dem Modus LINE werden auch hier nur die linienförmigen Objektbestandteile zur Geometrieerzeugung benutzt. **Aus flächenförmigen Geograf-Objekten werden Flächen ausschließlich aus der Definitionsgeometrie erzeugt, bei linienförmigen Geograf-Objekten gehen alle Linien in die Geometrie ein. Wie im Modus LINE werden alle anderen Objektbestandteile als Elementliste**

`out_elements{}` oder als Linklisten `out_object{}.link,`  
`out_object{}.linktyp` gespeichert. Bei Objekten ohne Liniengeometrie wird ein `fine_point` aus der Objektkoordinate erzeugt.

## OBJECT\_PARSER = AGGREGATE

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

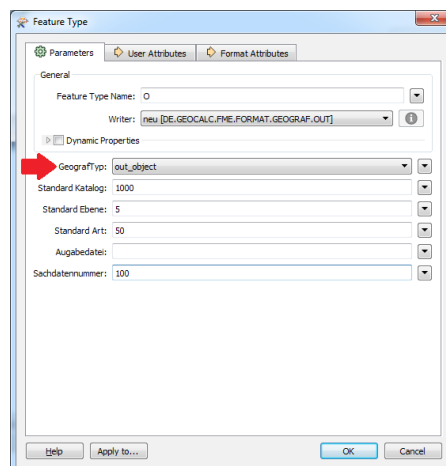
Mit `ADD_OBJECT_POINT` kann die Objektkoordinate als Point-Geometrie hinzugefügt werden, dann erhält dieser weitere Geometriebestandteil den Namen `out_snap`.

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.

Bei flächenförmigen Objektgeometrien (Geografobjekte mit Definitionsgeometrie) setzt der Reader die Umringslinien zu einem geschlossen Pfad zusammen. Die Linien der Ausgestaltungsgeometrie werden als Aggregat gelesen. Wenn der Parameter `CREATE_OBJECT_PATHS` gesetzt ist, werden die Ausgestaltungslinien soweit möglich zu Pfaden zusammengesetzt. Bei Netzwerken haben Linien mit gleicher Art den Vorrang bei der Pfadverknüpfung.

## 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 Speicher-

aufwand, da die Objekthalte 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 Objekthalte, 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
```

**Dann muss aber sichergestellt sein, dass bei mehreren Sachdatensätzen pro Feature eindeutige Attributnamen vorliegen, der Reader meldet sonst den Fehler:**

*Ueberschreibe Attributinhalt infolge MERGE\_DATA\_ATTRIBUTES*

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 der Picklisten 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
```

Soweit `MERGE_DATA_ATTRIBUTES` gesetzt ist, wird aus den aus den referenzierten Tabellen (Typen S und P) nur der erste Wert gelesen, hier steht i.d.R. der eigentliche Schlüsselwert. Der Attributname wird dann aus dem Attribut der referenzierenden Tabelle erzeugt. Obiges Beispiel würde folgende Featureattribute ergeben:

```
MERGE_DATA_ATTRIBUTES Yes
Id                       = 0815
DN                       = 250
Mat.                    = STZ
```

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

```
Bewohner{0}.Art         = 2
Bewohner{0}.Nummer     = 815
Bewohner{1}.Art         = 1
Bewohner{1}.Nummer     = 91
```

Bei `MERGE_DATA_ATTRIBUTES` entstehen folgende Attribute:

```
Art{0}                  = 2
Art{1}                  = 1
Nummer{0}               = 815
Nummer{1}               = 91
```

Um Listen erfassen zu können, müssen in Geograf Attributnamen besonders gekennzeichnet werden. Hier ein Beispiel:

Unter Geograf wurde ein Attribut mit dem Namen 'listAtt()' vereinbart, die geschwungenen Klammern dienen als Listenkennung. Als Attributtyp ist zwingend s=alphanumerisch zu vergeben. Als Wert können dann mehrere Werte, komma-getrennt erfasst werden:

```
Art()                   = 1012,1025,1088
```

Unter der Readerparametern müssen dann `READ_DATA_LIST`, `DATA_LIST_POSTFIX` `DATA_LIST_DELIM` dementsprechend gesetzt sein:

```
READ_DATA_LIST Yes
DATA_LIST_POSTFIX ()
DATA_LIST_DELIM ,
```

Der Reader erzeugt daraus die Liste:

```
Table.Art{0}           = 1012
Table.Art{1}           = 1025
Table.Art{2}           = 1088
```

oder bei gesetztem `MERGE_DATA_ATTRIBUTES`:

```
MERGE_DATA_ATTRIBUTES Yes
Art{0}                 = 1012
Art{1}                 = 1025
Art{2}                 = 1088
```

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:

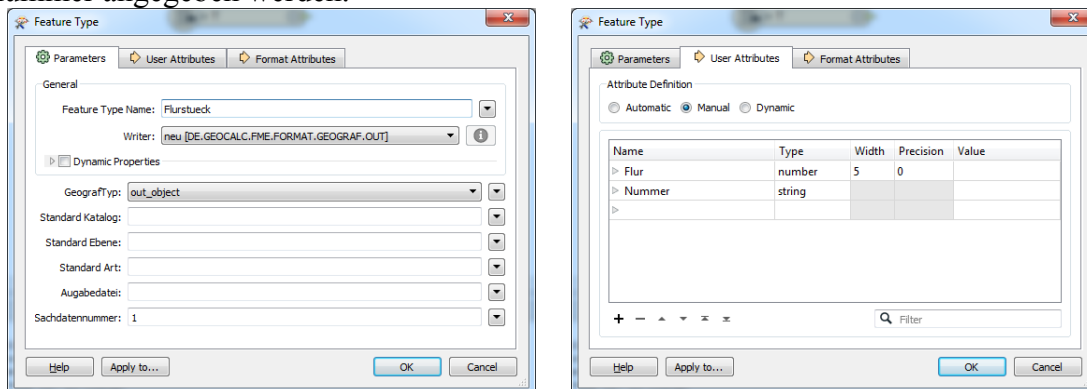
```
OUT_DEF Flurstueck \  
    out_data_id 1 \  
    Flur         number(5) \  
    Nummer      string    \  
    \
```

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.

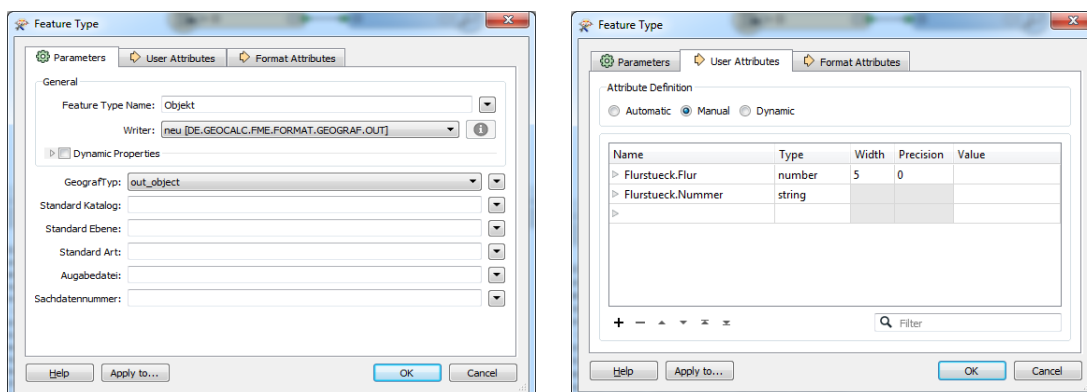
### Einfache Sachdaten:

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.



### Komplexe Sachdaten:

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.

Die Featureattribute müssen selbst nicht diese Langform aufweisen, es reicht als Attributname der letzte Namensteil. Im obigen Beispiel werden die Featureattribute:

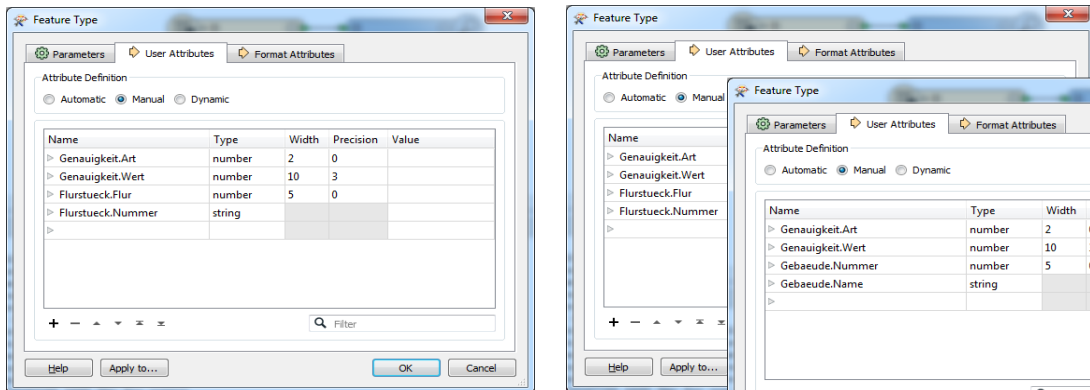
```
Flurstueck.Flur ..= 22  
Flurstueck.Nummer = 126/12
```

oder

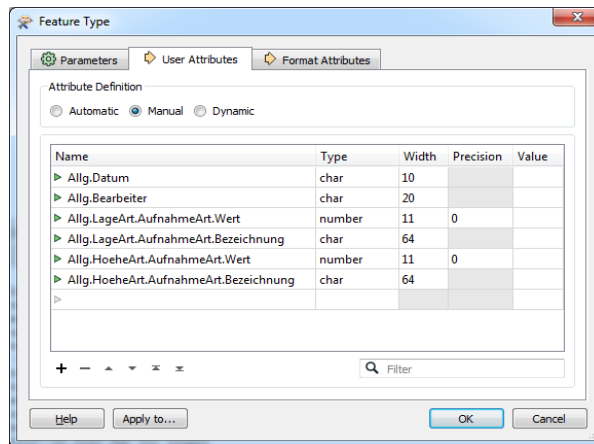


Flur ..= 22  
 Nummer = 126/12

als Sachdatum in die Tabelle Flurstueck geschrieben, Vorrang hat der qualifizierte Attributname.



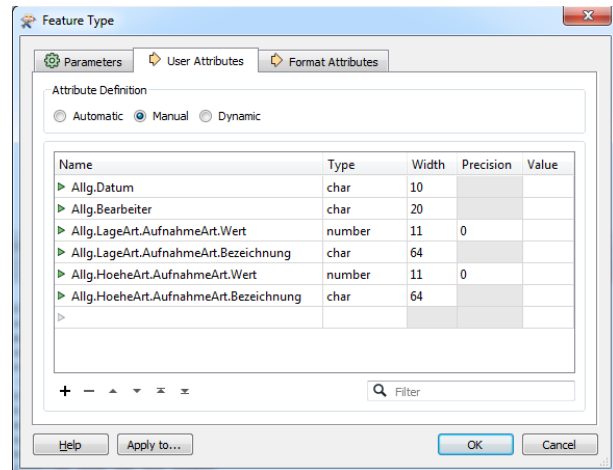
Auf diese Art und Weise können auch SubTabellen oder Picklisten vereinbart werden.



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 unter 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, braucht die Tabelle nur einmalig angelegt zu werden. Dies kann auch über ein AusgabeFeature geschehen, über welches keine Features ausgegeben werden.



Dem Ausgabe-Featuretype müssen dann in der Featuredefinition mit `out_data_id` die Sachdatentabellen mitgeteilt werden, die zum Feature erzeugt werden sollen.

Es können aber auch den zu schreibenden Features individuell die Sachdatentabellen mitgeteilt werden, es ist dann das Featureattribut `out_data` oder eine Liste `out_data{}` zu erzeugen, welche die Tabellennamen enthält. 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 FeatureAttributes kann dann komplett verzichtet werden. Den Features muss dann lediglich mitgeteilt werden, welche Sachdatensätze angelegt werden sollen (siehe Writer-Keywords `TEMPLATE_FILE` und `APPEND_TEMPLATE_DATA`). Dies kann in der Featuredefinition in der Zeile Sachdatennummer(n) geschehen, hier sind die zu benutzenden Sachdatennummern mit Komma getrennt anzugeben. Dem einzelnen zu schreibenden Feature kann aber auch eine Liste `out_data{}` mitgegeben werden, in der die Sachdatennamen anzugeben sind.

### Template:

Für Prozesse, bei denen dauerhaft Sachdaten nach einem festgelegten Schema erzeugt werden sollen, ist es empfehlenswert sich eine Geograf-Template-Datei anzulegen. Insbesondere bei der Benutzung von Subtabellen und Picklisten werden dann nur Werte, die tatsächlich in den Wertetabellen vorgegeben sind. Hier ein Beispiel:

```
*
Typ: GRAFBAT V9
*
FTable: PicTable1,31,P,DBId[da],IntAtt31[d],TextAtt31[s255]
FInfo: PicTable1,DBId:1,IntAtt31:3101,"TextAtt31:Text3101"
FInfo: PicTable1,DBId:2,IntAtt31:3102,"TextAtt31:Text3102"
FInfo: PicTable1,DBId:3,IntAtt31:3103,"TextAtt31:Text3103"
*
FTable: PicTable2,32,P,DBId[da],IntAtt32[d],TextAtt31[s255]
FInfo: PicTable2,DBId:1,IntAtt32:3201,"TextAtt32:Text2301"
FInfo: PicTable2,DBId:2,IntAtt32:3202,"TextAtt32:Text2302"
FInfo: PicTable2,DBId:3,IntAtt32:3203,"TextAtt32:Text2303"
*
FTable: SubTable1,21,S,DBId[da],IntAtt21[d],TextAtt21[s255],ListAtt21() [s255],RefPicAtt21[#32]
FInfo: SubTable1,DBId:1,IntAtt21:2101,"TextAtt21:Text2101","ListAtt21():L2101;L2102;L2103",RefPicAtt21:DBId=1
FInfo: SubTable1,DBId:2,IntAtt21:2102,"TextAtt21:Text2102","ListAtt21():L2111;L2112;L2113",RefPicAtt21:DBId=2
FInfo: SubTable1,DBId:3,IntAtt21:2103,"TextAtt21:Text2103","ListAtt21():L2121;L2122;L2123",RefPicAtt21:DBId=3
*
FTable: SubTable2,22,S,DBId[da],IntAtt22[d],TextAtt22[s255],ListAtt22() [s255],RefPicAtt22[#32]
FInfo: SubTable2,DBId:1,IntAtt22:2201,"TextAtt22:Text2201","ListAtt22():L2201;L2202;L2203",RefPicAtt22:DBId=1
FInfo: SubTable2,DBId:2,IntAtt22:2202,"TextAtt22:Text2202","ListAtt22():L2211;L2212;L2213",RefPicAtt22:DBId=2
FInfo: SubTable2,DBId:3,IntAtt22:2203,"TextAtt22:Text2203","ListAtt22():L2221;L2222;L2223",RefPicAtt22:DBId=3
*
FTable: UniTable,11,E,DBId[da],IntAtt11[d],TextAtt11[s255],ListAtt11() [s255],DocAtt11[n255],RefSubAtt11[#21]
FTypes: UniTable,Dot,"0[450]"
FTable: RelTable,12,M,DBId[da],IntAtt12[d],TextAtt12[s255],ListAtt12() [s255],RefSubAtt12[#22],RefPicAtt12[#31]
FTypes: RelTable,Dot,"0[450]"
```

# 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.
- 23.06.2018 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.  
Der Reader kann für Beschriftungstexte die Höhe aus dem beschrifteten Element ermitteln, soweit sie nicht durch Geograf selbst gesetzt sind.
- 03.07.2018 Der Writer kann jetzt DGMs aus Flächen erzeugen.
- 15.07.2018 Das Lesen von Dokumenten wurde überarbeitet, auch das Schreiben ist jetzt möglich. Es werden jetzt auch die eigentlichen Bilddaten verarbeitet.  
Plotboxen können gelesen und geschrieben werden (FeatureType X).  
Neuere Schraffurtypen wurden eingearbeitet.  
Verknüpfungen mit Oberobjekten werden jetzt auch durch den Writer erzeugt.  
Bisher war nicht immer sicher gestellt, dass alle Referenzen durch den Writer erzeugt werden. Problematisch war z.B. folgende Situation: Text als Objekthalt, der ein Element beschriftet, welches selbst nicht Objekthalt ist. Durch einen erweiterten Sortieralgorithmus wird dieses jetzt sichergestellt.  
Mit der Grafbat-Version 8 und 9 erzeugt Geograf einige neue oder veränderte Attribute. Hierzu wurden Reader und Writer auf den aktuellen Stand gebracht.  
Inzwischen können auch Mengen gelesen und geschrieben werden.
- 10.08.2018 Linien können jetzt auch als out\_pointline geschrieben werden.
- 17.08.2018 Die Modi OBJECT\_PARSER = LINE und AREA stellten im Zusammenhang mit den Schaltern REMOVE\_OBJECT\_GEOM und REMOVE\_OBJECT\_DATA nicht sicher, dass alle Elemente gelesen werden, dies wurde berichtigt.  
Wenn Beschriftungstexte als Objektbestandteile in der Liste out\_elements{} gespeichert wurden, war bisher nicht sichergestellt, dass der Writer die Beziehung zum beschrifteten Element wiederherstellt.  
Der Writer legt jetzt auch leere Sachdatensätze an, wenn keines der vereinbarten Attribute im Feature belegt ist.

Es können bei Schreiben jetzt auch Fehler-Features `out_type=out_error` ausgegeben werden. Diese Features werden ausschließlich in einer Geograf-Fehlerstapeldatei abgelegt.

Der Reader erzeugt jetzt bei fehlerhaft gebildeten Geograf-Objekten eine Fehlermeldung und markiert diese Features mit den Attributen `out_error_name` und `out_error_msg`. Werden so markierte Features wieder über den Writer ausgegeben, erzeugt der Writer zusätzlich eine Geograf-Fehlerstapeldatei.

23.09.2018 Die Readerparameter `CREATE_OBJECT_PATHS`, `RESOLVE_HHK_NAMES`, `READ_DATA_LIST`, `DATA_LIST_POSTFIX` und `DATA_LIST_DELIM` eingeführt.

Das Metafile wurde mit verbesserten GUI-Elementen überarbeitet.

01.11.2018 Der Readerparameter `READ_VOID_DATA` wurde erweitert.

09.11.2018 Der Writer prüft jetzt die Inhalte der Sachdatentabellen auf eindeutige sortierte Ids und doppelte Datensätze