

# GrafBat (OUT) Reader/Writer

## Version 2019

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 1 bis 10**, der Writer schreibt die Daten ausschließlich in der aktuellen **GrafBat-Version 10**.

Inzwischen kann der Reader auch die Daten aus den Geograf-**Katalogen** lesen. Unterstützt werden Artdateien, hier im **ArtBat**-Format ab **Version 7 bis zur aktuellen Version 8**, die Stiftdatei, die Leveldatei, beide liegen als Textdateien vor, und die binäre Farbdatei, in der die Bildschirmfarben festgelegt sind. Die Katalogdaten werden beim Lesen mit den geometrischen Features verknüpft. **Mit der Version 2019 können die Katalogdaten auch als separate Features geladen und verarbeitet werden.**

## 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 Überblick.

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

## Neuerungen der Version 2019

In der Geograf-Version 10 gab es nur kleinere Veränderungen im Datenmodell. So kann jetzt die Breite eines Linienmusters auch als Faktor zur Größe der Artendatei festgelegt werden.

In den Sachdaten kann jetzt auch ein Datum bzw. **Datum/Zeit** als Felddatentyp vereinbart werden.

Das Lesen und Schreiben dieser Neuerungen wurde integriert.

Weiterhin wurde in Reader und Writer die Verarbeitung des **undefinierten Sachdatentyps** <NN> (9999) implementiert.

**Sub- und Picklistentabellen** können jetzt gelesen und geschrieben werden. Damit soll vor allem das Erzeugen von umfangreichen Sachdatenstrukturen erleichtert werden. Auch Tabellen, die einen externen Schlüssel als Grafikverknüpfung benutzen, können so für Geograf generiert werden.

Das Lesen und Verknüpfen der **Katalogdaten** wurde überarbeitet, es können jetzt mehrere Artendateien auch mehrere dort enthaltene Views (maßstabsabhängige Darstellungen) gelesen werden. Der Reader lädt auch auf Wunsch die dort vereinbarten Stift-, Color- und Leveldateien automatisch nach.

Die **Farbinformationen** aus den Katalogdateien können jetzt mit den Features verknüpft werden. Ebenso können die in den Artendateien vereinbarten Textgrößen bei den TextFeatures gesetzt werden. Damit ist es auch möglich, den Textschwerpunkt umzusetzen, so dass die Texte optisch an der Stelle präsentiert werden, an der Sie unter Geograf erscheinen.

Die **Elemente der Katalogdaten** können jetzt auch als geometrieloze Feature gelesen werden.

Mit der Geograf-Version 10 wurde das Fortführungsmodul für alle Anwender freigegeben. Die Verarbeitung der **Elementaktualität** und der externen Schlüssel wurden bei Reader und Writer überarbeitet und damit eine einfache Möglichkeit geschaffen, Datenbanken fortzuführen.

Bei den Writer-Features kann jetzt eine **Sortierreihenfolge** angegeben werden. So kann bei übereinanderliegenden Punkten und Linien gesteuert werden, ob und wie die Verschmelzung zu dem Knoten-Kanten-Modell erfolgen soll.

## 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	No
Generic Color Support	Yes
Spatial Index	Never
Schema Required	No
Transaction Support	No
Geometry Type Attribute	out_type
Geometry Support	

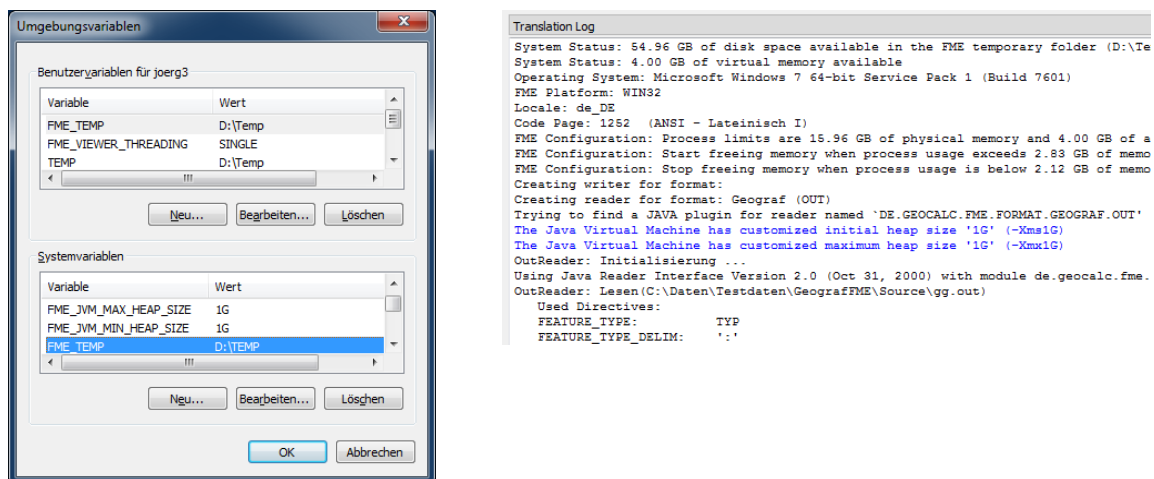
Geometry	Supported	Geometry	Supported
aggregate	yes	polygon	yes
circles	yes	donut polygon	yes
circular arc	yes	line	yes
elliptical arc	no	point	yes
ellipses	no	text	yes
none	yes	3D	yes

## Reader Überblick

Die Speicherung von Geometriedaten erfolgt in Geograf im Gegensatz zu vielen anderen Softwarelösungen punktorientiert, der Geograf-Punkt ist das wesentliche Geometrieelement. So haben Linien keine eigenen Koordinaten, es werden lediglich ein Verweise zum Linienanfangs- und Endpunkt gespeichert. Komplexe Geometrien werden als Objekte gespeichert, die selbst keine Geometrie, sondern eine Verweisliste auf atomaren Geometrieelemente vorhalten. Dieses Geograf eigene Knoten-Kanten-Modell wird vom Reader vollständig aufgelöst, so dass jedes in die FME übertragene Geograf-Element eine eigenständige Geometrie aufweist. Dabei wird die Verweisstruktur in den Feature-Attributen so abgelegt, dass sie beim Zurückschreiben nach Geograf wiederhergestellt werden kann.

Der Geograf-Reader importiert die Geograf-Daten in zwei Schritten. Die OUT-Datei wird zuerst in einem Zwischenspeicher komplett geladen und dabei miteinander verknüpft. 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 Geometrietyp Linien, Flächen oder Aggregate erzeugt. Mit den Reader 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 der 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. Mit der FME 2019 wurden für die 32bit-Version die gesonderten Umgebungsvariablen `FME32_JVM_MIN_HEAP_SIZE` und `FME32_JVM_MAX_HEAP_SIZE` eingeführt.



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

`FME32_JVM_MIN_HEAP_SIZE = 128m`

`FME32_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 Millionen 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 Bezeichnung Dialog	Value	Required Optional
DATASET Source GrafBat File:	Hiermit wird die zu lesende Eingabedatei vorgegeben.	Required
FEATURE_TYPE FeatureTypes als:	<p>Mit diesem Keyword wird die Erzeugung des FeatureTypes gesteuert. Die Einstellung TYP erzeugt folgende FeatureTypes:</p> <p><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  <b>E</b> für Subtabellen, Löschungen und Ummummerierungen  <b>R</b> für Katalogdaten</p> <p>Mit der Einstellung KATALOG wird der Feature Type aus der Katalognummer, mit der Einstellung EBENE aus der Geograf-Ebene erzeugt. Bei der Einstellung ART wird die Geograf-Art benutzt. Bei dem Schalter SACH wird der FeatureType aus dem Namen des Sachdatums erzeugt, mehrere Sachdaten gehen mit Punkt getrennt in den FeatureType ein. Die Namen von relationalen und Subtabellen fließen nicht in den FeatureType ein. Die Einstellungen können mit dem Trennzeichen “:“ in Kombination angegeben werden.</p> <p><b>Value:</b> TYP, KATALOG, EBENE, ART, SACH  <b>Default:</b> TYP  <b>Example:</b> OUT_FEATURE_TYPE TYP ART</p>	Required
FEATURE_TYPE_DELIM Trennzeichen:	<p>Wenn beim Keyword FEATURE_TYPE eine Kombination gesetzt ist, kann hiermit das vom Reader erzeugte Trennzeichen gesteuert werden.</p> <p><b>Value:</b> beliebiges ASCII-Zeichen  <b>Default:</b> “:“</p>	Optional
READ_MODEL Lese Modelle:	<p>Die Transformation von Bitmaps und Dokumenten speichert Geograf in Modellen, die als Standard nicht gelesen werden. Das Lesen der Modelle kann hiermit angeschaltet werden.</p> <p><b>Value:</b> Yes   No  <b>Default:</b> No</p>	Optional
READ_DOCUMENT Lese Dokumente:	<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
READ_RASTER Lese Bilddaten:	<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>	Optional

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
READ_PK0 Lese Punkte Art 0:	<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</p> <p><b>Default:</b> No</p>	Optional
READ_DIGIT Lese Digitalisierung:	<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</p> <p><b>Default:</b> No</p>	Optional
READ_OFFSET Lese Rissdarstellung:	<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</p> <p><b>Default:</b> No</p>	Optional
READ_UUID Lese UUID:	<p>Soweit die UUID aus Geograf in die GrafBat-Datei ausgegeben wird, belegt der Reader ein entsprechendes Attribut <code>out_uuid</code>. Das Attribut <code>out_akt</code>, das Geograf nur zusammen mit der UUID schreibt, wird immer belegt.</p> <p><b>Achtung!</b></p> <p>Werden die Elemente mit UUID über den Writer ausgegeben, müssen diese Schlüssel 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</p> <p><b>Default:</b> Yes</p>	Optional
READ_OPERATION Lese Fortführungsdaten:	<p>Die Aktualität der Geografelemente wird ausgewertet (siehe Attribut <code>out_akt</code>) und daraus das Attribut <code>fme_db_operation</code> gesetzt (UPDATE, DELETE oder INSERT). Wenn das Geograf-Projekt im Fortführungsmodus ausgewertet wurde, werden die Löschmarkierungen gelesen. Die folgenden Parameter <code>READ_UNCHANGE</code>, <code>DELETE_LAYERS</code> und <code>RESOLVE_OPERATION</code> werden nur bei aktivierter <code>READ_OPERATION</code> ausgewertet.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> No</p>	Optional

Keyword Suffix Bezeichnung Dialog	Value	Required Optional										
READ_UNCHANGE Unveränderte Elemente lesen:	<p>Elemente, die unter Geograf gegenüber dem eingelesenen Zustand nicht verändert wurden, können im Geograf-Fortführungsmodus bereits gefiltert werden. Bei der Standardauswertung steht diese Filterung nicht zur Verfügung. Für beide Varianten werden folgende Auswahlmöglichkeiten für Auswertung unveränderter Elemente zur Verfügung gestellt:</p> <p>YES unveränderte Elemente werden gelesen, <code>fme_db_operation</code> bleibt leer.</p> <p>NO unveränderte Elemente werden nicht gelesen</p> <p>UNCHANGE_OPERATION bei unveränderte Elemente wird das Attribut <code>fme_db_operation</code> auf 'unchange' gesetzt. Die erzeugten Features können nach diesem Wert gefiltert werden, beim Schreiben in eine Datenbank würde dieser Wert eine Fehlermeldung erzeugen.</p> <p>UNCHANGE_FEATURE unveränderte Elemente werden über den besonderen Import-FeatureType UNCHANGE gelesen, das Attribut <code>fme_db_operation</code> wird auf 'unchange' gesetzt.</p> <p><b>Value:</b> Yes, No, UNCHANGE_OPERATION, UNCHANGE_FEATURE</p> <p><b>Default:</b> Yes</p>	Optional										
DELETE_LAYERS Löschebenen:	<p>Nur im Geograf-Fortführungsmodus werden Löschmarkierungen in die GrafBat-Datei geschrieben. Oft ist es gewünscht, das bei der Geograf-Bearbeitung die Datenbankelemente nicht physisch gelöscht, sondern nur zur Löschung markiert werden. Eine übliche Variante ist es, hierfür besondere Löschebenen einzurichten.</p> <p>Mit dem hier verfügbaren Filter wird bei den Elementen, die das Filterkriterium erfüllen, das Attribut <code>fme_db_operation</code> auf 'DELETE' gesetzt.</p> <p>Der Filter besteht aus vier mit eckigen Klammern abgegrenzten optionalen Blöcken:</p> <p style="text-align: center;">[kataloge][ebenen][arten][attribute]</p> <p>In den Blöcken [kataloge], [ebenen] und [arten] können ähnlich wie zu den Geograf-Auswahlen die entsprechenden Werte mit Komma getrennt eingetragen werden. Möglich sind:</p> <table style="margin-left: 40px;"> <tr> <td>1,2,3</td> <td>Einzelwerte</td> </tr> <tr> <td>1-50, 500-599</td> <td>Bereiche</td> </tr> <tr> <td>!21, !510-520</td> <td>Negationen</td> </tr> </table> <p>Die Kataloge können nur über deren Katalognummer ausgewählt werden. Sind nur zwei Blöcke angegeben, werden diese als [katalog] und [ebenen] gewertet, besteht nur ein Block, gilt dieser als [ebenen] auf Katalog=0.</p> <p>Beispiel:</p> <p>Löschebenen sind</p> <p>[0,5][200-299,!250][1-9000] die Ebenen 200-249 und 251-299 auf den Katalogen 0 und 5, dabei nur der Artenbereich 1-9000,</p> <p>51-52 die Ebenen 51-52 auf dem Katalog 0.</p> <p>Mit dem Block [attribute] können zu löschende Features über Attribut-inhalte selektiert werden:</p> <table style="margin-left: 40px;"> <tr> <td>ZUSTAND==2000</td> <td>Features mit dem Attributwert gleich 2000</td> </tr> <tr> <td>ZUSTAND!=9000</td> <td>Features mit dem Attributwert ungleich 9000</td> </tr> </table> <p>Mehrere Attributabfragen können mit Und '&amp;' bzw. Oder ' ' verknüpft</p>	1,2,3	Einzelwerte	1-50, 500-599	Bereiche	!21, !510-520	Negationen	ZUSTAND==2000	Features mit dem Attributwert gleich 2000	ZUSTAND!=9000	Features mit dem Attributwert ungleich 9000	Optional
1,2,3	Einzelwerte											
1-50, 500-599	Bereiche											
!21, !510-520	Negationen											
ZUSTAND==2000	Features mit dem Attributwert gleich 2000											
ZUSTAND!=9000	Features mit dem Attributwert ungleich 9000											

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
	<p>werden, das Komma als Trennzeichen gilt ebenso als Oder. Und und Oder dürfen nicht gemischt verwendet werden.</p> <p>Beispiel:</p> <p>[50][ZUSTAND==2000 &amp; AKT!=Neu]</p> <p>Es werden Features der Ebene 50 als gelöscht markiert, bei denen die Attribute ZUSTAND auf '2000' und AKT auf 'Neu' gesetzt sind.</p> <p>Die Filterung erfolgt erst nach der vollständigen Übersetzung des Geograf-Elements. Der jeweilige Attributname ist so zu benennen, wie er je nach den Einstellungen des Sachdatenimports im Feature Type Dialog erscheint.</p> <p><b>Value:</b> formatierte Filtereinstellung</p> <p><b>Default:</b> leer</p>	
<b>RESOLVE_OPERATION</b> Löse Fortführungsattribute auf:	<p>Beim Erzeugen einer GrafBat-Datei mit dem Writer können aus Attributen, die ein Feature eindeutig kennzeichnen, besonders markierte externe Schlüssel erzeugt werden. Anhand dieser externen Schlüssel können insbesondere Löschbefehle wieder dem ursprünglichen Feature zugeordnet werden. Näheres ist im Kapitel <a href="#">Fortführung von Geodatenbanken</a> beschrieben.</p> <p>Ist dieser Schalter auf Yes gesetzt, werden aus den externen Schlüssel die dort enthaltenen Attribute erzeugt und bei Löschungen der ursprüngliche FeatureType wiederhergestellt.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> No</p>	Optional
<b>READ_TABLE</b> Lese Tabellen (Sub-/Pick):	<p>Das Lesen der Sub- und Picktabellen kann eingeschaltet werden. Aus diesen <u>Tabellen</u> werden dann Features ohne Geometrie erzeugt, die als Listennattribute den kompletten Inhalt der Tabellen enthalten.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> No</p>	Optional
<b>OBJECT_PARSER</b> Objektgeometrie als:	<p>Ein Geograf-Objekt ist eine Zusammenfassung von beliebigen geometrischen Einzelementen. Ein Geograf-Objekt weist außer der 2D-Objektcoordinate keine eigene Geometrie auf, es werden lediglich Referenzen auf die Einzelemente gespeichert. In der Regel werden lediglich flächenförmige Objekte, z.B. Flurstücke, erfasst. Es können aber auch beliebige Zusammenhänge zwischen Einzelementen 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.</p> <p>Um diesen Umständen Rechnung zu tragen, bietet der Schalter OBJEKT_PARSER verschiedene Einlesevarianten an:</p> <p>NO            Objekte werden nicht gelesen</p> <p>LINK         Es wird ein fme_point aus der Objektcoordinate erzeugt, bzw. soweit keine Objektcoordinate vorliegt ein Objekt ohne Geometrie. Das Objekt erhält das Attribut out_link_id, welches als Referenzschlüssel für die verlinkten Elemente gilt. Alle Objektbestandteile erhalten die Liste out_object{}.link, welche die Schlüssel der Objekte, in denen es referenziert ist enthält und die Liste out_object{}.linktyp mit der Art der Referenzierung (Definition oder Ausgestaltung).</p>	Optional



Keyword Suffix Bezeichnung Dialog	Value	Required Optional
	<p><b>LIST</b> 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 Geometrietyp (Definition oder Ausgestaltung) der referenzierten Einzelelemente enthalten. Die Objektbestandteile erhalten das Attribut <code>out_link_id</code>, auf das die Objektliste verweist.</p> <p><b>LINE</b> 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 abhängig vom Parameter <code>REMOVE_OBJECT_DATA</code> in einer Liste <code>out_elements()</code> gespeichert bzw. mit <code>out_object().link</code> verlinkt.</p> <p><b>AREA</b> Flächenhaft ausgebildete Geografobjekte werden als Polygon gespeichert, eventuell enthaltene Ausgestaltungslinien und alle anderen Elemente gehen abhängig vom Parameter <code>REMOVE_OBJECT_DATA</code> 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.</p> <p><b>AGGREGATE</b> Es wird ein Aggregat mit allen im Objekt enthaltenen Geometrien erzeugt. Soweit die Definitionslinien Flächen bilden, werden diese als <code>fme_polygon</code> bzw. <code>fme_donut</code> im Aggregat abgelegt.</p> <p>In den Varianten <b>LINE</b>, <b>AREA</b> und <b>AGGREGATE</b> werden Definitions- und Ausgestaltungselemente grundsätzlich getrennt behandelt. Die Geometrie erhält den Namen <code>out_definition</code> oder <code>out_decoration</code>. Bei der Variante <b>AGGREGATE</b> wird bei gemischten Geometrien ein IFMEAggregat mit dem Namen <code>out_composite</code> angelegt, welches zwei Geometriebestandteile mit den Namen <code>out_definition</code> oder <code>out_decoration</code> enthält. Alle Geometriebestandteile erhalten die Geograf-Eigenschaften als Geometrieattribute mit den Bezeichnungen, wie sie bei den Einzelelementen dokumentiert sind.</p> <p>Alle Varianten kann auch der Writer verarbeiten. Die Varianten <b>LINK</b> und <b>LIST</b> können speicherintensiv sein, da hier die Bestandteile teilweise zwischengespeichert werden.</p> <p><b>Value:</b> NO   LINK   LIST   LINE   AREA   AGGREGATE  <b>Default:</b> AGGREGATE</p>	
<b>REMOVE_OBJECT_GEOM</b> Objektgeometrie entfernen:	<p>Geometrische Elemente, die aufgrund des Parameters <code>OBJECT_PARSER</code> bereits in der Geometrie der Objekte aufgegangen sind, werden mit der Standardeinstellung dieses Parameters aus dem übrigen Datenfluss entfernt. Wird der Parameter auf <code>NO</code> gesetzt, werden die Objektbestandteile auch in ihrer elementaren Form gelesen.</p> <p>Soweit <code>OBJECT_PARSER</code> = <code>NO</code>, <code>LINK</code> oder <code>LIST</code> gesetzt ist, setzt der Reader diesen Schalter auf <code>NO</code> und gibt eine entsprechende Meldung aus.</p> <p><b>Value:</b> Yes   No  <b>Default:</b> Yes</p>	Optional

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
REMOVE_OBJECT_DATA Ausgestaltung entfernen:	<p>Dieser Parameter ist relevant, wenn Geograf-Objekte in dem Modus <code>OBJECT_PARSER = LINE</code> oder <code>AREA</code> gelesen werden. Ausgestaltungselemente (Punkte, Linien, Texte, Schraffuren) werden hier nicht in der Geometrie der Objekt-Features verarbeitet.</p> <p>In der Standardeinstellung <code>REMOVE_OBJECT_DATA = YES</code> werden die Ausgestaltungselemente in einer Liste <code>out_elements{}</code> des resultierenden Objekt-Features untergebracht, die Geometrie derer im Attribut <code>out_elements{}.geometry</code> codiert.</p> <p>Ist der <code>REMOVE_OBJECT_DATA</code> auf <code>NO</code> gesetzt, werden die Ausgestaltungselemente in ihrer elementaren Form gelesen und erhalten zusätzlich die Listenattribute <code>out_link{}.link</code> und <code>out_link{}.linktyp</code>, mit denen auf die Objekte verwiesen wird, in dem sie enthalten sind. Das Objekt selbst erhält dann das Attribut <code>out_link_id</code>.</p> <p>Soweit <code>OBJECT_PARSER = NO, LINK</code> oder <code>LIST</code> gesetzt ist, setzt der Reader den Schalter auf <code>NO</code> und gibt eine entsprechende Meldung aus.</p> <p><b>Value:</b> Yes   No <b>Default:</b> Yes</p>	Optional
CREATE_OBJECT_PATHS Objektlinien als Pfad bilden:	<p>In der Standardeinstellung dieses Parameters werden Linienzüge der Ausgestaltungsgeometrie als unsortierte Einzelsegmente in einem Aggregat zusammengefasst.</p> <p>Wenn dieser Schalter auf <code>Yes</code> 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.</p> <p><b>Achtung!</b></p> <p>Polylinien mit Bogenelementen werden dabei zuerst eine Einzelsegmente zerlegt, bevor sie in den neu zu bildenden Pfad eingefügt werden. Dadurch können sich Ids innerhalb der Geometrieattribute doppelnd.</p> <p><b>Value:</b> Yes   No <b>Default:</b> No</p>	Optional
ADD_OBJECT_POINT Objektkoordinate als PIP einfügen:	<p>Soweit mit dem <code>OBJECT_PARSER</code> Aggregate erzeugt werden, kann die Objektkoordinate als <code>fme_point</code> in das Aggregat übernommen werden. Diese wird dann als Punktgeometrie mit dem Namen <code>out_snap</code> angelegt.</p> <p><b>Value:</b> Yes   No <b>Default:</b> No</p>	Optional
ARC_TO_POLYLINE Bögen als Bogenpolygon:	<p>Je nach Erzeugung unter Geograf generiert der Reader Bögen nach dem FME-Geometriemodell. Diese werden so auch in die Umringspolygone der Objekt- und Schraffurflächen übernommen.</p> <p>Ist dieser Schalter auf <code>Yes</code> gesetzt, wandelt der Reader alle Bögen in Bogenpolygone.</p> <p><b>Value:</b> Yes   No <b>Default:</b> No</p>	Optional
SPLINE_PHANTOM_POINTS Spline interpolieren:	<p>Geograf-Splines werden als <code>fme_line</code> übersetzt. Als Zwischenpunkte werden die Koordinaten der Splinestützpunkte übernommen. Um eine annähernd gleiche Ausrundung zu erhalten, kann der Schalter auf <code>YES</code> gesetzt werden. Entsprechend der Angabe <code>SPLINE_SEGMENT_POINTS</code> werden dann weitere Zwischenpunkte interpoliert.</p> <p><b>Value:</b> Yes   No <b>Default:</b> No</p>	Optional

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
SPLINE_SEGMENT_POINTS Segmentpunkte:	Gibt die Anzahl der zu interpolierenden Zwischenpunkte bei der Splineinterpolation an. Circa 10 Zwischenpunkte werden benötigt, um eine Darstellung analog zu Geograf zu erhalten.  <b>Value:</b> integer <b>Default:</b> 0	Optional
TEXT_PARSER Beschriftungstext als:	Mit der Beschriftungsfunktion hält Geograf eine Verbindung des Beschriftungstextes mit dem beschrifteten Element vor. Der Schalter TEXT_PARSER bietet folgende Einlesevarianten an:  NO            Beschriftungen werden nicht besonders gelesen LINK        Ein Beschriftungstext erhält das Attribut <code>out_text_link</code> , welches den Verweis zu beschrifteten Element enthält. Dieses ist mit dem Attribut <code>out_text_id</code> gekennzeichnet. Bei Texten, die nicht Beschriftung sind, ist in <code>out_text_link</code> der Wert '0' abgelegt. LIST        Die beschrifteten Elemente erhalten eine Liste <code>out_text{}</code> , die alle Attribute der Beschriftungstexte enthält. Die Texte werden als solche erzeugt, ohne dass die Verbindung zum Beschriftungselement gespeichert wird. EXCLUSIVE   Wie in der Variante LIST erhält das beschriftete Element eine Liste mit den Textattributen. Die Texte werden dabei selbst nicht als Features erzeugt.  Die Varianten LINK und EXCLUSIVE werden auch durch den Writer unterstützt, bei der Variante LIST würden die Texte doppelt abgelegt werden.  <b>Value:</b> No   LINK   LIST   EXCLUSIVE <b>Default:</b> LINK	Optional
STANDARD_TEXT_SIZE Standard Textgröße:	Geograf speichert die Textgröße nur dann, wenn sie individuell gesetzt wird. In der Regel wird die Textgröße mit '0.0' gespeichert, bei der Darstellung wird sie aus der Artendatei entnommen und in den Präsentationsmaßstab umgerechnet.  Mit diesem Schalter kann eine Größe vorgegeben werden, die dann für alle Texte benutzt wird, die keine individuelle Textgröße aufweisen. Die ungleich '0.0' gelesene oder die hiermit erzeugte Textgröße wird in dem Attribut <code>out_text_size</code> abgelegt, der originale Geografwert (auch '0.0') bleibt im Attribute <code>out_size</code> erhalten und wird vom Writer vorrangig verarbeitet.  <b>Value:</b> float <b>Default:</b> 2.0	Optional
TEXT_SCALE Textgrößenfaktor:	Geograf wertet die gespeicherte Textgröße als Versalhöhe (Höhe Großbuchstabe), hier überschreiten die Unterlänge (wie bei den Zeichen 'gjq') und die Umlautpunkte der Versale bereits diese Größenangabe. Typischerweise schließt die Angabe zur Fontgröße diese Bereiche mit ein. Daher müssen die Textgrößen umgerechnet werden, damit die Texte in etwa gleicher Größe dargestellt werden.  Der Wert FME_DEFAULT ist so eingestellt, dass die Textgrößen im DataInspector annähernd identisch zu Geograf erscheinen. Es kann auch ein individueller Faktor angegeben werden.  Die Anwendung ist nur angebracht, wenn die <u>Katalogdaten</u> und damit die Angaben zur verwendeten Textgröße geladen werden.  <b>Value:</b> No, FME_DEFAULT oder float <b>Default:</b> No	Optional

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
RESET_TEXT_POSITION Textposition zurücksetzen:	<p>Unter Geograf kann angegeben werden, auf welchem Eckpunkt des Textrahmens die Textkoordinate gesetzt ist. Diese Angabe wird unter <code>out_position</code> abgelegt. Nicht alle Systeme können eine solche Angabe auswerten, so dass die Texte mit individuellen Textschwerpunkt an falscher Position präsentiert werden. Wenn die Artendateien und damit die Textgrößen gelesen werden, kann der Reader die individuellen Schwerpunkte auf die Standardposition (links-unten) umrechnen.</p> <p>Auch dadurch, dass Geograf die Unterlänge (Descent) der Schrift nicht in die Größenangabe mit einschließt, ergibt sich eventuell eine Abweichung in der Präsentation der Texte.</p> <p>Dabei wird die Textkoordinate verändert. Da die Angaben zur Schriftgröße oft unterschiedlich interpretiert werden, kann die Umrechnung allerdings nur annähernd erfolgen. Folgende Varianten stehen zur Verfügung:</p> <p><b>DESCENT</b> Es erfolgt eine Korrektur aufgrund der Unterlänge, die Textkoordinate wird vertikal versetzt.</p> <p><b>POSITION</b> Es erfolgt eine Korrektur aufgrund der Geografangabe des Textschwerpunktes, die Textkoordinate wird vertikal und horizontal verändert. Der Inhalt des Attributes <code>out_position</code> wird auf den Standardwert '0' zurückgesetzt.</p> <p><b>DESCENT_POSITION</b> Beide Verschiebungen erfolgen.</p> <p><b>Value:</b> No, DESCENT, POSITION, DESCENT_POSITION</p> <p><b>Default:</b> No</p>	Optional
MULTI_LINE_TEXT_DELIM Zeilen-Trennzeichen:	<p>Mehrzeilige Texte werden auch in dem Attribut <code>out_text_string</code> gespeichert. Dabei wird ein Zeilenumbruch '\n' als Standardtrennzeichen verwendet. Das Trennzeichen kann mit dem Schalter MULTI_LINE_TEXT_DELIM gesondert gesetzt werden.</p> <p><b>Value:</b> beliebiges ASCII-Zeichen</p> <p><b>Default:</b> '\n'</p>	Optional
MULTI_LINE_TEXT Mehrzeilentexte als Liste:	<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</p> <p><b>Default:</b> No</p>	Optional
TEXT_AS_3D Beschriftungstext in 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.</p> <p>Bei auf Yes gesetztem 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. Die Koordinate des Textbezugs hat dabei Vorrang.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> No</p>	Optional

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
MERGE_DATA_ATTRIBUTES Vereinige Sachdatenattribute	<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.</p> <p>Für Picklisten wird hier dann nur die erste Tabellenspalte als Attribut übernommen, dort sollte der eigentliche Schlüsselwert gespeichert werden und unter dem Attributname gespeichert, welches auf die Pickliste verweist. 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
READ_VOID_DATA Erzeuge leere Sachdatenattribute:	<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', '0.0' oder "" belegt.</p> <p><b>Value:</b> No, Null, Default <b>Default:</b> Default</p>	Optional
RESOLVE_HHK_NAMES Entferne 'HHK_'-Attributepräfix:	<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>	Optional
DATA_ATTRIBUTE_PREFIX Präfix Sachdaten:	<p>Die Attributnamen der Sachdaten können ein zusätzliches Präfix erhalten.</p> <p><b>Value:</b> string <b>Default:</b> ""</p>	Optional
READ_DATA_LIST Erkenne Listenattribute:	<p>Wenn dieser Schalter gesetzt ist, werden mit <code>DATA_LIST_POSTFIX</code> und <code>DATA_LIST_DELIM</code> besonders gekennzeichnete Sachdatenattribute als Liste aufgelöst.</p> <p><b>Value:</b> Yes   No <b>Default:</b> Yes</p>	Optional
DATA_LIST_POSTFIX Postfix Listenattribut:	<p>Hiermit können Erweiterungen von Attributnamen vereinbart werden, die der Reader als Liste interpretiert. Der Attributinhalt wird dann mittels der Trennzeichen <code>DATA_LIST_DELIM</code> 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>	Optional

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
DATA_LIST_DELIM Trennzeichen:	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)'	Optional
ART_SETTING_FILE Artbat Datei:	Hiermit werden die zusätzlich zu lesenden Artdateien vorgegeben. Die Artdateien müssen im ArtBat-Format ab Version 7 vorliegen. Die Informationen der Artdatei werden vom Reader an die Features angehängt. <b>Mit Angabe eines Sternchens '*' kann angezeigt werden, dass die in der GrafBat-Datei unter BLATT und ARTPLUS angegebenen Artdateien gelesen werden sollen. Dazu müssen die Artdateien im ArtBat-Format mit der Extension '.out' vorliegen:</b>  <ul style="list-style-type: none"> <li>• an dem in der GrafBat-Datei angegebenen absoluten oder relativen Pfad,</li> <li>• in dem Verzeichnis der GrafBat-Datei,</li> <li>• oder in einem Verzeichnis, welches mit der System-Umgebungsvariablen FME_GRAFBAT_KATALOG_DIR angezeigt werden kann.</li> </ul> <b>Value:</b> string <b>Default:</b> ""	Optional
PEN_SETTING_FILE Stift Datei:	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>Mit der Angabe eines Sternchens '*' wird angezeigt, dass der Reader die in den Artdateien vorgegebenen Stiftdateien lesen soll.</b>  <b>Value:</b> string <b>Default:</b> "*"	Optional
COLOR_SETTING_FILE Color Datei:	Hiermit wird die zusätzlich zu lesende Farbdatei vorgegeben. Die binär vorliegende Einstellungsdatei enthält die Farbinformationen zur Bildschirmdarstellung der Features. <b>Mit der Angabe eines Sternchens '*' wird angezeigt, dass der Reader die in den Artdateien vorgegebenen Colordateien lesen soll.</b>  <b>Value:</b> string <b>Default:</b> "*"	Optional
LEVEL_SETTING_FILE Stufen Datei:	Hiermit wird die zusätzlich zu lesende Stufenfile vorgegeben. Die Stufenfile 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>Mit der Angabe eines Sternchens '*' wird angezeigt, dass der Reader die in den Artdateien vorgegebenen Stufenfile lesen soll.</b>  <b>Value:</b> string <b>Default:</b> "*"	Optional
ART_KEEP_ATTRIBUTES Artenattribute:	Die sehr umfangreichen Attribute der Katalogdateien können mit diesem Parameter auf die notwendigen eingegrenzt werden. Die Angabe eines beliebigen Zeichens bewirkt, dass die Informationen der Katalogdateien ausgewertet werden (Textgrößen, Farben), aber keines der art_* Attribute gespeichert wird.  <b>Value:</b> string (Komma- bzw. Leerzeichen getrennte Liste der art_-Attribute) <b>Default:</b> ""	Optional

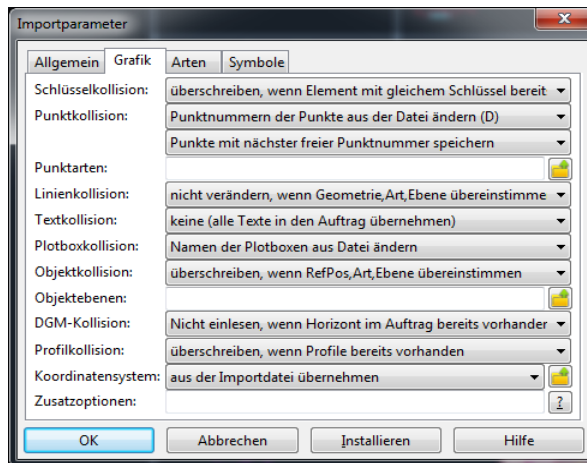
Keyword Suffix Bezeichnung Dialog	Value	Required Optional
VIEW_SCALE Präsentationsmaßstab:	<p>Die Artendateien können in sogenannten Views für verschiedene Präsentationsmaßstäbe unterschiedliche Einstellungen je Art speichern, Geograf ruft diese je nach aktuell eingestellten Maßstab ab. Der Reader verhält sich hier ebenso, als Standard verwendet er den Maßstab der Blatteinstellungen aus der GrafBat-Datei.</p> <p>Mit diesem Parameter kann ein individueller Maßstab angegeben werden, die Arten werden dementsprechend abgerufen.</p> <p><b>Value:</b> FROM_DATA, integer</p> <p><b>Default:</b> FROM_DATA (Maßstab aus Blatteinstellungen)</p>	Optional
COLOR_SOURCE Farbe erzeugen aus:	<p>Mit dieser Einstellung werden optional die FeatureAttribute <code>fme_color</code> und <code>fme_fill_color</code> aus den Informationen der Katalogdateien gesetzt. Folgende Varianten stehen zur Verfügung:</p> <p>No Die FME-Farbattribute werden nicht erzeugt.</p> <p>SCREEN_ART Es wird die Bildschirmfarbe der Art benutzt</p> <p>SCREEN_EBENE Es wird die Bildschirmfarbe der Ebene benutzt</p> <p>SCREEN_KATALOG Es wird die Katalogfarbe benutzt</p> <p>PLOT_ART Es wird die Plotfarbe der Art benutzt</p> <p>PLOT_EBENE Es wird die Plotfarbe der Ebene benutzt</p> <p><b>Value:</b> No, SCREEN_ART, SCREEN_EBENE, SCREEN_KATALOG, PLOT_ART, PLOT_EBENE</p> <p><b>Default:</b> No</p>	Optional
COLOR_INDIVIDUAL individuelle Farbe erzeugen aus:	<p>Die FeatureAttribute <code>fme_color</code> und <code>fme_fill_color</code> werden aus den individuellen Einstellungen des Geograf-Elements oder aus der Plotfarbe einer Menge erzeugt. Für die Plotfarbe müssen die Arten- und Stiftdateien gelesen werden. Folgende Varianten stehen zur Verfügung:</p> <p>No Die FME-Farbattribute werden nicht erzeugt</p> <p>FROM_ELEM Die individuelle Elementfarbe wird benutzt</p> <p>FROM_MENGE Die Plotfarbe der Menge wird benutzt</p> <p>FROM_ELEM_MENGE beide, Elementfarbe hat Vorrang</p> <p>FROM_MENGE_ELEM beide, Mengenfärbung hat Vorrang</p> <p><b>Value:</b> No, FROM_ELEM, FROM_MENGE, FROM_ELEM_MENGE, FROM_MENGE_ELEM</p> <p><b>Default:</b> No</p>	Optional
READ_RULES_AS_FEATURE Lese Katalogdaten als Feature:	<p>Die Elemente der <code>Katalogdaten</code> können zusätzlich als Features gelesen werden. Sie sind vom Type <code>out_rule</code> und besitzen keine Geometrie.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> No</p>	Optional

## Writer Überblick

Geograf ermöglicht die Speicherung von unterschiedlichen Geometrie- und Datenmodellen, die so auch von dem Writer unterstützt werden. Um alle Vorteile der Software nutzen zu können, sollte möglichst das strenge Geograf eigene Knoten-Kanten-Modell aufgebaut werden.

Der Writer verfügt über wesentlich weniger Parameter, dennoch werden alle vom Reader erzeugten Varianten der Datenmodellierung unterstützt. Diese werden an der Attributstruktur der zu schreibenden Features erkannt.

Der Writer baut die Verweisstruktur der Elemente bei Linien, Objekten und Beschriftungen größtenteils eigenständig auf. Die endgültige Entscheidung, ob und welche Punkte und Kanten verschmolzen werden, kann mit den Geograf-Importparametern getroffen werden.



Da der Punkt- und Linienabgleich unter Umständen abhängig von den FeatureTypes ist, bietet der Writer die Möglichkeit, Features zu sortieren. Die Sortierung ist ein wesentliches Entscheidungskriterium für die Verschmelzung von Punkten und Linien beim Import nach Geograf.

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 Objektlinien.

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

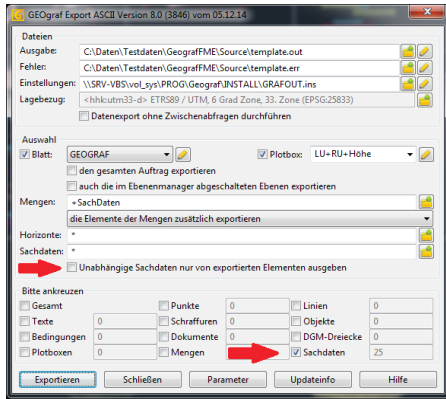
Der Writer ermöglicht auch das Schreiben von individuellen Attributen, die Geograf in Sachdatensätzen verwaltet. Geograf ermöglicht das Anlegen von mehreren Sachdatensätzen pro Feature. Auch diese können erzeugt werden, hierzu später mehr im Kapitel Sachdaten.

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

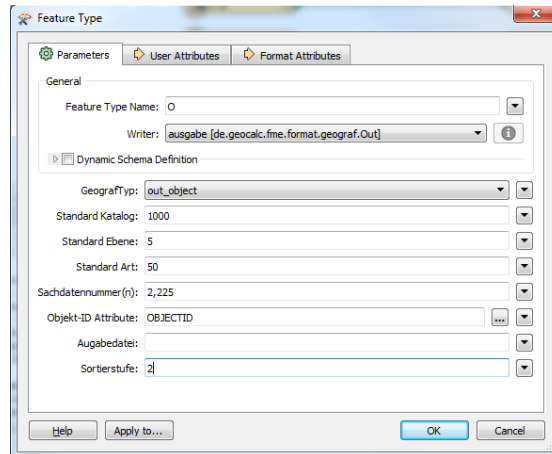
Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
DATASET	Hiermit wird die zu schreibende Ausgabedatei vorgegeben.	Required
TEMPLATE_FILE Template GrafBat Datei:	<p>Es können dem Writer unter Geograf bereits definierte Sachdaten vorgegeben werden. Dann kann auf die Definition der User-Attribute verzichtet werden.</p> <p>Insbesondere bei mehreren oder komplexen Sachdatensätzen mit Subtabellen und Picklisten ist dies eine einfache Möglichkeit, feste Datenschemata einzuhalten.</p> <p>Ein Template wird aus Geograf ausgegeben:</p>	Optional
		
	<p><b>Value:</b> string</p> <p><b>Default:</b> ""</p>	
ATTDEFS_OVERWRITE Definition überschreibend:	<p>Mit dem Schalter kann gesteuert werden, ob die formatspezifischen Attribute <code>out_katalog</code>, <code>out_ebene</code> und <code>out_art</code> der DEF-Zeile eventuell vorhandene FeatureAttribute überschreiben.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> Yes</p>	Optional
MERGE_COORDS Koordinaten verschmelzen:	<p>Der Writer erzeugt automatisch Geograf-Punkte an den Vertex von Linien und Objektflächen. 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.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> Yes</p>	Optional
CREATE_OBJECT_COO Objektkoordinate erzeugen:	<p>Der Writer erzeugt eine Objektkoordinate aus der Geometrie, wenn diese nicht mit <code>out_x</code>, <code>out_y</code> individuell vorgegeben ist. Der Algorithmus ist nicht identisch mit der automatischen Generierung unter Geograf, so erzeugt er z.B. die Objektkoordinate auf der Mitte eines Linienzuges.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> Yes</p>	Optional

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional
APPEND_TEMPLATE_DATA Template-Subtabellen ergänzen:	<p>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.</p> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> No</p>	Optional
CREATE_VOID_DATA Erzeuge leere Sachdatensätze:	<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> <p><b>Value:</b> Yes   No</p> <p><b>Default:</b> No</p>	Optional
DATA_LIST_POSTFIX Postfix Listenattribut:	<p>Der Writer erzeugt aus Listenattributen ein einzelnes Geografattribut, der Attributname wird dann mit dieser Kennung erweitert. Die Daten aus der Attributliste werden dann, separiert mit DATA_LIST_DELIM, als ein einzelner Attributwert geschrieben.</p> <p>Mit dem Parameter <u>READ_DATA_LIST</u> können so erzeugte Attribute beim Lesen wieder aufgelöst werden.</p> <p><b>Value:</b> string</p> <p><b>Default:</b> () (runde Klammern)</p>	Optional
DATA_LIST_DELIM Listen-Trennzeichen:	<p>Das Trennzeichen, mit dem Listenattribute zu einen Geografattributwert zusammengefügt werden. Es sollte mit dem Readerparameter <u>DATA_LIST_DELIM</u> korrespondieren.</p> <p><b>Value:</b> char(1)</p> <p><b>Default:</b> ';' (Semikolon)</p>	Optional
DEF	<p>Für jeden FeatureType können die Inhalte der formatspezifischen Attribute vorgegeben werden. Diese werden in der DEF-Zeile aufgelistet.</p> <p>Beispiel:</p> <pre>OUT_DEF Flurstueck \   out_type          out_object \   out_katalog       1000      \   out_ebene         5         \   out_art           50        \   out_data_id       2,225     \   out_match_attr    OBJECTID  \   out_file          \   out_sort          2</pre> <p>Für alle Attribute, die nicht mit dem Formatkennzeichen out_ beginnen, wird ein Sachdatensatz erzeugt. Das Sachdatum erhält den Namen des FeatureTypes.</p> <p>Die Sachdatennummer wird automatisch generiert, soweit sie nicht mit dem DEF-Attribut <u>out_data_id</u> vorgegeben wird. Geograf ermöglicht eine Vergabe in dem Bereich von 1-9998.</p> <p>Soweit ein Template bei der Ausgabe verwendet wird, brauchen die Sachdatenattribute nicht definiert werden. Unter <u>out_data_id</u> müssen dann die Nummern der im Template bestehenden Sachdatentabellen angegeben werden, die zu dem Feature erzeugt werden sollen.</p> <p><b>Wenn mit <u>out_match_attr</u> Attributnamen angegeben werden, werden aus diesen und ihren Inhalten speziell gekennzeichnete externe Schlüssel erzeugt, die beim Reimport in die FME wiederhergestellt werden. Näheres dazu im Kapitel: <u>Fortführung von Geodatenbanken</u>.</b></p> <p>Wird bei out_file ein Dateiname angegeben, wird der FeatureType in diese Datei umgeleitet.</p>	Required

Keyword Suffix	Value	Required
Bezeichnung Dialog		Optional

Mit `out_sort` kann eine Sortierreihenfolge angegeben werden. Dies ist dann relevant, wenn beim Einlesen in Geograf sich überlagernde Punkte und Linien verschmolzen werden sollen. Die Reihenfolge entscheidet dann, welche Art die führende ist. Näheres dazu ist im Kapitel [Knoten-Kanten-Modell](#) beschrieben.

In der Workbench werden die Feature-Attribute über den Reiter 'Parameters' vergeben:



Folgende Attributtypen sind für die User-Attribute vorgesehen:

- `boolean` entspricht dem GeografAttributtyp `b=boolscher Wert`
- `number(w,0)` entspricht dem GeografAttributtyp `d=Ganzzahl`
- `number(w,p)` entspricht dem GeografAttributtyp `f=Gleitkommazahl`
- `buffer` entspricht dem GeografAttributtyp `s=alphanumerisch` (ohne Längenbegrenzung, Memofeld)
- `string` entspricht dem GeografAttributtyp `s=alphanumerisch` (ohne Längenbegrenzung, Memofeld)
- `char(w)` entspricht dem GeografAttributtyp `s=alphanumerisch` (mit Längenbegrenzung)
- `date` entspricht dem GeografAttributtyp `t=Datum/Zeit` (ohne Längenbegrenzung)
- `file` entspricht dem GeografAttributtyp `n=alphanumerisch` (ohne Längenbegrenzung)

Beispiel:

```

OUT_DEF Flurstueck \
  out_type          out_object \
  out_katalog       1000 \
  out_ebene         5 \
  out_art           50 \
  out_data_id       2,225 \
  out_match_attr    OBJECTID \
  out_file          \
  out_sort          2 \
  AREA             number(18,5) \
  KENNZAHl         number(10,0) \
  ZONE             char(20) \
  USED             boolean \
  USER_NAME        buffer \
  FILE_NAME        file \
  BEMERKUNG        string

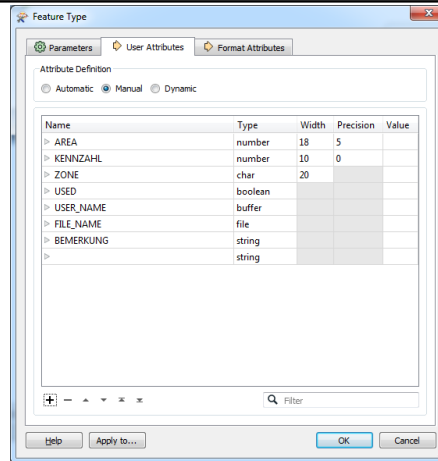
```

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

**Keyword Suffix**  
Bezeichnung Dialog

**Value**

**Required**  
**Optional**



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

# Feature Repräsentation

Geograf speichert in der GrafBat-Datei neben den allgemeinen Projekteinstellungen alle zu einem Projekt gehörigen Geometriedaten. Diese sind in dem folgenden Kapitel [Geometrie Features](#) beschrieben.

Weiterhin werden in der GrafBat-Datei die Beschreibung der Sachdaten und einige wenige Steuerbefehle gespeichert. Die daraus resultierenden Features werden im Kapitel [Features ohne Geometrie](#) beschrieben.

Die Regeln zur Darstellung der Grafikelemente auf dem Bildschirm und für den Druck werden in den [Katalogdaten](#) abgelegt. Die binär gespeicherten Artendateien können aus Geograf in ArtBat-Dateien ausgelesen werden. Die Definitionen der Plotterstifte (Farben und Strichbreiten) und die Regeln zur Reihenfolge der Plotbefehle werden in Textdateien abgelegt. Die Bildschirmfarben legt Geograf in einer binären Datei ab. Diese Katalog-Dateien kann der Reader lesen, die grundlegenden Einstellungen werden dabei den Geometrie-Features zugeordnet.

## Geometrie Features

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 in Geograf als Dreiecksnetze gespeichert.

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

Die Geometrieelemente einer GrafBat-Datei besitzen einen eindeutigen Schlüssel, über den sie untereinander referenziert sind. Solche Referenzen löst Geograf beim Einlesen einer GrafBat-Datei nur auf, wenn verweisende Elemente hinter dem Verweiselement stehen. Aus diesem Grund ist die Reihenfolge der Elemente in einer GrafBat-Datei nicht beliebig. Der Writer schreibt die Daten in der notwendigen Abfolge, muss hierzu aber teilweise Elemente zwischenspeichern.

## allgemeine Attribute

Die im Folgenden beschriebenen Attribute können bei allen geometrischen Features gesetzt sein.

Attribute Name	Content
out_type	<p>Dieses Attribut speichert den Geograf-Typ des Elements.</p> <p>Der Writer wertet vorrangig ein individuell gesetztes Featureattribut aus, es kann aber auch je FeatureType festgelegt werden. Ist dieses Attribut nicht vorgegeben, erzeugt der Writer ein Element, welches dem FME-Geometrietyp entspricht.</p> <p><b>Value:</b> <a href="#">out_point</a>, <a href="#">out_line</a>, <a href="#">out_polyline</a>, <a href="#">out_pointline</a>, <a href="#">out_arc</a>, <a href="#">out_circle</a>, <a href="#">out_clothoid</a>, <a href="#">out_spline</a>, <a href="#">out_text</a>, <a href="#">out_multi_text</a>, <a href="#">out_schraffur</a>, <a href="#">out_boeschung</a>, <a href="#">out_object</a>, <a href="#">out_model</a>, <a href="#">out_doc</a>, <a href="#">out_raster</a>, <a href="#">out_dgm</a>, <a href="#">out_box</a></p>
out_key	<p>Der Elementschlüssel ist je GrafBat-Datei eindeutig. Der 32bit-Schlüssel setzt sich aus einer Geometrikennung und einer laufenden Nummer zusammen. Der Reader speichert den Schlüssel in dem Attribut <code>out_key</code>. Der Writer wertet dieses Attribut nicht aus. Dort, wo Referenzen der Elemente untereinander abgelegt werden (Beschriftungen und Objekthalte), wird dieser Schlüssel in gesonderten Attributen abgelegt (<a href="#">out_text_id</a>, <a href="#">out_link_id</a>).</p> <p><b>Value:</b> Integer</p> <p><b>Default:</b> No default</p>
out_file_key	<p>Der eindeutige Elementschlüssel, zusammen mit der Elementkennung in der Formatierung der GrafBat-Datei (nur Reader)</p> <p><b>Value:</b> char(12)</p> <p><b>Default:</b> No default</p>

Attribute Name	Content
out_extkey out_extkey{}	<p>Der externe Schlüssel des Elements</p> <p>Das Attribut wird nur angelegt, wenn die Quelldatei grundsätzlich externe Schlüssel enthält. Bei Elementen ohne externen Schlüssel ist das Attribut dann mit &lt;null&gt; belegt.</p> <p>Externe Schlüssel müssen im Geograf-Projekt nicht eindeutig sein, eine dementsprechende Prüfung erfolgt weder durch den Writer, noch beim Geograf-Import.</p> <p>Geograf ermöglicht auch die Speicherung von mehreren externen Schlüsseln je Element, die Anwendung dürfte allerdings nur selten vorkommen. In diesem Fall speichert der Reader alle vorkommenden Schlüssel in der Liste out_extkey{}, zusätzlich wird der erste Schlüssel unter out_extkey gespeichert. Die Liste ist bei den FeatureTypes nicht unter 'Format Attributes' aufgeführt, sie muss bei Bedarf mit dem AttributeExposer offen gelegt werden. Der Writer wertet Liste und Attribut aus.</p> <p>Externe Schlüssel dürfen keine Kommas enthalten.</p> <p><b>Value:</b> char(30) <b>Default:</b> &lt;null&gt;</p>
out_katalog	<p>Der Katalog, auf der das Element liegt</p> <p><b>Range:</b> 0-9999 <b>Default:</b> 0</p>
out_ebene	<p>Die Ebene, auf der das Element liegt</p> <p><b>Range:</b> 0-9999 <b>Default:</b> 0</p>
out_art	<p>Die Art des Elements</p> <p><b>Range:</b> 0-99999 <b>Default:</b> 0</p>
out_akt	<p>Geograf speichert bei Fortführungsaufträgen die Aktualität eines Elements mit einer Ganzzahl. Je nach Ausgabemodus wird die Aktualität zusammen mit der UUID oder bei externen Schlüsseln gespeichert.</p> <p>0 = Element ist unverändert 1 = Element wurde geändert 2 = Element wurde hinzugefügt</p> <p>In älteren GrafBat-Formaten wurde in diesem Attribut ein 16bit-Wert abgelegt, der bei der ALK-Fortführung die Aktualität zum externen Schlüssel speicherte.</p> <p><b>Range:</b> 0-2 <b>Default:</b> 0</p>
out_uuid_flag	<p>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.</p> <p>Geograf setzt in der aktuellen Version das Flag nur, wenn die Struktur oder die Attribute des Features geändert werden (z.B. Linie teilen, Art ändern). Das Verschieben von Punkten samt der anhängenden Linien bewirkt noch kein Setzen des Flags.</p> <p><b>Value:</b> boolean <b>Default:</b> No</p> <p>durch out_akt ersetzt</p>

Attribute Name	Content
out_data{ out_data	<p>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. Soweit je Feature nur ein Sachdatensatz erzeugt werden soll, reicht das Setzen des Attributes out_data (nur Writer).</p> <p>Wenn bei der Writer-Featurdefinition die Sachdatennummer(n) angegeben werden, ist dieses individuelle Listenattribut nicht notwendig.</p> <p><b>Value:</b> string <b>Default:</b> No default</p>
out_menge{ out_menge	<p>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.</p> <p><b>Value:</b> char(50) <b>Default:</b> No default</p>
out_error_name out_error_msg	<p>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.</p> <p>Trifft der Writer diese optionalen Attribute an, werden die so markierten Features zusätzlich in eine Fehlerstapeldatei abgelegt, die im Geograf-Fehlermanager abgefragt werden kann.</p> <p><b>Value:</b> string <b>Default:</b> No default</p>
<b>Attribute bei <u>READ_UUID = Yes</u></b>	
out_uuid	<p>Beim Export aus Geograf können UUIDs erzeugt werden. Diese werden gelesen, wenn der Schalter <u>READ_UUID</u> gesetzt ist. Sie setzen sich zusammen aus dem hexadezimalen Wert des Elementschlüssels (<u>out_key</u>) und einer projektbezogenen UUID. Das Attribut wird nur angelegt, wenn die Quelldatei generell UUIDs enthält. Bei Elementen ohne UUID ist das Attribut dann mit &lt;null&gt; belegt.</p> <p>UUIDs müssen je Geograf-Projekt eindeutig sein, beim Geograf-Import werden Elemente mit identischer UUID überschrieben. Der Writer prüft die Eindeutigkeit der Elemente nicht.</p> <p><b>Value:</b> char(40) <b>Default:</b> No default</p>
<b>Attribute bei <u>READ_OPERATION = Yes</u></b>	
fme_db_operation	<p>Dieses Attribut wird angelegt, wenn der Reader-Parameter <u>READ_OPERATION</u> gesetzt ist. Die Belegung wird mit den Parametern <u>READ_UNCHANGE</u>, <u>DELETE_LAYERS</u> und <u>RESOLVE_OPERATION</u> gesteuert, eine ausführliche Beschreibung erfolgt im Kapitel <u>Fortführung von Geodatenbanken</u>.</p> <p><b>Value:</b> char(8), DELETE, UPDATE, INSERT, unchange, "" <b>Default:</b> ""</p>

Attribute Name	Content
<b>Attribute bei ART_SETTING_FILE != "" und COLOR_SOURCE != No</b>	
fme_color fme_fill_color	<p>Die Informationen zur farblichen Präsentation der Geometrielemente werden unter Geograf in den <u>Katalogdaten</u> gespeichert. Wenn diese mit dem Parameter <u>ART_SETTING_FILE</u> geladen werden erzeugt der Reader die Attribute <u>art_katalog_scr_color</u>, <u>art_layer_scr_color</u>, <u>art_layer_pen_color</u>, <u>art_scr_color</u> und <u>art_pen_color</u> mit den FME-konformen Farbinformationen. Mit dem Parameter <u>COLOR_SOURCE</u> kann ausgewählt werden, ob die FME-Farbe aus der Bildschirm- oder Stiftfarbe, aus der Art, der Ebene oder dem Katalog erzeugt werden soll. Mit dem Parameter <u>COLOR_INDIVIDUAL</u> kann entschieden werden, ob unter Geograf individuell gesetzte Farben die Farbe aus den Katalogdaten ersetzen soll.</p> <p>Steht die ausgewählte Farbinformation aufgrund unvollständig geladener Katalogdaten nicht zur Verfügung, werden die Attribute nicht erzeugt.</p> <p><b>Value:</b> char(20), RGB-Farbwerte entsprechend FME-Konventionen <b>Default:</b> No default</p>
<b>Attribute bei OBJECT_PARSER = LINK oder REMOVE_OBJECT_DATA = Yes</b>	
out_link_id out_object{}.link out_object{}.linktyp	<p>Im Modus <u>OBJECT_PARSER = LINK</u> setzt der Reader beim Objekt das Attribut <u>out_link_id</u>, bei den Objektelementen die Listenattribute <u>out_object{}.link</u> und <u>out_object{}.linktyp</u>. Diese Verweisstruktur wird auch vom Writer ausgewertet, die Objekte aus den Objektteilen zusammengesetzt.</p> <p>In den Modi <u>OBJECT_PARSER = LINE</u> oder <u>AREA</u> wird diese Struktur für die Elemente erzeugt, die nicht mit <u>REMOVE_OBJECT_DATA</u> gelöscht werden.</p> <p>Der Reader erzeugt die Links aus dem numerischen Elementschlüssel <u>out_key</u>, dem <b>Writer können alphanumerische Schlüssel vorgegeben werden.</b></p> <p><b>Value:</b> integer (Reader), <b>string (Writer) für id und link</b> <b>Value:</b> integer für linktyp (1=Ausgestaltung, 2=Definition) <b>Default:</b> 0</p>
<b>Attribute bei OBJECT_PARSER = LIST</b>	
out_link_id out_link{}.link out_link{}.linktyp	<p>Im Modus <u>OBJECT_PARSER = LIST</u> setzt der Reader beim Objekt die Listenattribute <u>out_link{}.link</u> und <u>out_link{}.linktyp</u>, bei den Objektbestandteilen das Attribut <u>out_link_id</u>. Auch diese Verweisstruktur wird vom Writer ausgewertet.</p> <p>Der Reader erzeugt die Links aus dem numerischen Elementschlüssel <u>out_key</u>, dem <b>Writer können alphanumerische Schlüssel vorgegeben werden.</b></p> <p><b>Value:</b> integer (Reader), <b>String (Writer) für id und link</b> <b>Value:</b> integer für linktyp (1=Ausgestaltung, 2=Definition) <b>Default:</b> 0</p>
<b>Attribute bei OBJECT_PARSER = LINE oder AREA und REMOVE_OBJECT_DATA = No</b>	
out_elements{} out_elements{}.geometry	<p>Im Modus <u>OBJECT_PARSER = LINE</u> oder <u>AREA</u> speichert der Reader alle Elemente, die nicht in der Featuregeometrie aufgegangen sind und mit <u>REMOVE_OBJECT_GEOM</u> oder <u>REMOVE_OBJECT_DATA</u> gelöscht werden als Liste. Die Geometrie der Elemente wird im Attribut <u>out_elements{}.geometry</u> als OGC-WKT gespeichert. Auch diese Listen wertet der Writer aus.</p>



Attribute Name	Content
<b>Attribute bei TEXT_PARSER = LINK</b>	
out_text_id out_text_link	<p>Im Modus <code>TEXT_PARSER = LINK</code> erhalten beschriftete Elemente das Attribut <code>out_text_id</code>, die Beschriftungstexte verweisen auf die ID mit dem Attribut <code>out_text_link</code>. Der Writer wertet diese Verlinkung aus.</p> <p>Der Reader erzeugt die Links aus dem numerischen Elementschlüssel <code>out_key</code>, dem Writer können alphanumerische Schlüssel vorgegeben werden.</p> <p><b>Value:</b> integer (Reader), string (Writer) für id und link</p> <p><b>Default:</b> 0</p>
<b>Attribute bei TEXT_PARSER = LIST oder EXCLUSIVE</b>	
out_text{}	<p>Im Modus <code>TEXT_PARSER = LIST</code> und <code>EXCLUSIVE</code> erhalten beschriftete Elemente eine Liste mit den Beschriftungstexten. Die Liste <code>out_text{}</code> enthält alle FeatureAttribute des Elements <code>out_text</code>.</p>
<b>Attribute zur Steuerung des Writers</b>	
out_file	<p>Mit dem Attribut können Features in verschiedene GrafBat-Dateien umgeleitet werden. Wenn keine Pfadangabe enthalten ist, erzeugt der Writer die Datei im Verzeichnis des DATASET-Parameters. (nur Writer)</p> <p><b>Value:</b> string</p> <p><b>Default:</b> No default</p>
out_sort	<p>Mit dem Attribut kann eine Sortierung vorgegeben werden, der Writer sortiert Features mit belegten Attribut in aufsteigender Reihenfolge.</p> <p><b>Value:</b> integer</p> <p><b>Default:</b> No default</p>
out_comment	<p>Mit dem Attribut können Kommentarzeilen zu den einzelnen Features in die auszugebende GrafBat-Dateien geschrieben werden. (nur Writer)</p> <p><b>Value:</b> string</p> <p><b>Default:</b> No default</p>
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	<p>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.</p> <p>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.(nur Writer)</p> <p><b>Range:</b> 0-99999 für *_art, 0-9999 für *_ebene und *_katalog</p> <p><b>Default:</b> 0</p>

## Punkte

### FeatureType: P

out\_type: out\_point

fme\_type: fme\_point

fme\_geometry: fme\_point

Da Geograf aus dem Vermessungswesen stammt, ist das grundlegende Geometrieelement der vermessene Punkt. Linien und Objekte verweisen im Regelfall auf die Punkte, eine Änderung der Punktkoordinaten bewirkt immer auch eine Änderung der darauf verweisenden Geometrien.

Punkte können mit einer 15-stelligen Punktnummer belegt sein, diese muss dann innerhalb eines Projektes eindeutig sein.

Zu einem Punkt können mehrere Berechnungsansätze mit verschiedenen Ergebniskoordinaten vorliegen, diese werden gelesen und in einer Liste gespeichert, wenn `READ_DIGIT` gesetzt ist.

Attribute Name	Content
out_number	<p>Die Nummer des Punktes, sie ist innerhalb eines Projektes eindeutig. Groß- und Kleinschreibung wird nicht unterschieden, Sonderzeichen außer dem Sternchen '*' dürfen vorkommen.</p> <p>Es ist entweder out_number oder out_name belegt.</p> <p><b>Value:</b> char(15) <b>Default:</b> No default</p>
out_name	<p>Name des Punktes, wenn die Punktnummer nicht als eindeutig gekennzeichnet ist. Er kann mehrfach in dem Projekt vorkommen.</p> <p>Es ist entweder out_number oder out_name durch den Reader belegt. Der Writer wertet vorrangig das Attribut out_number aus.</p> <p><b>Value:</b> char(15) <b>Default:</b> No default.</p>
out_rotation	<p>Der Drehwinkel des Punktes in Grad entsprechend den FME-Konventionen</p> <p><b>Range:</b> 0.0 – 360.0 <b>Default:</b> 0.0</p>
out_size	<p>Die Symbolgröße des Punktes, im Regelfall ist diese auf 0.0 gesetzt., die Präsentation erfolgt dann aufgrund der Größeneinstellung der <u>Katalogdaten</u>.</p> <p><b>Range:</b> float <b>Default:</b> 0.0</p>
out_size_prop	<p>Eigenschaften der Symbolgröße</p> <p>Wenn unter Geograf die Symbolgröße individuell eingestellt wird, ist dieses Attribut ungleich 'A' belegt.</p> <p>A = Größe aus Artdefinition, out_size ist mit 0.0 belegt M = Größe in Meter F = Größe als Faktor zur Artdefinition</p> <p>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.</p> <p><b>Range:</b> string <b>Default:</b> A</p>

Attribute Name	Content
out_l_stat	Lagegenauigkeit
out_h_stat	Höhengenauigkeit 0 = ungültig 1 = digitalisiert 2 = gerechnet 3 = fest <b>Range:</b> 0-3 <b>Default:</b> 2
out_date	Datum der Berechnung, Geograf speichert nur Jahrgänge von 1900 bis 2027 <b>Value:</b> date, entsprechend FME-Datumskonventionen <b>Default:</b> No default
out_symbol{}.out_oska	Mehrfachsymbole werden in dieser Liste abgelegt. Das Attribut oska beinhaltet Katalog, Ebene und Art, mit Punkt getrennt. <b>Value:</b> string für out_oska und out_size_prop <b>Default:</b> No default <b>Value:</b> float für out_rotation und out_size <b>Default:</b> 0.0
out_symbol{}.out_rotation	
out_symbol{}.out_size	
out_symbol{}.out_size_prop	
out_color	Individuell vergebene Symbolfarbe <b>Value:</b> char(20), RGB-Farbwerte entsprechend FME-Konventionen <b>Default:</b> <null>
Attribute bei <u>READ_OFFSET</u> = Yes	
out_x_off	Verschiebevektor für die Rissansicht <b>Range:</b> float <b>Default:</b> 0.0
out_y_off	
out_z_off	
Attribute bei <u>READ_DIGIT</u> = Yes	
out_ansatz	Die Ansatznummer der Koordinatenberechnung <b>Value:</b> integer <b>Default:</b> 0
out_dig_trafo	Transformationspunkt 1 = Punkt ist Passpunkt für die Transformation <b>Range:</b> 0-1 <b>Default:</b> 0
out_dig_dot	digitalisierter Punkt 1 = Punkt ist digitalisiert <b>Range:</b> 0-1 <b>Default:</b> 0
out_dig_aus	ungültig für Transformation 1 = Punkt ist als Passpunkt ungültig für die Transformation gesetzt <b>Range:</b> 0-1 <b>Default:</b> 0

Attribute Name	Content
out_dig_rand	<p>Punkt ist verzerrt (in älteren GrafBat-Versionen wurde der Schalter zum kennzeichnen von Randpunkten benutzt)</p> <p>1 = Punkt ist verzerrt</p> <p><b>Range:</b> 0-1 <b>Default:</b> 0</p>
out_dig_soll	<p>Punkt ist Passpunkt aus ehemalg digitalisiertem Punkt</p> <p>1 = Punkt ist Passpunkt aus Digitalisierung</p> <p><b>Range:</b> 0-1 <b>Default:</b> 0</p>
out_dig_gew	<p>Gewicht der digitalisierten Koordinaten</p> <p><b>Range:</b> 0-9999 <b>Default:</b> 0</p>
out_l_gew out_h_gew	<p>Lage-, Höhengewicht</p> <p><b>Range:</b> 0-9999 <b>Default:</b> 0</p>
out_l_ctrl out_h_ctrl	<p>Lage-, Höhenkontrolle</p> <p><b>Range:</b> 0-9999 <b>Default:</b> 0</p>
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	<p>Geograf speichert Mehrfachbestimmungen eines Punktes fortlaufend. Die Koordinate und die Bestimmungsparameter speichert der Reader in der Liste out_dig.</p>

## Linien

### FeatureType: L

out\_type: out\_line

out\_type: out\_polyline

out\_type: out\_pointline

out\_type: out\_arc

out\_type: out\_circle

out\_type: out\_clothoid

fme\_type: fme\_line

fme\_geometry: fme\_line, fme\_point für Bögen

Linien werden als gerade oder bogenförmige Verbindung zwischen zwei Punkten aufgefasst, so ist es auch datentechnisch modelliert. Erst in den letzten GrafBat-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 oder out\_pointline markiert. Die als out\_pointline gespeicherte Linie unterscheidet sich von der mit out\_polyline markierten dahingehend, dass hier auch die Vertexkoordinaten an bestehende Punkte gebunden sind. Lageänderungen der Punkte wirken sich dann auch auf die Polylinie aus. Die mit out\_polyline gekennzeichnete Linie hält in Geograf eigene Vertexkoordinaten vor, sie sind unabhängig von eventuell an gleicher Stelle liegender Punkte.

Der Writer bindet den Anfangs- und Endpunkt einer Linie immer an vorhandene Punkte, bei einer out\_pointline werden auch die Vertex mit Punkten verknüpft. Wenn solche in den zu schreibenden Daten nicht vorkommen, werden neue erzeugt. Neu erzeugte Punkte erhalten dann die Informationen aus den Linienattributen out\_point\_katalog, out\_point\_ebene, out\_point\_art.

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. Soweit der Parameter ARC\_TO\_POLYLINE gesetzt ist, werden Kreise und Bögen vom Reader in Bogenpolygone umgewandelt, sie erhalten dann das Attribute out\_type = out\_polyline.

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.

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 werden dann auf die Einzellinien kopiert. Wenn Polylinien erzeugt werden sollen, muss das Attribut out\_type explizit gesetzt sein oder der Geograftyp beim Ausgabe FeatureType angegeben werden.

Attribute Name	Content
out_pattern_side_change	Das Linienmuster wird an der Linie gespiegelt. <b>Value:</b> boolean <b>Default:</b> No
out_pattern_start_change	Das Linienmuster wird umgedreht, es verläuft vom Endpunkt zum Anfangspunkt der Linie. <b>Value:</b> boolean <b>Default:</b> No
out_size	Die Linienmusterbreite gibt den Abstand von parallele Linienmustern zur Linienachse an. In der Regel ist sie auf '0.0' gesetzt, es wird dann die in den Katalogdaten angegebene Breite benutzt. <b>Range:</b> float <b>Default:</b> 0.0

Attribute Name	Content
out_size_prop	<p>Eigenschaften der Linienmusterbreite</p> <p>Wenn unter Geograf die Größe individuell eingestellt wird, ist dieses Attribut ungleich 'A' belegt.</p> <p>A = Größe aus Artdefinition, <code>out_size</code> ist mit 0.0 belegt</p> <p>M = Größe in Meter</p> <p>F = Größe als Faktor zur Artdefinition</p> <p><b>Range:</b> string</p> <p><b>Default:</b> A</p>
out_pattern_dist	<p>Startwert des Linienmusters in Koordinateneinheiten</p> <p><b>Range:</b> float</p> <p><b>Default:</b> 0.0</p>
out_color	<p>Individuell vergebene Linienfarbe</p> <p><b>Value:</b> char(20), RGB-Farbwerte entsprechend FME-Konventionen</p> <p><b>Default:</b> &lt;null&gt;</p>
out_number1	<p>Nummer des Linienanfangspunktes (nur Reader)</p> <p><b>Value:</b> char(15)</p> <p><b>Default:</b> &lt;null&gt;</p>
out_number2	<p>Nummer des Linienendpunktes (nur Reader)</p> <p><b>Value:</b> char(15)</p> <p><b>Default:</b> &lt;null&gt;</p>
out_rotation	Die Geometrieattribute von Bögen, Kreisen und Klothoiden
out_primary_axis	<b>Value:</b> float
out_secondary_axis	<b>Default:</b> 0.0
out_start_angle	
out_sweep_angle	
out_clothoid_param	

## Splines

### FeatureType: L

out\_type: out\_spline

fme\_type: fme\_line

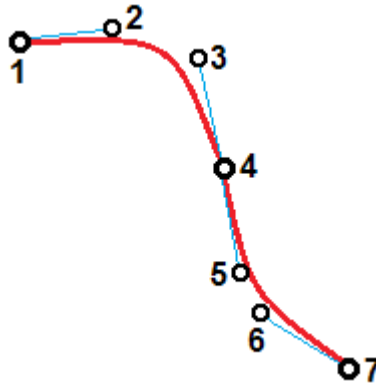
fme\_geometry: fme\_line

Geograf unterscheidet bei der Erzeugung drei Varianten von Splines, den polygonalen, den tangential und den durch die Stützpunkte gerundeten Splines.

Der polygonal erzeugte Spline unterscheidet sich von der Polylinie out\_polyline nur dadurch, dass er nicht an Punkte gebunden werden kann, er weist für Start- und Endpunkt und für die Vertex immer eigene Koordinaten aus. Er ist somit auch nicht für topologische Netze geeignet.

Der Reader erzeugt aus polygonalen Splines Linienzüge aus allen Stützpunkten, die Schalter `SPLINE_SEGMENT_POINTS` hat hier keine Auswirkung. Unter Geograf können für den polygonalen Spline auch bogenförmige Segmente erzeugt werden, der Reader wandelt diese Segmente immer in Bogenpolygone um. Der Reader setzt bei den polygonalen Splines das Geometrieattribut `out_spline_info` = linear.

Die tangential oder durch die Stützpunkte gerundeten Splines speichert Geograf identisch als kubische Splines, die Art der Erzeugung hat lediglich Auswirkung auf die Lage der Stützpunkte.



*Stützpunkte eines Splines*

Splines weisen  $3 \cdot (h-1) + 1$  Stützpunkte auf. Zwischen 2 Hauptpunkten, die auf der Kurve liegen, befinden sich jeweils zwei Kontrollpunkte, die selbst neben der Kurve liegen und die Tangentenrichtung im Hauptpunkt vorgeben.

Ist der Parameter `SPLINE_PHANTOM_POINTS` auf No gesetzt, erzeugt der Reader einen Linienzug aus allen Haupt- und Kontrollpunkten, das Geometrieattribut `out_spline_info` wird dann auf 'controls' gesetzt. Der Writer erzeugt aus einem so generierten Linienzug einen identisch ausgerundeten Geograf-Spline.

Ist der Schalter `SPLINE_PHANTOM_POINTS` auf Yes gesetzt, wird der Linienzug vorerst nur aus den Hauptpunkten ( $n=1, 4, 7, \dots$ ) erzeugt. Das Geometrieattribut lautet dann `out_spline_info` = '1 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+1> segments gekennzeichnet.

Wenn bei einer Transformation von Geograf nach Geograf die Geometrien der Splines nicht verändert werden, sollte der Schalter `SPLINE_PHANTOM_POINTS` auf No belassen werden. So werden die Splines originalgetreu wiederhergestellt. Werden die Linienzüge geometrisch verändert, 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 identisch vorliegen, gelingt dies nicht originalgetreu.

Wenn die Splinerzeugung 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. So generierte Splines werden nur 2D oder mit einheitlicher Höhe (2,5D) übertragen.

Geograf speichert Splines als 2,5D oder 3-dimensionale Kurven. Bei den 2,5D-Splines werden nur 2-dimensionale Stützpunkte abgelegt, ein Höhenwert für die gesamte Kurve wird als Attribut gespeichert. Bei den 3D-Splines wird für jeden Stützpunkt eine 3-dimensionale Koordinate abgelegt. Der Reader erzeugt aus beiden Varianten einen 3D-Linienzug. Der Writer untersucht einen übergebenen Linienzug, bei einheitlichen Höhenwerten wird ein 2,5D-Spline, ansonsten ein 3D-Spline erzeugt. Ebenso wird ein offener bzw. geschlossener Splinetyp anhand der Geometrie gesetzt, das Attribut `out_typ` wird nicht ausgewertet.

Älteren GrafBat-Versionen schrieben, soweit Splines aus gefangenen Polygonen erzeugt wurden, die ursprünglichen Linien mit in die Ausgabedaten, unter Geograf wurden diese bei der Visualisierung unterdrückt. Der Reader unterdrückt solche Linien ebenso, wenn sie in älteren Versionen auftreten. Ab der Geograf-Version 10 werden solche Linien nicht mehr geschrieben.

Attribute Name	Content
<code>out_rotation</code>	Werte ohne Bedeutung
<code>out_size</code>	
<code>out_typ</code>	<p>Splinetyp (nur Reader)</p> <p>1 = linear offen</p> <p>2 = linear geschlossen</p> <p>3 = kubisch offen</p> <p>4 = kubisch geschlossen</p> <p>6 = <del>linear offen</del>, 3D</p> <p>7 = <del>linear geschlossen</del>, 3D</p> <p>8 = <del>kubisch offen</del>, 3D</p> <p>9 = <del>kubisch geschlossen</del>, 3D</p> <p>Die Typen 6-9 werden von Geograf ab GrafBat-Version 10 nicht mehr gesondert erzeugt.</p> <p><b>Range:</b> 1-9</p> <p><b>Default:</b> No default</p>
<code>out_spline_info</code>	<p>Art der Linienzugerzeugung</p> <p>linear = aus linearen Spline</p> <p>controls = aus kubischen Spline mittels Kontrollpunkten</p> <p>&lt;n&gt; segments = aus kubischen Spline mit &lt;n&gt; Segmenten</p> <p>generate = Spline soll aus Linienpunkten erzeugt werden (nur Writer)</p> <p>detect = Linienzug wird analysiert und die Stütz- und Kontrollpunkte erzeugt (nur Writer)</p> <p><b>Value:</b> string</p> <p><b>Default:</b> No default</p>



## Texte

### FeatureType: T

out\_type: out\_text

out\_type: out\_multi\_text

fme\_type: fme\_text

fme\_geometry: fme\_point

Geograf-Texte besitzen eine eigene Koordinate und einen Winkel, der die Orientierung im Koordinatensystem angibt. Die Textgröße wird nur gespeichert, wenn sie individuell vergeben wurde, ansonsten gilt die in den Katalogdaten maßstabsabhängige angegebene Größe.

Texte 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.

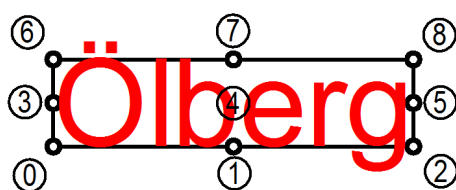
Bei mehrzeiligen Texten erzeugt der Reader das Attribut out\_text\_string mit allen Textzeilen, getrennt mit dem MULTI\_LINE\_TEXT\_DELIM, als Standard ist hier der Zeilenumbruch '\n' voreingestellt. Die Textgröße wird mit der Zeilenanzahl multipliziert.

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 ebenso 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\_stringf, die die einzelnen Textzeilen enthält. Der Text wird dann mit out\_type = out\_multi\_text gekennzeichnet.

Mit der Geograf-Version 8 können 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.

Eine Herausforderung ist die korrekte Umsetzung von Textpositionen. Geograf setzt die Koordinate an der Grundlinie bzw. an der Oberkante eines Großbuchstabens an, Umlautpunkte und Unterlängen (Descent) ragen über diesen Bereich hinaus. Die Textgröße out\_size wird als die Höhe eines Großbuchstabens interpretiert. Andere Systeme setzen den Abstand zweier Grundlinien (Ascent+Descent+Zeilenabstand) als Textgröße an, dementsprechend liegen dann die Textschwerpunkte oberhalb des Ascent und unterhalb des Descent.



*Geograf Textschwerpunkt*



*Font-Metrik*

Um diesbezüglich Korrekturen vornehmen zu können, stehen die Parameter TEXT\_SCALE und RESET\_TEXT\_POSITION zur Verfügung. Die Verwendung dieser Parameter ist nur sinnvoll, wenn zusätzlich die Katalogdaten gelesen werden, da nur dann die von Geograf visualisierten Textgrößen zur Verfügung stehen.

Mit TEXT\_SCALE kann ein Textskalierungsfaktor angebracht werden. Der Wert voreingestellte Wert FME\_DEFAULT ist auf die Darstellung im Data Inspector abgestimmt.

Die Parametereinstellung RESET\_TEXT\_POSITION = DESCENT versetzt die Textposition um den Descent nach unten. Für Systeme, die den Text immer von der Standardposition links-unten = 0 aus zeichnen, kann der Parameter auf POSITION gesetzt werden, der Reader rechnet dann mit Textlänge und -höhe die Textkoordinate auf die Standardposition zurück. Dabei wird auch der Wert in out\_position auf 0 zurückgesetzt. Die Einstellung DESCENT\_POSITION wendet beide Korrekturen an.

Attribute Name	Content
out_rotation	Drehwinkel des Textes in Grad entsprechend den FME-Konventionen <b>Value:</b> float <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 Artendatei. Der Reader belässt diesen Wert bei 0.0, wenn keine individuelle Größe vergeben ist. Der Writer benutzt den hier vergebenen Wert, auch 0.0. Nur wenn out_size nicht vergeben ist, wird auf <u>out_text_size</u> zurückgegriffen. <b>Value:</b> float <b>Default:</b> 0.0
out_size_prop	Eigenschaften der Textgröße Wenn unter Geograf die Textgröße individuell eingestellt wird, ist dieses Attribut ungleich 'A' belegt. A = Größe aus Artdefinition, <u>out_size</u> ist mit 0.0 belegt M = Größe in Meter F = Größe als Faktor zur Artdefinition <b>Value:</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 <u>STANDARD_TEXT_SIZE</u> belegt. <b>Wenn die Katalogdaten gelesen werden, wird out_text_size aus der Größenangabe der Artendatei und dem <u>VIEW_SCALE</u> errechnet.</b> Der Writer benutzt diese Angabe nur, wenn <u>out_size</u> nicht belegt ist. <b>Value:</b> float <b>Default:</b> <u>STANDARD_TEXT_SIZE</u>
out_text_string	Inhalt des Textes, mehrzeilige Texte erhalten als Trennzeichen <u>MULTI_LINE_TEXT_DELIM</u> . <b>Value:</b> string <b>Default:</b> ""
out_text_string{}	Inhalt der Textzeilen bei Multitexten, wird vom Reader nur bei <u>MULTI_LINE_TEXT=Yes</u> erzeugt. <b>Value:</b> string <b>Default:</b> ""
out_text_formel	Textformel bei Beschriftungstexten, mehrzeilige Texte erhalten als Trennzeichen <u>MULTI_LINE_TEXT_DELIM</u> . Wenn mit dem Writer Beschriftungen erzeugt werden, sollte auch immer <u>out_text_formel</u> mit einem unter Geograf zur Verfügung stehenden Beschriftungsformat gesetzt werden. Nur dann werden auch die Textinhalte und Schwerpunkte bei Veränderungen aktualisiert. <b>Value:</b> string <b>Default:</b> ""
out_text_formel{}	Textformel bei Multitexten, wird vom Reader <u>MULTI_LINE_TEXT=Yes</u> erzeugt. <b>Value:</b> string <b>Default:</b> ""

Attribute Name	Content
out_rahmen	<p>Besondere Hervorhebung durch Textrahmen</p> <ul style="list-style-type: none"> <li>0 = ohne Rahmen, Bezug vom Zentrum</li> <li>1 = Kasten, Bezug von den Ecken</li> <li>2 = großer Kreis, Bezug vom Zentrum</li> <li>3 = einfach unterstrichen, breit, Bezug vom Unterstrich</li> <li>4 = doppelt unterstrichen, eng, Bezug vom Unterstrich</li> <li>5 = <del>durchgestrichen, Bezug vom Durchstrich</del></li> <li>6 = ohne Rahmen, Bezug von den Ecken</li> <li>7 = doppelt unterstrichen, breit, Bezug vom oberen Unterstrich</li> <li>8 = kleiner Kreis, Bezug vom Zentrum</li> <li>9 = Wurzelzeichen</li> <li>10 = einfach überstrichen</li> <li>11 = einfach unterstrichen, eng</li> </ul> <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 <u>out_crossout</u> auf Yes.</p> <p><b>Range:</b> 0-11 <b>Default:</b> 0</p>
out_crossout	<p>Der Text ist durchgestrichen, die Präsentation erfolgt zusätzlich zum gesetzten Rahmen <u>out_rahmen</u>.</p> <p><b>Value:</b> boolean <b>Default:</b> false</p>
out_frei	<p>Freistellart</p> <ul style="list-style-type: none"> <li>0 = Kasten</li> <li>1 = ohne</li> <li>2 = Kreis</li> </ul> <p><b>Range:</b> 0-2 <b>Default:</b> 0</p>
out_position	<p>Position, auf die sich die Textkoordinate bezieht.</p> <ul style="list-style-type: none"> <li>0 = links-unten</li> <li>1 = mitte-unten</li> <li>2 = rechts-unten</li> <li>3 = links-mitte</li> <li>4 = mitte-mitte</li> <li>5 = rechts-mitte</li> <li>6 = links-oben</li> <li>7 = mitte-oben</li> <li>8 = rechts-oben</li> </ul> <p>Hinweis: Der FME-Data Inspector 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>

Attribute Name	Content
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 hier eine Kopie der Textkoordinate. Dem Writer müssen die Werte nicht angegeben werden, es wird dann die Textkoordinate eingesetzt.</p> <p><b>Range:</b> double <b>Default:</b> 0.0</p>
out_color	<p>Textfarbe entsprechend den FME-Konventionen</p> <p><b>Value:</b> char(20), RGB-Farbwerte entsprechend FME-Konventionen <b>Default:</b> &lt;null&gt;</p>
<b>Attribute bei TEXT_PARSER != No</b>	
out_rel_flag	<p>Flags für relative Texte, die Bit-Belegung findet sich in der GrafBat-Dokumentation.</p> <p><b>Range:</b> 0-127 <b>Default:</b> 0</p>
out_x_rel out_y_rel	<p>Koordinatenoffset der Beschriftung</p> <p><b>Value:</b> float <b>Default:</b> 0.0</p>
out_r_rel	<p>Winkeloffset der Beschriftung</p> <p><b>Value:</b> float <b>Default:</b> 0.0</p>
<b>Attribute bei READ_OFFSET = Yes</b>	
out_x_off out_y_off	<p>Verschiebevektor für die Rissansicht</p> <p><b>Value:</b> float <b>Default:</b> 0.0</p>
out_r_off	<p>Winkeloffset für die Rissansicht</p> <p><b>Value:</b> float <b>Default:</b> 0.0</p>
<b>Attribute bei READ_DIGIT = Yes</b>	
out_dig_flag	<p>Digitalisierflags</p> <p><b>Range:</b> 0-9 <b>Default:</b> 0</p>

## Schraffuren

### FeatureType: S

out\_type: out\_schraffur

fme\_type: fme\_polygon

fme\_type: fme\_line (nur out\_sub\_typ = 8)

fme\_geometry: fme\_area oder fme\_aggregate

Schraffuren weisen unter Geograf eine flächenförmige Geometrie auf, auch Saum- oder Linienschraffuren sind über ihren Umring definiert. Dementsprechend setzt der Reader sie als Polygon um. Die Pfade des Polygonumrings können auch Bögen enthalten.

Der Schraffurtyp 8 (offene Saumschraffur) nimmt eine Sonderstellung ein. Er wird unter Geograf entlang eines Linienzuges gebildet, so wird er auch vom Reader als fme\_line erzeugt.

In neueren Geograf-Versionen können Schraffuren auch 3-dimensional erzeugt werden, der Reader und der Writer setzen diese entsprechend um.

Für den Writer muss das Attribut out\_type explizit gesetzt sein oder der Geograf-Typ beim Ausgabe Feature Type angegeben 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.

Attribute Name	Content
out_rotation	Ausrichtung der Schraffur in Grad entsprechend den FME-Konventionen. <b>Range:</b> 0.0-360.0 Grad <b>Default:</b> 0.0
out_size	Musterabstand bei linienhaften oder Punktsymbolschraffuren in [mm] Plot-einheiten. <b>Value:</b> float <b>Default:</b> 0.0
out_size_prop	Eigenschaften des Musterabstandes 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>Value:</b> string <b>Default:</b> A

Attribute Name	Content
out_sub_typ	<p>Spezifizierung der Schraffurdarstellung</p> <p>0 = normale Schraffur (Linien), <u>out_art</u> aus <u>Linienarten</u></p> <p>1 = Kreuzschraffur, <u>out_art</u> aus <u>Linienarten</u></p> <p>2 = Rasterschraffur, <u>out_art</u> aus <u>Punktarten</u></p> <p>3 = Flächenschraffur, <u>out_art</u> aus <u>Linienarten</u></p> <p>4 = schraffierte Abstandsfläche, <u>out_art</u> aus <u>Linienarten</u></p> <p>5 = geschlossene Saumschraffur, <u>out_art</u> aus <u>Linienarten</u></p> <p>6 = deckende Rasterschraffur, <u>out_art</u> aus <u>Punktarten</u></p> <p>7 = Schraffur via Schraffurart, <u>out_art</u> aus <u>Schraffurarten</u></p> <p>8 = offene Saumschraffur, <u>out_art</u> aus <u>Linienarten</u></p> <p>10 = unregelmäßige Rasterschraffur, <u>out_art</u> aus <u>Punktarten</u></p> <p>11 = Linienumring, <u>out_art</u> aus <u>Linienarten</u></p> <p><b>Range:</b> 0-11</p> <p><b>Default:</b> 0</p>
out_color	<p>Schraffurfarbe entsprechend den FME-Konventionen</p> <p><b>Value:</b> char(20), RGB-Farbwerte entsprechend FME-Konventionen</p> <p><b>Default:</b> &lt;null&gt;</p>
out_name	<p>individueller Name für die Schraffur</p> <p><b>Value:</b> char(80)</p> <p><b>Default:</b> &lt;null&gt;</p>

## Böschungen

### FeatureType: B

out\_type: out\_boeschung

fme\_type: fme\_line

fme\_geometry: fme\_aggregate

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](#).

Attribute Name	Content
out_rotation	Werte ohne Bedeutung
out_size	
out_color	Schraffurfarbe entsprechend den FME-Konventionen <b>Value:</b> char(20), RGB-Farbwerte entsprechend FME-Konventionen <b>Default:</b> <null>

## Objekte

### FeatureType: O

out\_type: out\_object

fme\_type: fme\_point

fme\_geometry: fme\_aggregate, fme\_point

Geograf-Objekte sind Zusammenfassungen der obig beschriebenen atomaren Geometrien: Punkte, Linien, Splines, Texte, Schraffuren und Böschungen. Das Objekt selbst besitzt als Geometrie lediglich ein 2-dimensionales Koordinatenpaar, das den Fangpunkt repräsentiert. Auf die Geometrie wird mittels einer Referenzliste verwiesen. In der Referenzliste werden die Objektbestandteile nach Definitions- und Ausgestaltungsgeometrie unterschieden. Die Elemente der Definitionsgeometrie bilden eine oder mehrere Flächen, auch Donuts sind möglich. Die Elemente der Ausgestaltung erzeugen keine zusammenhängende Geometrie.

Die atomaren Geometrien können Bestandteil mehrerer Objekte sein. Üblicherweise werden Flächen so erfasst, dass eine Kante Definitionsgeometrie zweier benachbarter Flächen ist. Auch Elemente der Ausgestaltung können Bestandteil mehrerer Objekte sein.

Die Geometrieerzeugung des Readers wird mit dem Parametern OBJECT\_PARSER, REMOVE\_OBJECT\_GEOM und REMOVE\_OBJECT\_DATA gesteuert. Im Folgenden werden die angebotenen Varianten beschrieben.

OBJECT\_PARSER = NO

Die Objekte werden hier nicht gelesen, es erfolgt auch keine Markierung der Objektbestandteile.

OBJECT\_PARSER = LINK

Für das Geograf-Objekt wird ein punktförmiges Feature aus der Objektkoordinate erzeugt. Das Graf-Bat-Format sieht auch Objekte ohne Koordinaten vor, in diesem Fall wird ein 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 Objektgeometrien zusammengesetzt werden.

OBJECT\_PARSER = LIST

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

Es werden die Listen out\_link{}.link und out\_link{}.linktyp erzeugt, die für jeden Geometriebestandteil den Referenzschlüssel und den Geometriotyp (1=Ausgestaltung, 2=Definition) angeben. Die atomaren Geometrien, die Bestandteil eines oder mehrerer Objekte sind, erhalten das Attribut out\_link\_id, auf dessen Inhalt verweist die Liste out\_link{}.link. Die Objektbestandteile können mit dem *ListBasedFeatureMerger* zu den Objektgeometrien zusammengesetzt werden.

Die drei vorgenannten Parametereinstellungen verlangen, dass die Objektbestandteile als atomare Elemente gelesen werden, die Parameter REMOVE\_OBJECT\_GEOM und REMOVE\_OBJECT\_DATA werden automatisch auf No gesetzt.

OBJECT\_PARSER = LINE

Alle linienförmigen Geometriebestandteile werden zu einem Feature zusammengesetzt. Aus Linien der Definitionsgeometrie werden Flächen erzeugt. Die Linien der Ausgestaltungsgeometrie, hierzu gehören auch Splines und Böschungen, werden zu einem Aggregat aus Einzellinien zusammengesetzt.

Alle anderen Objektbestandteile (Punkte, Texte, Schraffuren) werden abhängig vom Parameter REMOVE\_OBJECT\_DATA als Elementliste out\_elements{} oder mit einer Liste out\_link{}.link gespeichert. Ist REMOVE\_OBJECT\_DATA auf Yes gesetzt, werden die Objektbestandteile nicht als Punkte, Texte oder Schraffuren gelesen, sie gehen im Objekt-Feature als Liste out\_elements{} auf. Dort werden alle Geometrieattribute abgelegt, auch eventuell vorhandene Sachdaten. Zusätzlich wird in der Liste die jeweilige Geometrie im Attribut out\_elements{}.geometry als OGC-WKT gespeichert.

Ist REMOVE\_OBJECT\_DATA auf No gesetzt, werden die Objektbestandteile im regulären Datenfluss als Punkte, Texte oder Schraffuren gelesen, Sie erhalten wie bei OBJECT\_PARSER = LINK die Listen



`out_object{}.link` und `out_object{}.linktyp`, das Objekt-Feature selbst das Attribut `out_link_id`, so dass ihre Objekt-zugehörigkeit auswertbar ist.

Bei Objekten ohne Liniengeometrie wird ein `fme_point` aus der Objektkoordinate erzeugt.

`OBJECT_PARSER = AREA`

Wie bei dem Modus `LINE` werden auch hier nur die linienförmigen Objektbestandteile zur Geometrieerzeugung benutzt. Bei flächenförmigen Geograf-Objekten wird die Geometrie ausschließlich aus den Linien der Definitionsgeometrie erzeugt. Eventuell vorhandene Ausgestaltungslinien gehen im Flächen-Feature als Liste `out_elements{}` auf oder werden mit `out_object{}.link` und `out_object{}.linktyp` verknüpft.

Bei linienförmigen Geograf-Objekten gehen alle Linien in die Geometrie ein, es wird ein Aggregat aus Einzellinien erzeugt.

Wie im Modus `LINE` werden alle anderen nicht linienförmigen Objektbestandteile als Elementliste `out_elements{}` oder als Linklisten `out_object{}.link`, `out_object{}.linktyp` gespeichert. Bei Objekten ohne Liniengeometrie wird auch hier ein `fme_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 dann den Namen `out_decoration`.

Wenn das Geograf-Objekt Definitions- und Ausgestaltungsgeometrie enthält, wird der Geometrienamen auf `out_composite` gesetzt. Es sind dann zwei Geometrien enthalten, deren Geometrienamen jeweils auf `out_definition` oder `out_decoration` gesetzt ist.

Da im Modus `AGGREGATE` alle Objektbestandteile in der Objektgeometrie aufgehen, ist der Schalter `REMOVE_OBJECT_DATA` hier ohne Auswirkung.

Die mit den Einstellungen `LINE`, `AREA` oder `AGGREGATE` erzeugten Geometrien erhalten die sonst üblichen Geometrieattribute als `Geometry Traits`, diese können mit dem `GeometryPropertyExtractor` ausgelesen werden. Eventuell vorhandene Sachdatenattribute der einzelnen Objektbestandteile werden nicht in das Objekt-Feature übernommen.

Ist bei den drei letztgenannten Einstellungen der Parameter `REMOVE_OBJECT_GEOM` auf `No` gesetzt, werden alle zur Objektgeometrie zusammengesetzten Objektbestandteile zusätzlich als Einzelgeometrien gelesen. Ein Verweis auf die Objektzugehörigkeit wird dabei nicht gespeichert.

Mit `ADD_OBJECT_POINT` kann dem Objekt-Feature die Geograf-Objektkoordinate als `fme_point` hinzugefügt werden, dann erhält dieser weitere Geometriebestandteil den Namen `out_snap`.

Bei flächenförmigen Objektgeometrien (Geograf-Objekte mit Definitionsgeometrie) setzt der Reader die Umringslinien zu einem geschlossenen Pfad zusammen. Die Linien der Ausgestaltungsgeometrie werden als Aggregat aus Einzellinien gelesen. Wenn der Parameter `CREATE_OBJECT_PATHS` gesetzt ist, werden auch die Ausgestaltungslinien soweit möglich zu Pfaden zusammengesetzt. Bei netzförmigen Geometrien haben Linien mit gleicher Art den Vorrang bei der Pfadverknüpfung.

Die Objektart 9999 ist für Beschriftungsobjekte reserviert. Bis zur GrafBat-Version 3 wurden Beschriftungen als Objekte mit dieser Art gebildet.

Schraffuren, die auf einem Flächenobjekt erzeugt werden, erhalten keine eigene Geometrie, sie füllen immer die Definitionsgeometrie aus. Der Reader erzeugt hieraus ausschließlich die Attribute `out_sch_art`, ... . Um mit dem Writer Objektschraffuren zu erzeugen, muss mindestens das Attribut `out_sch_art` gesetzt sein.

Objekte, deren Geometrie unter Geograf im Arbeitsablauf beschädigt wurde, werden auch fehlerhaft in die GrafBat-Datei ausgegeben. Der Reader erkennt Fehler bei der Geometriebildung und erzeugt eine Fehlermeldung. Die fehlerhaften Elemente der Definitionsgeometrie werden als Ausgestaltung in das Objekt-Feature eingefügt, zusätzlich werden die Attribute `out_error_name` und `out_error_msg` erzeugt. Wird ein so markiertes Feature wieder über den Writer ausgegeben, wird die Fehlermeldung in einer zusätzlichen Stapeldatei abgelegt.

Alle für den Reader aufgeführten Möglichkeiten der Objekterzeugung werden auch durch den Writer unterstützt. Es werden flächenförmige Geometrien als Definitionsgeometrie, alle anderen als Ausgestaltungsgeometrie an Geograf übergeben. Die `Geometry Traits` werden soweit vorhanden ausgewertet. Ist bei Geometriebestandteilen das Trait `out_type` gesetzt, werden die Bestandteile als dementsprechende

Geograf-Typen umgesetzt. Ansonsten wird der Geograf-Elementtyp aus dem `fine_type` und der Geometrie ermittelt.

Es werden auch die gemischten Varianten unterstützt. So ist es legitim, ein Flächenobjekt aus der Featuregeometrie zu erzeugen, Ausgestaltungsgeometrie wie Texte und Punktsymbole aus einer Liste `out_elements{}` zu erzeugen oder über Verlinkung mit `out_object{}.link` an das Objekt zu hängen.

Attribute Name	Content
<code>out_name</code>	Der Name des Objektes. Im Attribut <code>out_name_key</code> ist festgelegt, ob der Objektname eindeutig ist. <b>Value:</b> char(80) <b>Default:</b> Void
<code>out_object_typ</code>	Objekttyp 0 = Normales Objekt 1 = Splineobjekt 2 = Beschriftung <b>Range:</b> 0-2 <b>Default:</b> 0
<code>out_date</code>	Datum der Objekterzeugung (nach FME-Datumskonventionen) <b>Value:</b> date <b>Default:</b> <null>
<code>out_x</code> <code>out_y</code>	Koordinaten des Objektreferenzpunktes <b>Value:</b> double <b>Default:</b> 0.0
<code>out_ansatz</code>	Ansatznummer des Objektes, aktuelle Geograf-Versionen speichern diesen Wert nicht <b>Value:</b> integer <b>Default:</b> 0
<code>out_l_stat</code>	Lagegenauigkeit, aktuelle Geograf-Versionen speichern diesen Wert nicht 0 = ungültig 1 = digitalisiert 2 = gerechnet 3 = fest <b>Range:</b> 0-3 <b>Default:</b> 2
<code>out_oberobjekt</code>	Elementschlüssel des Oberobjekts Der Reader erzeugt den Schlüssel aus dem numerischen Elementschlüssel <code>out_key</code> , dem Writer können alphanumerische Schlüssel vorgegeben werden. <b>Value:</b> integer (Reader), string (Writer) <b>Default:</b> 0
<code>out_link_id</code>	Oberobjekte, auf die verwiesen wird, erhalten diesen Schlüssel. Der Reader legt hier eine Kopie des Wertes aus <code>out_key</code> ab. Das Unterobjekt verweist mit <code>out_oberobjekt</code> diesen Schlüssel. Der Reader erzeugt den Wert aus dem numerischen Elementschlüssel <code>out_key</code> , dem Writer können alphanumerische Schlüssel vorgegeben werden. <b>Value:</b> integer (Reader), string (Writer) <b>Default:</b> 0
<code>out_name_key</code>	Gibt an, ob der Objektname eindeutig ist. <b>Value:</b> boolean <b>Default:</b> No

Attribute Name	Content
out_sch_katalog out_sch_ebene out_sch_art out_sch_sub_typ out_sch_size out_sch_rotation out_sch_color	<p>Objektschraffuren besitzen keine eigene Geometrie, sie füllen immer die Definitionsgeometrie eines Flächenobjektes aus.</p> <p>Wenn eine Objektschraffur gesetzt ist, werden die Schraffurwerte in diesen Attributen gespeichert.</p> <p>Der Writer erzeugt eine Objektschraffur, wenn out_sch_art gesetzt ist. Katalog und Ebene werden vom Objekt abgeleitet, wenn sie nicht explizit gesetzt sind.</p> <p>Die Werte entsprechen den Angaben in den <u>Schraffuren</u>.</p>
out_object_geom	<p>Auflistung der Inhalte der Objektgeometrie (nur Reader)</p> <p>Der String gibt Auskunft über die Geometrietypen des Objektinhalts.</p> <ol style="list-style-type: none"> <li>1. Zeichen: A = Fläche</li> <li>2. Zeichen: L = Linie</li> <li>3. Zeichen: S = Schraffur</li> <li>4. Zeichen: B = Böschung</li> <li>5. Zeichen: T = Text</li> <li>6. Zeichen: P = Punkt</li> </ol> <p>Geometrietypen, die nur einmalig im Objekt vorkommen, werden mit Kleinbuchstaben, mehrmalige mit Großbuchstaben markiert. Ist ein Typ nicht enthalten, ist das Zeichen '0' (Null).</p> <p><b>Value:</b> char(6)</p> <p><b>Default:</b> 000000</p>
Attribute bei READ_OFFSET = Yes	
out_x_off out_y_off	<p>Verschiebevektor in Koordinateneinheiten</p> <p><b>Value:</b> float</p> <p><b>Default:</b> 0.0</p>

## Modelle

### FeatureType: M

out\_type: out\_model

fme\_type: fme\_point

fme\_geometry: fme\_point

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

Attribute Name	Content
out_system	Systemnummer <u>Dokumente</u> , die über eine Transformation mit mehreren Passpunkten georeferenziert wurden, erhalten diese Systemnummer im Attribut <u>out_system</u> . <b>Value:</b> integer <b>Default:</b> No default
out_old_x	Rechtswert des Schwerpunktes im alten System <b>Value:</b> double <b>Default:</b> No default
out_old_y	Hochwert des Schwerpunktes im alten System <b>Value:</b> double <b>Default:</b> No default
out_new_x	Rechtswert des Schwerpunktes im neuen System <b>Value:</b> double <b>Default:</b> No default
out_new_y	Hochwert des Schwerpunktes im neuen System <b>Value:</b> double <b>Default:</b> No default
out_par_a out_par_b out_par_c out_par_d	Transformationsparameter <b>Value:</b> double <b>Default:</b> No default
out_ansatz	Nummer des Berechnungsansatzes. <u>Punkte</u> , die in diesem System digitalisiert worden sind, erhalten diese Ansatznummer im Attribut <u>out_ansatz</u> . <b>Range:</b> 0-9999 <b>Default:</b> 0
out_par	Anzahl der Transformationsparameter <b>Value:</b> 4 oder 6 <b>Default:</b> 4
out_name	Systemname <b>Value:</b> string <b>Default:</b> ""

## Dokumente

### FeatureType: D

out\_type: out\_doc

out\_type: out\_raster

fme\_type: fme\_area oder fme\_raster

fme\_geometry: fme\_polygon oder fme\_aggregate

Geograf ermöglicht die Einbindung von externen Dokumenten. In der GrafBat-Datei wird der Dateiname mit Pfadangabe und die Georeferenzierung zu diesem Dokument gespeichert. Die Dokumente werden als Feature gelesen, wenn der Parameter `READ_DOCUMENT` auf Yes gesetzt ist.

Aus Geograf-Dokumenten werden Flächen-Features erzeugt. Das Umringspolygon wird immer im Uhrzeigersinn erzeugt, die erste Koordinate entspricht der linken oberen Ecke des Dokuments. Der Writer interpretiert Polygone auf selbige Weise.

Ist der Parameter `READ_RASTER` gesetzt, werden für Dokumente des Typs 0 = Bitmaps die Rasterdaten soweit möglich aus der referenzierten Bilddatei mit der Funktion `@File()` nachgeladen, die Geometrie dabei in ein `fme_raster` umgewandelt. Die Funktion `@File()` lädt auch die Georeferenzierung der Bitmaps aus eventuell vorliegenden World-Files nach. Da nicht sicher gestellt ist, dass diese der aktuelle Georeferenzierung entsprechen, werden die Koordinaten des Rasters aus den Angaben der GrafBat-Datei beibehalten.

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
out_name	Name des Dokuments <b>Value:</b> char(255) <b>Default:</b> No default
out_typ	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>Range:</b> 0-6 <b>Default:</b> 0
out_pen	Stift für 1Bit-Bilder <b>Range:</b> 0-9999 <b>Default:</b> 0
out_trans	Transparente Darstellung <b>Range:</b> 0-1 <b>Default:</b> 1

Attribute Name	Content
out_stufe	Hierarchiestufe der Darstellung <b>Range:</b> 0-9999 <b>Default:</b> 0
out_system	Nummer des Digitalisiersystems. Wenn die Georeferenzierung des Dokuments über eine Transformation in Geograf erzeugt wurde, verweist diese Nummer auf <u>out_system</u> des Modells. <b>Range:</b> 0-9999 <b>Default:</b> 0
out_colorkey	Farbnummer für 1Bit-Bilder, die Nummer verweist auf eine der <u>Bildschirmfarben</u> . <b>Range:</b> 0-9999 <b>Default:</b> 0
out_doc_lux out_doc_luy out_doc_rox out_doc_roy oder: out_doc_width out_doc_height	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 dieses Datensatzes 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). <b>Value:</b> double <b>Default:</b> No default

## DGM

### FeatureType: G

out\_type: out\_dgm

fine\_type: fine\_surface

fine\_geometry: fine\_aggregate

Der Reader kombiniert einen Horizont und die zu ihm gehörigen DGM-Dreiecke zu einem 3D-Feature, es wird eine Geometrie 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\_pointEbene und out\_point\_katalog gesteuert werden.

Attribute Name	Content
out_horizont	Horizontnummer <b>Range:</b> 0-9997 <b>Default:</b> No default
out_name	Name des Horizonts <b>Value:</b> char(255) <b>Default:</b> No default
out_line_art	Linienart für die Horizontlinie <b>Range:</b> 0-99999 <b>Default:</b> 0
out_text_art	Textart für die Horizontbezeichnung <b>Range:</b> 0-99999 <b>Default:</b> 0
out_level	Darstellungsstufe für den Horizont im Profil <b>Range:</b> 0-9999 <b>Default:</b> 0
out_color_min out_color_max	Bildschirmfarbe der minimalen und maximalen Höhe für die DGM-Darstellung. <b>Value:</b> char(20), RGB-Farbwerte entsprechend FME-Konventionen <b>Default:</b> <null>
out_color_ramp	Die Farbpalette für die DGM-Darstellung als Komma-separierter String. Es wird der String der GrafBat-Datei übernommen, die Farbwerte sind hier hexadezimal in umgekehrter Reihenfolge angegeben (BGR statt RGB). <b>Value:</b> char(255) <b>Default:</b> <null>
out_pen_inside	Stiftnummer für die Dreiecksmaschen <b>Range:</b> 0-9999 <b>Default:</b> 0
out_pen_break	Stiftnummer für die Bruchkanten <b>Range:</b> 0-9999 <b>Default:</b> 0

---

out_pen_outside	Stiftnummer für die Umringslinien <b>Range:</b> 0-9999 <b>Default:</b> 0
out_station3d	Art der Stationierung im Profil No = 2D-horizontal Yes = 3D-schräg <b>Value:</b> boolean <b>Default:</b> No
out_soft_coloring	Die Flächenfüllung soll mit weichen Farbübergängen dargestellt werden. <b>Value:</b> boolean <b>Default:</b> No

---



## Plotboxen

### FeatureType: X

out\_type: out\_box

fme\_type: fme\_area

fme\_geometry: fme\_polygon

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

Der Writer erzeugt Plotboxen aus beliebigen Geometrien, wenn diese nicht den obig genannten Konventionen entspricht, wird die Plotbox aus der Bounding-Box des Features erzeugt.

Attribute Name	Content
out_name	Name der Plotbox, er muss innerhalb eines Projektes eindeutig sein. <b>Range:</b> char(50)

## Features ohne Geometrie

**FeatureType: E** (für alle folgenden Features)

### Löschungen

out\_type: out\_delete

fme\_type: fme\_no\_geom

Geografprojekte, die als Fortführungsauftrag angelegt worden sind, speichern die Löschung von Elementen dauerhaft und geben diese in die GrafBat-Datei aus. Wenn ein Element einen externen Schlüssel aufweist, bleibt ausschließlich dieser erhalten, anderenfalls wird die eingelesene oder beim Import erzeugte GUID gespeichert. Die Geometrie oder eventuell vorhandene Sachdaten sind an bei Löschung nicht mehr vorhanden.

Löschungen werden nur gelesen, wenn der Parameter `READ_OPERATION` auf Yes gesetzt ist.

Attribute Name	Content
out_key	Bei aktuellen Versionen wird der ursprüngliche Elementschlüssel der gelöschten Geometrie gespeichert. (nur Reader) <b>Value:</b> integer <b>Default:</b> No default
out_file_key	Ist immer mit DEL belegt. <b>Value:</b> char(12) <b>Default:</b> DEL
out_extkey out_uuid	Der Schlüssel des Elements. Es sind jeweils entweder out_extkey oder out_uuid belegt. Beim Schreiben hat die out_uuid Vorrang. <b>Value:</b> string <b>Default:</b> No default
out_complete	Objekte werden zusammen mit ihren Inhalten gelöscht, wenn out_complete auf Yes gesetzt ist. <b>Value:</b> boolean <b>Default:</b> No
fme_db_operation	Ist hier immer mit 'DELETE' belegt. <b>Value:</b> char(8) <b>Default:</b> DELETE

## Umnummerierungen

out\_type: out\_renumber

fme\_type: fme\_no\_geom

Umnummerierungen exportiert Geograf selbst nicht, sie können beim Import von Daten angelegt werden.

Beim Import muss die alte Nummer in einem bestehenden Projekt oder in den Importdaten selbst vorhanden sein. Die neue Nummer darf noch nicht bestehen.

Attribute Name	Content
out_file_key	Ist immer mit PKNUM belegt. (nur Reader) <b>Value:</b> char(12) <b>Default:</b> PKNUM
out_number1	Punktnummer des vorhandenen Punktes <b>Value:</b> char(15)
out_number2	neue Punktnummer <b>Value:</b> char(15)

## Tabellen

out\_type: out\_table

fme\_type: fme\_no\_geom

Bei eingeschalteten Parameter `READ_TABLE` erzeugt der Reader aus den Sub- oder Picklistentabellen Features ohne Geometrie. Diese enthalten die Tabelleneigenschaften als Formatattribute und als 'User Attributes' die Inhalte der Tabellen.

Die elementbezogenen Tabellen werden hier nicht gelesen, deren Inhalte werden den Features zugeordnet.

Das Lesen der Tabellen ist relevant, wenn unter Geograf angelegte Sachdatenstrukturen dokumentiert oder in andere Systeme übertragen werden sollen. Der Parameter `READ_TABLE` ist per Default auf No gesetzt.

Das Schreiben von Subtabellen und Picklisten ist dann angebracht, wenn Wertetabellen als CSV- oder XML-Dateien vorliegen und mit diesen ein Geograf-Sachdatenmodell erzeugt werden soll.

Auch können so Tabellen geschrieben werden, die Punktnummern, Objektnamen oder externe Schlüssel zur späteren Verknüpfung mit den Geograf-Elementen enthalten.

Attribute Name	Content
out_key	Der Elementschlüssel der Tabelle (nur Reader) <b>Value:</b> integer <b>Default:</b> No default
out_file_key	Ist immer mit FTable belegt. (nur Reader) <b>Value:</b> char(12) <b>Default:</b> FTable
out_table_name	Name der Tabelle, er muss innerhalb eines Projektes eindeutig sein. <b>Value:</b> string <b>Default:</b> No default
out_table_nr	Nummer der Tabelle, sie muss innerhalb eines Projektes eindeutig sein. <b>Range:</b> 1-9998 <b>Default:</b> No default
out_table_typ	Typ der Tabelle S = sub, keine Zuordnung zum Grafikelement P = picklist, Daten werden als Pickliste genutzt Dem Writer können auch die Typen mit 'sub, pic' angezeigt werden. <b>Value:</b> char(1) <b>Default:</b> P
out_attribute{}.name	Die Namen der in der Tabelle vereinbarten Sachdatenattribute. <b>Value:</b> string <b>Default:</b> No default
out_attribute{}.type	Die Typen der in der Tabelle vereinbarten Sachdatenattribute. <b>Value:</b> string <b>Default:</b> No default

## 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.

Werden geometrische Features als Fehlermeldung ausgegeben ohne dass explizit eine Koordinate mit out\_x, out\_y oder stp\_x, stp\_y angegeben, erzeugt der Writer selbständig eine Koordinate im Zentrum der Geometrie.

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>Value:</b> string <b>Default:</b> DATASET + '.stp'
out_x, out_y oder stp_x, stp_y	Fangpunktkoordinaten der Fehlermeldung, beim Abarbeiten im Geograf-Fehlermanager zeigt der Cursor auf diese Koordinaten. Sind diese nicht angegeben, wird eine Koordinate aus der Featuregeometrie erzeugt. <b>Value:</b> double <b>Default:</b> No default
out_error_name oder out_name oder stp_name	Name des Fehlers <b>Value:</b> string <b>Default:</b> <Geometriebezeichnung>
out_error_msg oder out_msg oder stp_msg	Fehlermeldung <b>Value:</b> string <b>Default:</b> 'ohne Meldung'

## Katalogdaten

Der Reader kann neben den GrafBat-Dateien auch folgende Dateien aus den Katalogdaten lesen:

- Artdateien im ArtBat-Format (\*.out)
- Stiftdateien (\*.pen)
- Dateien der Bildschirmfarben (\*.col)
- Leveldateien (\*.lvl)

Der Reader erkennt diese Dateien an dem Dateinhalt unabhängig von der Erweiterung.

Soweit die Katalogdateien als Quelldatei angewählt werden, erzeugt der Reader geometrieloze Features vom Type `out_rule`. Die Features können anhand des Attributes `out_type` klassifiziert werden und lassen sich anhand der Katalog-, View- und Artnummer den eigentlichen Geometrie-Features zuordnen.

Der Reader kann beim Lesen einer GrafBat-Datei auch die dort verwendeten Katalogdaten zusätzlich lesen. Die Dateien können manuell mit den Parametern:

`ART_SETTING_FILE`

`PEN_SETTING_FILE`

`LEVEL_SETTING_FILE`

`COLOR_SETTING_FILE`

ausgewählt werden. Bei diesen Parametern kann auch jeweils ein Sternchen '\*' angegeben werden, der Reader liest dann die in den Blattparametern angegebenen Artdateien nach und soweit dort auch Stift-, Farb- und Leveldatei angegeben sind, auch diese.

Der Reader übernimmt die ausgewählten Informationen und fügt sie den eigentlichen Geometriedaten hinzu. 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 in die Geometriefeatures 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 Artdatei 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 die Attribute:

`art_*_scr_color`

Beim Nachladen der Katalogdaten können diese zusätzlich als selbständige Feature gelesen werden, dazu ist der Parameter `READ_RULES_AS_FEATURE` auf `Yes` zu setzen.

## Alle Arten

out\_type: out\_rule

Die folgenden Attribute erhalten alle Arten, die als separates Feature gelesen werden. Sie dienen auch als Schlüssel bei der Verknüpfung mit den Grafik-Elementen.

Attribute Name	Content
art_key	Schlüsselnummer der Art, sie ist innerhalb eines Views eindeutig <b>Value:</b> integer
art_id	Die Nummer der Art, der Farbe, des Stiftes, des Levels <b>Range:</b> 0-9999
art_type	Kennung des Artenelements (art_point, art_line, ...) <b>Value:</b> char(20)
art_katalog	Nummer des Kataloges, in der die Art gespeichert ist. <b>Range:</b> 0-9999
art_view	Nummer des Views, in der die Art gespeichert ist. <b>Range:</b> 0-9999

Die folgenden Attribute werden nur bei den Geometriefeatures gespeichert, die mit einer Art verknüpft wurden. Es werden aus der zum Element tatsächlich vergebenen Katalog-, Ebenen- und Artnummer die dort vereinbarten Werte geholt.

Attribute Name	Content
art_katalog_alias	Alias des Kataloges <b>Value:</b> char(4)
art_katalog_name	Bezeichnung des Kataloges <b>Value:</b> String
art_katalog_scr_color	Farbe des Kataloges <b>Value:</b> char(20) (RGB-Farbwerte)
art_view_name	Bezeichnung des Views <b>Value:</b> String
art_group_name	Name der Ebenengruppe <b>Value:</b> char(55)
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> char(20) (RGB-Farbwerte)
art_layer_pen_mode	Stiftmodus des Abschluss einer Linie 0 = rund 1 = eckig <b>Range:</b> 0-1
art_layer_color	Farbnummer der Ebene (Bildschirmfarbe) <b>Range:</b> 0-9999

Attribute Name	Content
art_layer_scr_color	Farbe der Ebene (Bildschirmfarbe) <b>Value:</b> char(20) (RGB-Farbwerte)
art_color	Farbnummer der Art (Bildschirmfarbe) <b>Range:</b> 0-9999
art_scr_color	Farbe der Ebene (Bildschirmfarbe) <b>Value:</b> char(20) (RGB-Farbwerte)
art_pen_size	Stärke des Stiftes <b>Value:</b> float
art_pen_color	Farbe des Stiftes <b>Value:</b> char(20) (RGB-Farbwerte)
art_pen_mode	Stiftmodus des Abschluss einer Linie 0 = rund 1 = eckig <b>Range:</b> 0-9999



## Katalog

art\_type: art\_katalog

Attribute Name	Content
art_alias	Kurzbezeichnung des Katalogs <b>Value:</b> char(5)
art_description	Langtext zur Beschreibung des Katalogs <b>Value:</b> string
art_version	Versionsinfo <b>Value:</b> string
art_copyright	Copyright-Info <b>Value:</b> string
art_scr_color	Katalogfarbe <b>Value:</b> char(20)

## Dateien

art\_type: art\_file

Attribute Name	Content
art_filename	Name der Symbol-, Stift-, Farb- oder Leveldatei <b>Value:</b> string

## Views

In einem View liegen individuell auf einen Maßstabsbereich angepasste Arten.

art\_type: art\_view

Attribute Name	Content
art_start_scale	Startmaßstabszahl <b>Range:</b> 0-9999
art_end_scale	Endmaßstabszahl <b>Range:</b> 0-9999
art_description	Bezeichnung <b>Value:</b> string
art_factor	Faktor des Views, der beim Anlegen verwendet wurde <b>Value:</b> float

## Gruppen

art\_type: art\_group

Attribute Name	Content
art_name	Bezeichnung <b>Value:</b> char(55)

## Ebenen

art\_type: art\_layer

Attribute Name	Content
art_name	Bezeichnung <b>Value:</b> char(55)
art_color	Nummer der Bildschirmfarbe <b>Range:</b> 0-255
art_pen	Nummer des Stiftes <b>Range:</b> 0-9999
art_group	Nummer der Gruppe zu der die Ebene gehört <b>Range:</b> 0-9999

## Punktarten

art\_type: art\_point

Attribute Name	Content
art_symbol	Symbolnummer <b>Range:</b> 0-9999
art_size	Symbolgröße in [mm] <b>Value:</b> float
art_pen	Nummer des Stiftes <b>Range:</b> 0-9999
art_color	Nummer der Bildschirmfarbe <b>Range:</b> 0-255
art_name	Bezeichnung <b>Value:</b> char(55)
art_absolute	Symbolwinkel ist: true=absolut, false=relativ <b>Value:</b> boolean
art_rotation	Symbolwinkel <b>Value:</b> float
art_draw_mode	Zeichnungsmodus: 0 = Schraffur-Deckend 1 = Schraffur- und Linien-Deckend 2 = VollDeckend 3 = VollTransparent <b>Range:</b> 0-3
art_layer	Default-Ebene <b>Range:</b> 0-9999
art_level	Plotstufe <b>Range:</b> 0-9999
art_pnr_mode	Punktnummer bei der Erzeugung: 0 = nicht eindeutig 1 = nicht eindeutig (als Info) 2 = eindeutig <b>Range:</b> 0-2
art_displace_mode	Verdrängungsmodus: 0 = nein 1 = ja <b>Range:</b> 0-1
art_nbz	Nummerierungsbezirk <b>Value:</b> string
art_displaceable	Symbol kann verdrängt werden <b>Value:</b> boolean

## Linienarten

art\_type: art\_line

Attribute Name	Content
art_color	Nummer der Bildschirmfarbe <b>Range:</b> 0-255
art_name	Bezeichnung <b>Value:</b> char(55)
art_draw_mode	Zeichnungsmodus 0 = deckend 1 = transparent <b>Range:</b> 0-1
art_layer	Default-Ebene <b>Range:</b> 0-9999
art_level	Plotstufe <b>Range:</b> 0-9999
art_displace_mode	Verdrängungsmodus: 0 = Linien gleicher Geometrie und kleinerer Stufe nicht verdrängen 1 = Linien gleicher Geometrie und kleinerer Stufe verdrängen <b>Range:</b> 0-1
art_alias	interne ID <b>Value:</b> string
art_displaceable	Linie kann verdrängt werden <b>Value:</b> boolean
art_form{}.form	Formnummer <b>Range:</b> 0-9999
art_form{}.distance	Abstand der Form von der Grundlinie <b>Value:</b> float
art_form{}.pen	Stiftnummer der Form <b>Range:</b> 0-9999

## Linienbegleitende Symbole

art\_type: art\_line\_symbol

Attribute Name	Content
art_symbol	Symbolnummer <b>Range:</b> 0-9999
art_pen	Nummer des Stiftes <b>Range:</b> 0-9999
art_size	Symbolgröße in [mm] <b>Value:</b> float
art_rotation	Symbolwinkel <b>Value:</b> float
art_name	Bezeichnung <b>Value:</b> char(55)
art_absolute	Symbolwinkel ist: Yes=absolut, No=relativ <b>Value:</b> boolean
art_draw_mode	Zeichnungsmodus: 0 = Schraffur-Deckend 1 = Schraffur- und Linien-Deckend 2 = VollDeckend 3 = VollTransparent <b>Range:</b> 0-3

## Linienmuster

art\_type: art\_line\_form

Attribute Name	Content
art_loc_mode	Lage der Linienform: 0 = Ausrichtung zum Anfangspunkt der Linie 1 = Ausrichtung zum Mittelpunkt 2 = Ausrichtung zum Anfangs- und Endpunkt 3 = Ausrichtung zum Endpunkt <b>Range:</b> 0-3
art_name	Bezeichnung <b>Value:</b> char(55)
art_factor	Darstellungslänge in [%] <b>Value:</b> float
art_alias	interne ID <b>Value:</b> string
art_sequence	Definition des Linienmusters <b>Value:</b> string

## Textarten

art\_type: art\_text

Attribute Name	Content
art_slant	Textneigung in Grad <b>Value:</b> float
art_size	Textgröße in [mm] <b>Value:</b> float
art_factor	Verhältnis Breite/Höhe in Prozent <b>Value:</b> float
art_pen	Nummer des Stiftes <b>Range:</b> 0-9999
art_color	Nummer der Bildschirmfarbe <b>Range:</b> 0-255
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_name	Bezeichnung <b>Value:</b> char(55)
art_absolute	Textwinkel ist: Yes=absolut, No=relativ <b>Value:</b> boolean
art_font	Name des Fonts <b>Value:</b> char(40)
art_arrow_size	Pfeilgröße für Bezug in mm <b>Value:</b> float
art_pos_absolute	Schwerpunkt ist fest <b>Value:</b> boolean
art_draw_mode	Zeichenmodus 0 = SchraffurDeckend 1 = Schraffur- und LinienDeckend 2 = VollDeckend 3 = VollTransparent <b>Range:</b> 0-3 <b>Default:</b> 0
art_layer	Default-Ebene <b>Range:</b> 0-9999

Attribute Name	Content
art_level	Plotstufe <b>Range:</b> 0-9999
art_bold	TrueTypeFont fett <b>Value:</b> boolean
art_border	Linienart für den Textrahmen <b>Range:</b> 0-99999
art_line	Linienart für Bezugsstrich <b>Range:</b> 0-99999
art_area	Schraffurart für Texthintergrund <b>Range:</b> 0-99999
art_cross	Linienart für die Textdurchstreichung <b>Range:</b> 0-99999
art_rahmen	Besondere Hervorhebung durch Textrahmen 0 = ohne Rahmen, Bezug vom Zentrum 1 = Kasten, Bezug von den Ecken 2 = Kreis, Bezug vom Zentrum 3 = einfach unterstrichen, Bezug vom Unterstrich 4 = doppelt unterstrichen, eng, Bezug vom Unterstrich 6 = ohne Rahmen, Bezug von den Ecken 7 = doppelt unterstrichen, breit Bezug vom oberen Unterstrich 8 = kleiner Kreis, Bezug vom Zentrum <b>Range:</b> 0-8 <b>Default:</b> 0

## Schraffurarten

art\_type: art\_area

Attribute Name	Content
art_name	Bezeichnung <b>Value:</b> char(55)
art_typ	Schraffurtyp der Fläche 0 = Normale Schraffur 1 = Kreuzschraffur 2 = Rasterschraffur 3 = Flächenfüllung bzw. Saum 6 = deckende Rasterschraffur <b>Range:</b> 0-3, 9 <b>Default:</b> 0
art_sub_art	Linienart bzw. Punktart, je nach Schraffurtyp <b>Range:</b> 0-9999
art_absolute	Schraffurwinkel ist: Yes=absolut, No=relativ <b>Value:</b> boolean
art_rotation	Symbolwinkel <b>Value:</b> float
art_size	Abstand der Schraffur in [mm] <b>Value:</b> float
art_layer	Default-Ebene <b>Range:</b> 0-9999
art_color	Nummer der Bildschirmfarbe <b>Range:</b> 0-255
art_pen	Nummer des Stiftes <b>Range:</b> 0-9999
art_transparent	Transparenz <b>Value:</b> boolean
art_level	Plotstufe <b>Range:</b> 0-9999
art_draw_mode	Plotreihenfolge bei Typ=3 0 = in Konkurrenz zu Schraffuren zeichnen 1 = in Konkurrenz zu Flächen/Säumen zeichnen 2 = in Konkurrenz zu Punkten/Linien/Texten zeichnen <b>Range:</b> 0-2



## Objektarten

art\_type: art\_object

Attribute Name	Content
art_color	Nummer der Bildschirmfarbe <b>Range:</b> 0-255
art_pen	Nummer des Stiftes <b>Range:</b> 0-9999
art_level	Plotstufe <b>Range:</b> 0-9999
art_oberobject	Nummer der Oberobjektart <b>Range:</b> 0-9999
art_name	Bezeichnung <b>Value:</b> char(55)
art_point	Nummer der Standard-Punktart <b>Range:</b> 0-9999
art_line	Nummer der Standard-Linienart <b>Range:</b> 0-99999
art_text	Nummer der Standard-Textart <b>Range:</b> 0-99999
art_add_point	Nummer der Zusatz-Punktart <b>Range:</b> 0-99999
art_add_point_layer	Nummer der Ebene für Zusatzpunkte <b>Range:</b> 0-99999
art_add_text	Nummer der Zusatz-Textart <b>Range:</b> 0-99999
art_add_text_layer	Nummer der Ebene für Zusatztexte <b>Range:</b> 0-99999
art_layer	Default-Ebene <b>Range:</b> 0-9999
art_draw_mode	Zeichnungsmodus 0 = deckend 1 = transparent <b>Range:</b> 0-1
art_name_mode	Speicherung des Objektnamens 0 = nicht speichern 1 = speichern, Name ist nicht eindeutig 2 = speichern, Name ist eindeutig <b>Range:</b> 0-2
art_name_praefix	Präfix für Objektnamen <b>Value:</b> char(12)
art_area	Nummer der Standard-Schraffurart <b>Range:</b> 0-99999
art_area_scr	Nummer der Anzeige-Schraffurart <b>Range:</b> 0-99999

Attribute Name	Content
art_show_mode	Plotreihenfolge: 0 = hierarchische Anzeige hinter der restlichen Grafik 1 = gemeinsame Anzeige mit Flächen 2 = gemeinsame Anzeige mit Schraffen 3 = gemeinsame Anzeige mit Punkte, Linien, Texte <b>Range:</b> 0-3
art_geom_mode	Objektgeometrie 0 = nicht flächenförmig 1 = flächenförmig möglich <b>Range:</b> 0-1

## Plotfarben

art\_type: art\_pen\_color

Die Farbanteile werden in der Stiftdatei in der Regel mit Werten von 0-255 gespeichert. Mit SPMODE kann dort ein anderer maximaler Wert angegeben werden.

Attribute Name	Content
art_typ	Farbsystem 0 = RGB 2 = CYM 3 = HLS <b>Range:</b> 0-3
art_c1	RGB=Rot, CYM=Cyan, HLS=Farbwinkel <b>Range:</b> 0-255
art_c2	RGB=Grün, CYM=Gelb, HLS=Helligkeit <b>Range:</b> 0-255
art_c3	RGB=Blau, CYM=Magenta, HLS=Sättigung <b>Range:</b> 0-255
art_c4	RGB, CYM=Schwarz <b>Range:</b> 0-255
art_pen_color	RGB-Farbe <b>Value:</b> char(20)

## Stifte

art\_type: art\_pen

Attribute Name	Content
art_size	Breite im [mm] <b>Value:</b> Fließkommawert
art_color	Farbnummer der Plotfarbe <b>Range:</b> 1-10000
art_mode	Linienabschluss 0 = rund 1 = eckig <b>Range:</b> 0-1

## Bildschirmfarben

art\_type: art\_scr\_color

Attribute Name	Content
art_scr_color	RGB-Farbe <b>Value:</b> char(20)

## Stufen

art\_type: art\_level

Attribute Name	Content
art_sequence_pa	Punktarten, die diesem Level zugeordnet sind. Der String enthält eine kom-maseparierte Liste von Einzelarten oder Artenbereichen, z.B.: „1001,1002,1010-1090,2000,3000-3999“ Value: string
art_sequence_la	Linienarten, die diesem Level zugeordnet sind. Value: string
art_sequence_ta	Textartenarten, die diesem Level zugeordnet sind. Value: string
art_sequence_aa	Schraffurarten, die diesem Level zugeordnet sind. Value: string
art_sequence_oa	Objektarten, die diesem Level zugeordnet sind. Value: string

## Sachdaten

Geograf ermöglicht die Erfassung von Sachdaten für Punkte, Linien und Splines, Texte und Objekte. 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 bzw. DB3.

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

Bis zur Version 2018 wurde der Typ 'date' als String übergeben, Geograf sah hierfür keinen speziellen Attributtyp vor. Dieser wurde mit der Geograf-Version 10 eingeführt und wird auch dementsprechend von Reader und Writer interpretiert. Die GrafBat-Datei enthält verschieden formatierte Zeichenketten, je nach dem, ob eine Uhrzeit mit angegeben wurde: ein Datum wird als '01.09.2009' gespeichert, Datum und Uhrzeit als '2009-19-01 12:13:14'. Der Reader erzeugt aus beiden Varianten das Standard FME Date Format 'YYYYMMDD' oder das Datetime Format 'YYYYMMDDhhmmss'. Auf der Geograf-Seite besteht weiterhin die Einschränkung, dass nur die Jahresangaben 1900 bis 2027 verarbeitet werden können, der Writer erzeugt eine Fehlermeldung beim Verlassen dieses Bereiches.

## Reader

### Sachdatentabellen

Mit den Standardeinstellungen erzeugt der Reader die unter Geograf vereinbarten Attributnamen mit dem vorangestellten Tabellennamen, z.B.:

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

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

```
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:

*Überschreibe Attributinhalt infolge MERGE\_DATA\_ATTRIBUTES*

Mit dem Keyword `DATA_ATTRIBUTE_PREFIX` kann ein zusätzlicher Präfix vereinbart werden, z.B.:

```
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` = Default 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. Bei `READ_VOID_DATA = Null` erzeugt der Reader die Attribute ohne Wert.

## Subtabellen und Picklisten

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 Attributwerte 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 gesetztem `MERGE_DATA_ATTRIBUTES` entstehen dann folgende Attribute:

```
MERGE_DATA_ATTRIBUTES Yes

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

## Listenattribute

Um Listenattribute 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, mit Komma getrennt erfasst werden:

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

Unter der Reader-Parametern müssen dann `READ_DATA_LIST`, `DATA_LIST_POSTFIX` und `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
READ_DATA_LIST Yes
DATA_LIST_POSTFIX ()
DATA_LIST_DELIM ,

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.

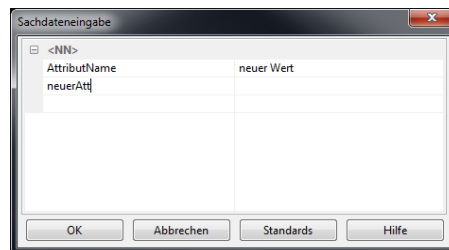
```

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

```

## Undefinierte Sachdaten vom Typ <NN>

Geograf kennt den besonderen Sachdatentyp <NN>. Dieser fest mit der Sachdatennummer 9999 verbundener Tabellename ermöglicht die individuelle Vergabe von Attributen an Geograf-Elemente. Die Geograf-Sachdateneingabe öffnet einen leeren Eingabedialog, bei dem Attributnamen und -werte individuell vergeben werden können.



Die Attributwerte sind hier immer vom Typ `char(255)`.

Der Reader liest diese Attribute. Da diese individuell je Feature angelegt sind, werden sie nicht im Feature-Schema aufgeführt. Das Listenattribut `out_data{}` erhält einen Eintrag:

```
out_data{*}           'NN'
```

das Protokoll enthält eine Warnung:

*Daten gg.out enthalten undefinierte Sachdaten vom Typ: <NN>, diese sind nicht als 'User Attributes' registriert.*

Weiterhin werden einmalig je Attributkombination die enthaltenen Attributnamen als Information protokolliert. Um solche Attribute in der Workbench verarbeiten zu können, müssen diese mit dem *AttributeExposer* und *Tester* detektiert werden.

Bei ausgeschalteten `MERGE_DATA_ATTRIBUTES` erhalten die Attribute den Präfix 'NN.'

## 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    \

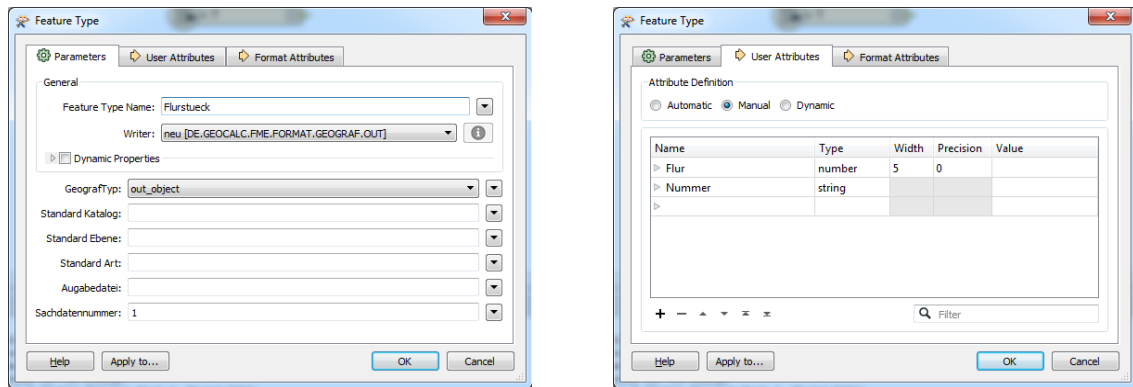
```

Sachdatensätze unter Geograf besitzen einen eindeutigen Namen und eine eindeutige Nummer. Der Sachdatensatzname entspricht dem FeatureType, die Sachdatensatznummer kann mit dem Schemaattribut `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.

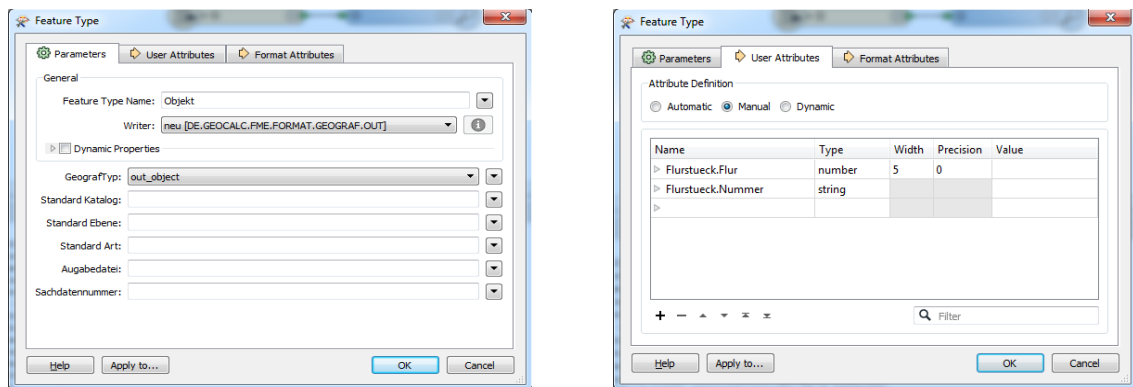
### Einfache Sachdaten

In der Regel enthält ein Feature einen Satz an Attributen. Diese werden unter dem Reiter 'User Attributen' 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 Ausgabe-Features verwendet werden.

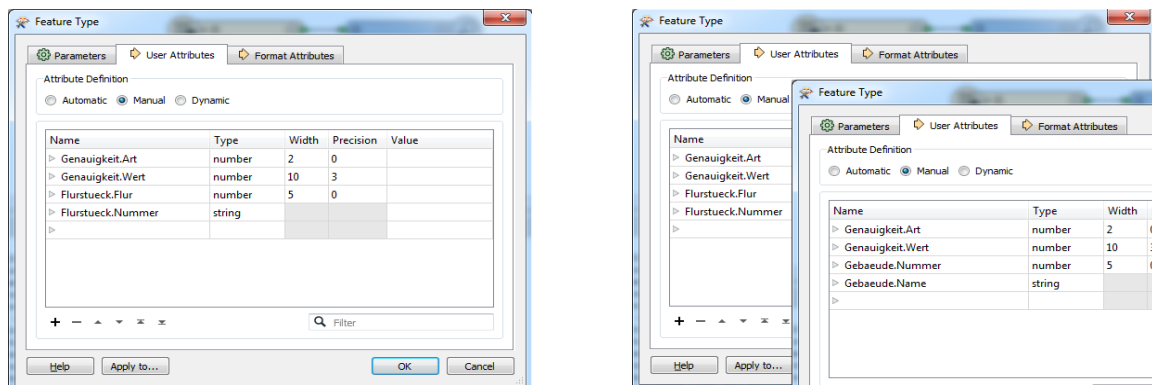
Die Featureattribute müssen selbst nicht diese Langform aufweisen, es reicht als Attributname der letzte Namensteil. Im obigen Beispiel werden die Featureattribute:

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

oder

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

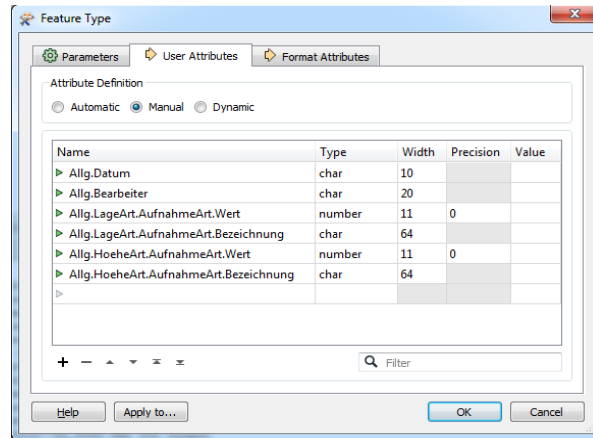
als Sachdatum in die Tabelle Flurstueck geschrieben, Vorrang hat der qualifizierte Attributname.





## Subtabellen und Picklisten

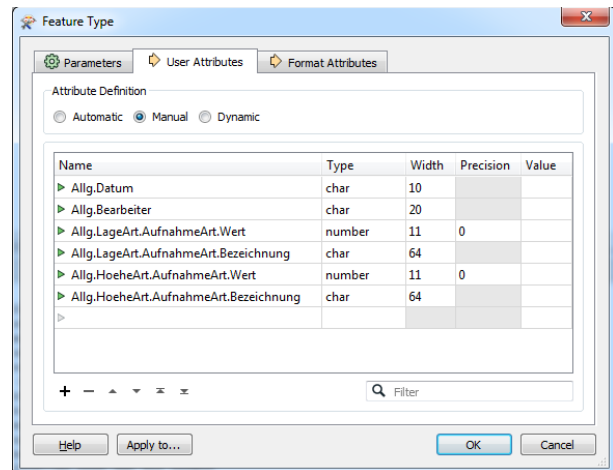
Auf diese Art und Weise können auch SubTabellen oder Picklisten vereinbart werden.



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 Ausgabe-Features mit identischen Sachdatentabellen angelegt werden sollen, braucht die Tabelle nur einmalig angelegt zu werden. Dies kann auch über ein Ausgabe-Feature 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'
```

## Undefinierte Sachdaten vom Typ <NN>

Die Sonderform der undefinierten Sachdaten vom Typ <NN> werden dann vom Writer erzeugt, wenn ein Feature in der Liste `out_data{}` einen Eintrag 'NN' enthält. Es werden dann alle nicht leeren Attribute erzeugt, deren Attributname der Präfix 'NN.' vorangestellt ist.

## Template-Datei

Soweit in Geograf die Sachdatentabellen bereits definiert sind, kann die Definition der Sachdaten aus einem GrafBat-Template gelesen werden. Auf die Definition der Feature-Attributes kann dann komplett verzichtet werden. Den Features muss dann lediglich 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.

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 geschrieben, die tatsächlich in den Wertetabellen vorgegeben sind. Hier ein Beispiel:

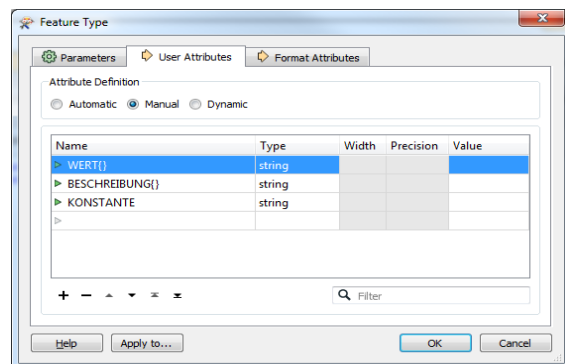
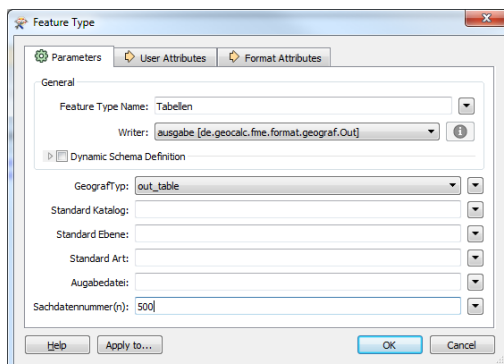
```
*
Typ: GRAFBAT V10
*
* Vorlage Stadtkarte
*
BLATT: "GIS","F500.art","din.sym","Hp.ins","GIS.mde",500,0,"",",",,"g_color.col"
*
FTable: Zustand,1104,P,DBId[da],Wert[d],Bez[s50]
FInfo: Zustand,"DBId:1","Wert:1000","Bez:in Betrieb"
FInfo: Zustand,"DBId:2","Wert:1200","Bez:stillgelegt"
FInfo: Zustand,"DBId:3","Wert:2000","Bez:im Bau"
FInfo: Zustand,"DBId:4","Wert:8000","Bez:in Planung"
FInfo: Zustand,"DBId:5","Wert:9000","Bez:beseitigt"
FInfo: Zustand,"DBId:6","Wert:9100","Bez:falsch erfasst"*
*
FTable: GebaeudeBauteilFunktion,2041,P,DBId[da],Wert[d],Bez[s50]
FInfo: GebaeudeBauteilFunktion,"DBId:1","Wert:2001","Bez:Pfeiler"
FInfo: GebaeudeBauteilFunktion,"DBId:2","Wert:2002","Bez:Lichtschacht"
FInfo: GebaeudeBauteilFunktion,"DBId:3","Wert:2003","Bez:Schornstein"
*
FTable: GebaeudeBauteil,204,E,DBId[da],FUNKTION[#2041],Zustand[#1104],BtName[s255]
FTypes: GebaeudeBauteil,Dot,"0[250]"
```

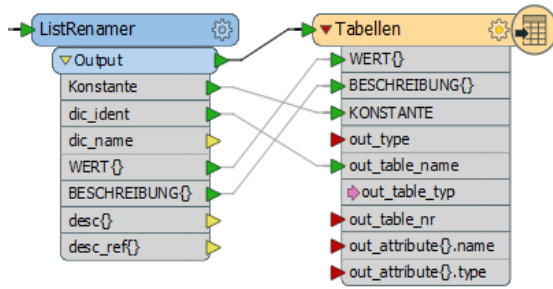
## Subtabellen und Picklisten aus Schemadaten

Datenmodelle geben oft eingeschränkte Inhalte von Attributen vor, die sich unter Geograf als Picklisten realisieren lassen. Üblicherweise werden solche Auswahllisten in CSV- bzw. XML-Dateien definiert. Daraus können mit dem Writer die Tabelleninhalte für Geograf vorbereitet werden.

Die Inhalte der Picklisten und Subtabellen müssen als Listenattribute je zu erzeugender Tabelle vorliegen. Die Definition der Tabellenzeilen kann auf zwei Wegen erfolgen:

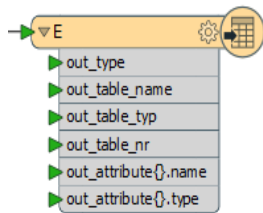
### einheitliches Schema:





Der FeatureType wird als Geograftyp `out_table` ausgegeben. Es kann eine Tabellenummer vorgegeben werden, alle ausgegebenen Tabellen werden hiermit fortlaufend nummeriert. Alle Tabellen erhalten im obigen Beispiel nach dem festgelegte Schema die Spalten 'WERT' und 'BESCHREIBUNG', die Werte werden aus diesen Listen erzeugt. Es können auch konstante Werte ausgegeben werden, in jeder Tabellenzeile wird der konstante Wert gesetzt.

**variables Schema:**



```

E: Feature is:
+++++
Feature Type: 'E_LOGGED'
Attribute(32 bit integer) : 'Art(0)' has value '1001'
Attribute(32 bit integer) : 'Art(1)' has value '1002'
Attribute(32 bit integer) : 'Art(2)' has value '1003'
Attribute(encoded: UTF-8) : 'Bezeichnung(0)' has value 'unbefestigt'
Attribute(encoded: UTF-8) : 'Bezeichnung(1)' has value 'befestigt'
Attribute(encoded: UTF-8) : 'Bezeichnung(2)' has value 'mit Bord'
Attribute(string) : 'fme_feature_type' has value 'E'
Attribute(entangled: string): 'fme_type' has value 'fme_no_geom'
entangled to [out_type]
Attribute(32 bit integer) : 'multi_reader_full_id' has value '0'
Attribute(32 bit integer) : 'multi_reader_id' has value '0'
Attribute(string) : 'multi_reader_keyword' has value 'de.geocalc.fme.format.geograf.Out_1'
Attribute(string) : 'multi_reader_type' has value 'DE.GEOCALC.FME.FORMAT.GEOGRAF.OUT'
Attribute(encoded: UTF-8) : 'out_attribute{0}.name' has value 'Art'
Attribute(encoded: UTF-8) : 'out_attribute{0}.type' has value 'number(11,0)'
Attribute(encoded: UTF-8) : 'out_attribute{1}.name' has value 'Bezeichnung'
Attribute(encoded: UTF-8) : 'out_attribute{1}.type' has value 'char(50)'
Attribute(encoded: UTF-8) : 'out_file_key' has value 'FTable'
Attribute(32 bit integer) : 'out_key' has value '51389625'
Attribute(encoded: UTF-8) : 'out_table_name' has value 'StraßenKanteArt'
Attribute(32 bit integer) : 'out_table_nr' has value '9401'
Attribute(encoded: UTF-8) : 'out_table_typ' has value 'P'
Attribute(encoded: UTF-8) : 'out_type' has value 'out_table'
Geometry Type: Unknown (0)
    
```

Jede auszugebende Tabelle kann ihr eigenes Schema definieren. Die Tabellenspalten werden aus dem Listenattribut `out_attribute{}.name` generiert, mit dem Listenattribut `out_attribute{}.type` kann der Datentyp der Spalte festgelegt werden.

Bei beiden Methoden können Tabellenname, -typ und -nummer über die Attribute `out_table_name`, `out_table_typ` und `out_table_nr` erzeugt werden. Liegen diese nicht vor, werden sie aus dem FeatureType generiert.

Tabellenname und -nummer dürfen nur einmal je Ausgabedatei vorkommen, dies wird durch den Writer sicher gestellt.

**Achtung:**

Die hier auszugebenden Tabellen sollten nicht zusammen mit Grafikelementen ausgegeben werden, wenn diese dort Verwendung finden sollen. In diesem Falle sollte mit der hier beschriebenen Methodik eine `Template-Datei-Datei` generiert werden, die dann bei der Ausgabe der Grafikdaten benutzt wird.

**Tabellen mit externen Schlüsseln**

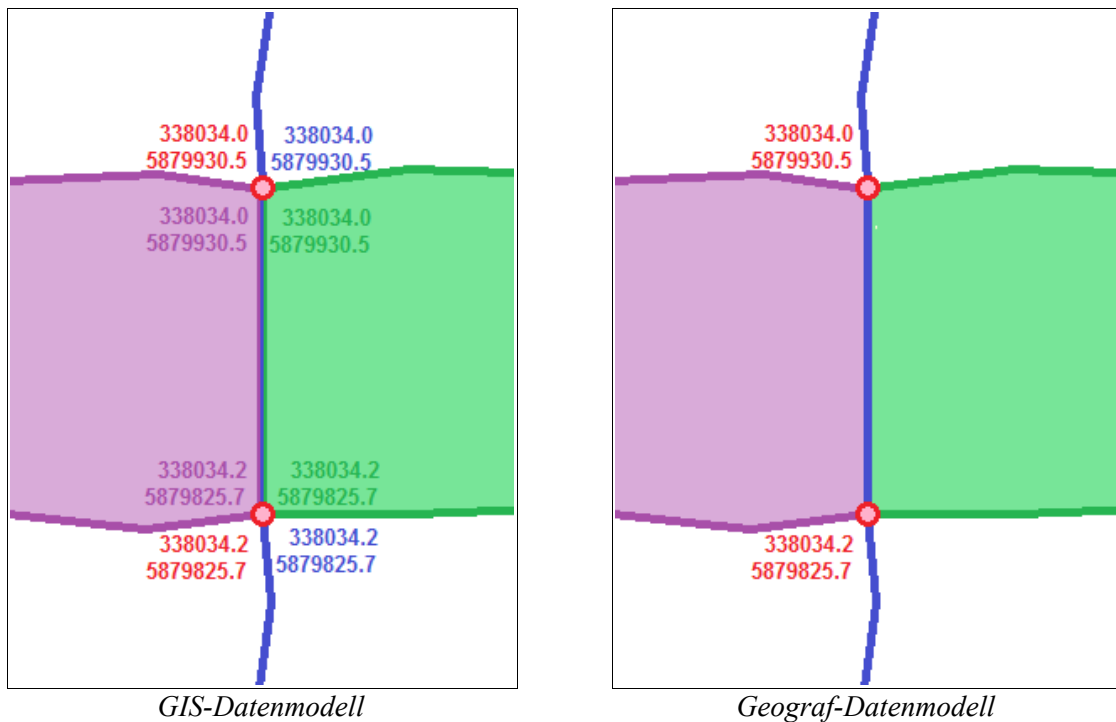
Unter Geograf eingegebene Sachdaten werden automatisch mit den Grafikelementen verknüpft. Es besteht aber auch die Möglichkeit, die Grafikelemente über ein Tabellenfeld zu verknüpfen, möglich sind: externe Schlüssel, Punktnummer, UUID oder eindeutige Objektamen.

Um solche externen Sachdaten nach Geograf einzuspielen ist eine Subtabelle (`out_table_typ = S`) auf dem vorher beschriebenen Weg zu erzeugen und in Geograf einzulesen. In Geograf ist dann in den Sachdatenparametern der Tabellentyp auf 'Einfache Zuordnung zu einem Grafikelement' umzustellen, das Schlüsselfeld und der Typ und die Elementauswahl zu setzen. Geograf erzeugt daraufhin die Verknüpfung zu den Grafikelementen.

Damit ist auch eine Möglichkeit gegeben, Sachdaten zu erzeugen, mit denen ein Geograf-Sachdaten-abgleich durchgeführt werden kann.

## Knoten-Kanten-Modell

In der folgenden Abbildung wird das Datenmodell von typischen GIS-Systemen dem unter Geograf üblichen Datenmodell gegenüber gestellt.

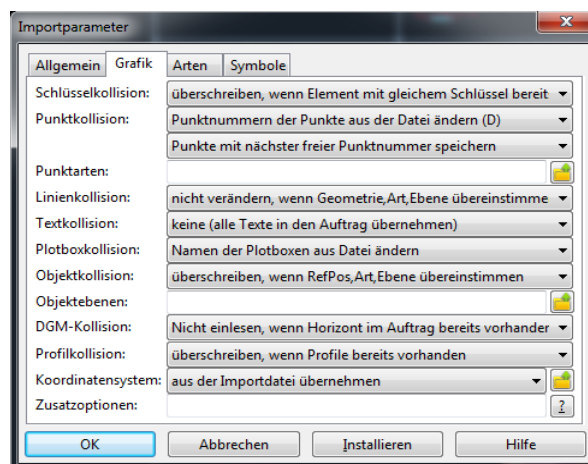


GIS-Systeme speichern zu jedem Feature eine eigene Geometrie, die aus einer oder einer Abfolge von Koordinaten besteht. Die topologischen Beziehungen der Features werden über Werkzeuge oder Routinen sichergestellt.

Unter Geograf ist die grundlegende Geometrie der Punkt, der als separates Element mit eigenen Koordinaten gespeichert ist. Einfache Linien werden als Verbindung zwischen zwei Punkten gespeichert, wobei Referenzen auf den Anfangs- und Endpunkt gesetzt werden. Eigene Koordinaten speichern nur Polylinien, hier aber nur für die zwischen Anfangs- und Endpunkt liegenden Vertex.

Auch linien- oder flächenhafte Objekte speichern neben dem Referenzpunkt keine eigenen Koordinaten, sie verweisen auf die über Punkte definierte Linien.

Der Reader wertet das Geograf-Datenmodell so aus, dass die Referenzen aufgelöst werden und jedem entstehenden Feature eigene Koordinaten zugewiesen werden. Der Writer bereitet die Daten so vor, dass das Geograf-Importmodul GrafBat das übliche Datenmodell erzeugen kann. Die dafür notwendigen Einstellungen sind in diesem Kapitel beschrieben.



*Geograf Importparameter*

## Punkte

Unter Geograf wird der Punkt anhand seiner alphanumerischen Nummer identifiziert, er gilt unabhängig von seiner Koordinate und sonstigen Eigenschaften als identisch, wenn eine gleiche Punktnummer vorliegt.

Der Writer erzeugt Geograf-Punkte als Standard ohne Punktnummer. Diese Punkte gelten beim Einlesen als identisch, wenn gleiche 2D-Koordinaten und gleiche Art und Ebene vorliegen. Mehrere Punkte mit gleichen Werten werden unabhängig von den Importparametern zu einem Punkt verschmolzen. Auch wenn eine nicht eindeutige Punktnummer (out\_name) vergeben ist, erfolgt die Verschmelzung bei identischen Koordinaten und Arten. Hierbei gehen auch eventuell vergebene Sachdaten verloren.

In der Regel ist die Verschmelzung bei identischen Arten erwünscht, gerade wenn die Punkte aus Linien- oder Flächen-Vertex entstehen. Sieht aber das Datenmodell vor, dass gleich attribuierte Punkte übereinander liegen sollen, kann die Verschmelzung unterdrückt werden, in dem eindeutige Punktnummern (out\_number) gegebenenfalls mit einem *Counter* vergeben werden. Auch bei Vergabe einer UUID (out\_uuid) oder eines externen Schlüssels (out\_extkey) erfolgt seitens Geograf keine Verschmelzung.

## Linienpunkte

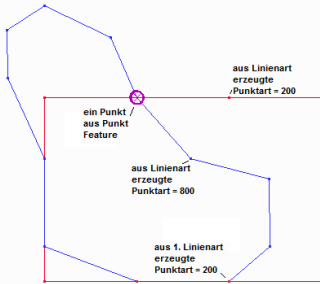
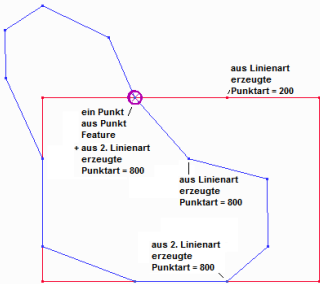
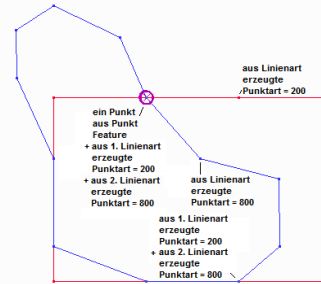
Linien besitzen unter Geograf keine eigenen Koordinaten, sie sind als Verbindung zwischen gesondert gespeicherten Punkten definiert. Der Reader übernimmt diese Verknüpfung nur soweit, dass bei Linien die Punktnummer des Anfangs- und Endpunktes als Attribut gesetzt wird.

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

Der Writer speichert bereits geschriebene Punkte per Default in einem Koordinatenindex zwischen, nachfolgend geschriebene Linienvertex fangen bereits erzeugte Punkte ein. Mit dem Writer-Parameter MERGE\_COORDS = No kann dieses Verhalten abgeschaltet werden, dann erzeugt jeder Linienvertex einen neuen eigenen Punkt.

Zum Schreiben der Linien können die Featureattribute out\_point\_art, out\_point\_ebene und out\_point\_katalog gesetzt werden, dann werden den Linienpunkten diese Werte gegeben. Bei Punkten, auf die mehrere Linien treffen, setzt beim Standard MERGE\_COORDS = Yes die erste geschriebene Linie die Punktwerte. Soweit beim aufeinandertreffen von mehreren Linien Prioritäten gelten sollen, müssen die Linien beim Schreiben sortiert werden. Die Sortierstufe kann beim Ausgabe-Featuretype als Ganzzahl angegeben werden, Features mit größerer Sortierstufe werden später geschrieben, sie fangen dann die Punkte der Features mit geringerer Sortierstufe ein. Die Sortierstufe kann auch individuell mit dem Featureattribut out\_sort vergeben werden.

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

<p><u>MERGE_COORDS</u> = Yes</p> <p>Art der Linienpunkte gleich Linienart:</p> <p><u>out_point_art</u> = <u>out_art</u></p> <p>Sortierung:</p> <ol style="list-style-type: none"> <li>1. Punkte</li> <li>2. Linien nach <u>out_art</u> aufsteigend</li> </ol>	<p><u>MERGE_COORDS</u> = Yes</p> <p>Art der Linienpunkte gleich Linienart:</p> <p><u>out_point_art</u> = <u>out_art</u></p> <p>Sortierung:</p> <ol style="list-style-type: none"> <li>1. Linien nach <u>out_art</u> absteigend</li> <li>2. Punkte</li> </ol>	<p><u>MERGE_COORDS</u> = No</p> <p>Art der Linienpunkte gleich Linienart:</p> <p><u>out_point_art</u> = <u>out_art</u></p> <p>Sortierung: ohne</p>
		

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

## Objektlinien

Wenn Geometriefeatures als Geograf-Objekt `out_object` ausgegeben werden, erzeugt der Writer vorerst die atomaren Geograf-Elemente `Punkte`, `Linien`, `Texte` und fasst diese dann zu einem Objekt zusammen. Die Attribute `out_art`, `out_ebene` und `out_katalog` werden ausschließlich für das Geograf-Objekt vergeben, die Geometriebestandteile erhalten die Defaultwerte 0.

In der Regel sollen die Geometriebestandteile eigene Angaben zu Art, Ebene und Katalog erhalten, diese können auf das Feature bezogen vergeben werden mit:

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

Einzelne Ausgestaltungselemente, typischerweise ein Objektsymbol oder -text können dem Objekt auch als Liste `out_elements{}` mitgegeben werden, die individuellen Arten werden dann aus dieser Attributen dieser Liste erzeugt. Auch die Verlinkung von Geometriebestandteilen mit `out_object{}.link` führt zu individuellen Arten der Geometriebestandteile innerhalb eines Objektes.

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 oben aufgeführten Attribute `out_point_art`, ..., mitgegeben werden, erhalten alle Geometriebestandteile diese Geograf-Arten, für die keine individuellen Arten vergeben sind.

Unter Geograf ist es üblich, dass dort wo Flächenobjekte aufeinandertreffen, die gemeinsamen Kanten nur einmalig als Linien erfasst werden, die dann Bestandteil der Definitionsgeometrie beider Objekte sind. Beispiel sei eine Verkehrsfläche, die durch eine Gebäudekante begrenzt wird. Erfasst würde hier eine Linie mit der Art=Gebäudekante, die dann Definitionsgeometrie des Gebäude- und des Verkehrsflächenobjektes ist.

Der Writer erzeugt aus jedem zu schreibenden Flächenfeature eigene Linien aus dem Umring, jeweils mit der Art, so wie sie bei dem Flächenfeature festgelegt ist. Benachbarte Flächen erzeugen somit doppelte Linien. Identisch attribuierte Linien verschmilzt Geograf automatisch. Treffen unterschiedliche Arten aufeinander, kann beim Geograf-Import festgelegt werden, ob geometriegleiche Linien verschmolzen werden sollen, es stehen hier verschiedene Varianten zur Verfügung.

Da Geograf eine GrafBat-Datei sequentiell einliest, ist die Sortierung der Linien innerhalb der Datei entscheidend. Wird im Modus 'überschreiben' gelesen, entscheidet die letzte auftretende Linie über die verschmolzene Linienart. Im Modus 'nicht verändern' gibt die erste auftretende Linie die Art vor. Wie schon oben beschrieben, kann eine Sortierreihenfolge beim Ausgabe-Featuretype oder auch je Feature individuell festgelegt werden.

**Für die Sortierung der Features erzeugt der Writer eine Pipeline mit einem *Sorter*, der die Features numerisch aufsteigend nach dem Inhalt des Attributes `out_sort` weiterleitet. Features, für die eine Sortierstufe angegeben wurde, werden in diese Pipeline eingefügt, die erst abgearbeitet wird, wenn die Features ohne Sortierung fertig geschrieben wurden.**

Wenn die Verschmelzung von Linien nur für bestimmte Kombinationen geschehen soll, kann dies nach dem Import mit Geograf-Makros ausgeführt werden. Als dauerhafte Lösung bietet es sich an, die dafür notwendigen Makrobefehle in einer Stapeldatei anzulegen, die nach dem Import mit dem Fehlermanager ausgewählt und gestartet wird. Hier ein Beispiel für eine Stapeldatei, die schrittweise die Verschmelzung ausführt:

```
StapelDatei LinienVerschmelzen.stp
CMDf10Stop CMDf10FilterClear CmdLinMerge F1 "1-20100" RET CMDf10ArtenEbenen "1-20100" RET "*" \
RET CMDf10SelectAlles CMDf10SelectStart ESC CMDMacroStapel
CMDf10Stop CMDf10FilterClear CmdLinMerge F1 "1-20200" RET CMDf10ArtenEbenen "1-20200" RET "*" \
RET CMDf10SelectAlles CMDf10SelectStart ESC CMDMacroStapel
CMDf10Stop CMDf10FilterClear CmdLinMerge F1 "1-39999" RET CMDf10ArtenEbenen "1-39999" RET "*" \
RET CMDf10SelectAlles CMDf10SelectStart ESC CMDMacroStapel
CMDf10Stop CMDf10FilterClear CmdLinMerge F1 "1-40200" RET CMDf10ArtenEbenen "1-40200" RET "*" \
RET CMDf10SelectAlles CMDf10SelectStart ESC CMDMacroStapel
```

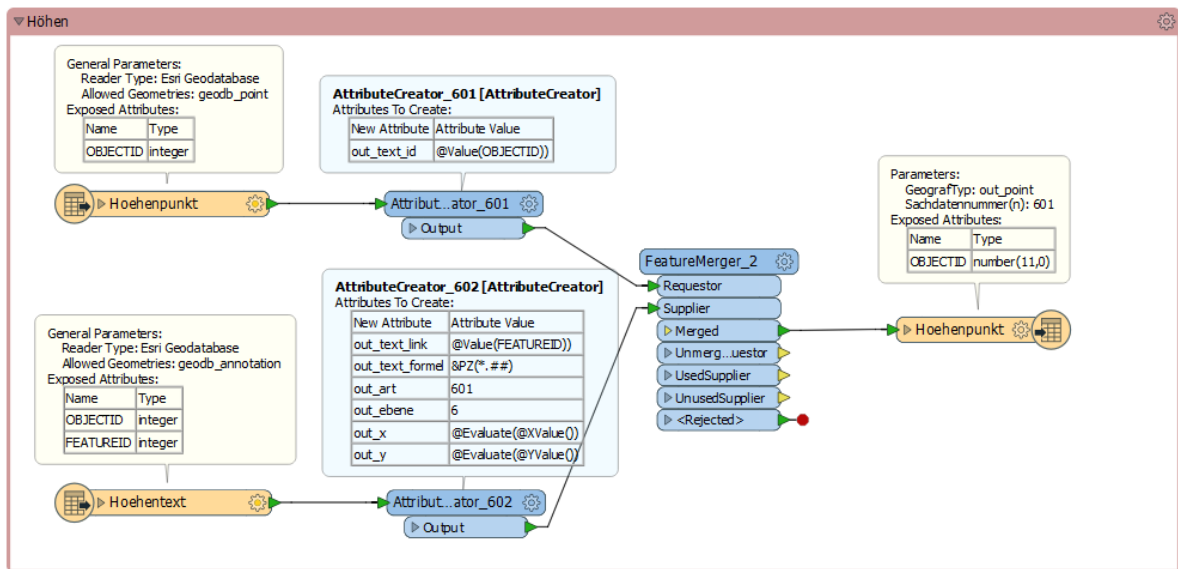
Möglich ist auch, die Verschmelzung bereits mit FME-Mitteln durchzuführen, in dem die gewünschten Linieneigenschaften als Geometrieattribute an die Flächenfeatures angebracht werden. Dies ist aber mit einem nicht unerheblichen Aufwand beim Gestalten der Workbench verbunden.

# Beschriftungen

Geograf-Beschriftungen sind Texte, die eine Geometrie-eigenschaft eines Grafikelement repräsentieren. So werden z.B. Höhentexte als Beschriftung des Z-Wertes eines Punktes erzeugt. Beschriftungstexte werden die Beziehung zum beschrifteten Element dauerhaft, bei Veränderungen der Geometrie wird automatisch der Textinhalt aktualisiert.

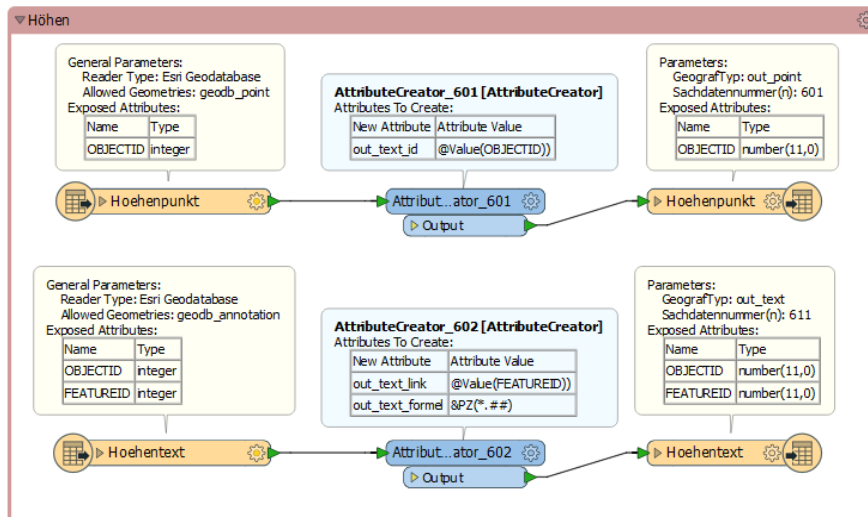
Die Beziehung des Textes zum beschrifteten Element wird auch der Reader aus, gemäß dem Reader-Parameter `TEXT_PARSER` wird entweder zu dem beschrifteten Feature das Listenattribut `out_text` erzeugt oder am Beschriftungstext das Attribut `out_text_link` gesetzt, welches auf das Beschriftungselement verweist.

Beide Varianten werden auch vom Writer unterstützt, die je nach Ausgangslage gewählt werden können. Der Writer erfordert dafür keine besondere Parametereinstellungen, die Beschriftung wird aus den zum Feature gesetzten Attributen `out_text` bzw. `out_text_link` erzeugt.



*Beschriftung als Listenattribut*

Es ist darauf zu achten, dass am Text auch die Textformel `out_text_formel` gesetzt wird, nur dann sind unter Geograf alle Funktionalitäten einer Beschriftung verfügbar.



*Beschriftung als Verlinkung*

In der resultierenden GrafBat-Datei müssen die beschrifteten Elemente immer vor dem Beschriftungstext stehen, ansonsten wird die Verknüpfung nicht aufgebaut. Der Writer erzeugt die dafür notwendige Sortierung selbständig. Wenn manuell Sortierstufen gesetzt werden, ist darauf zu achten, dass die beschrifteten Elemente auf kleinerer oder gleicher Stufe liegen als der Beschriftungstext. Ist dies nicht der Fall, schreibt der Writer den Beschriftungstext erst in der Stufe, die für das beschriftete Element gesetzt ist.

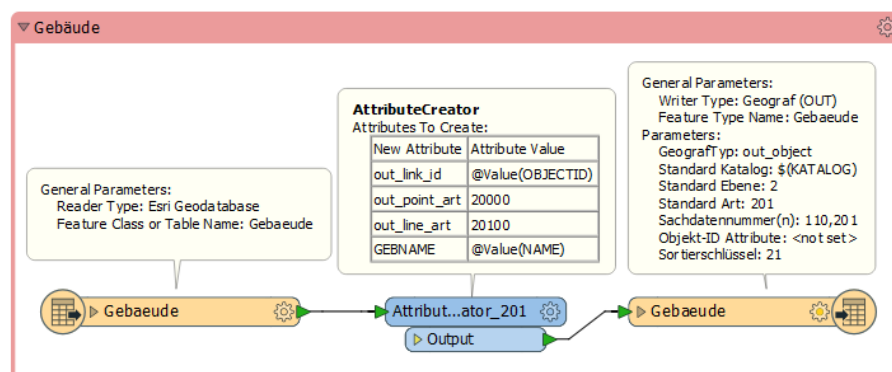


## Fortführung von Geodatenbanken

Geograf ist ein im Vermessungswesen gern eingesetztes CAD-System, weil es die Vermessungskordinate in den Vordergrund und die für Vermessungstechniker üblichen Berechnungs- und Konstruktionsmethoden einfach zugänglich zur Verfügung stellt. Es bietet sich an Geograf auch für die Fortführung von Geodatenbanken einzusetzen, insbesondere dann wenn in der Örtlichkeit erzeugte Vermessungswerte verarbeitet werden. Hier sollen zwei Methoden zur Fortführung von Geodatenbanken am Beispiel einer ESRI-Geodatabase vorgestellt werden.

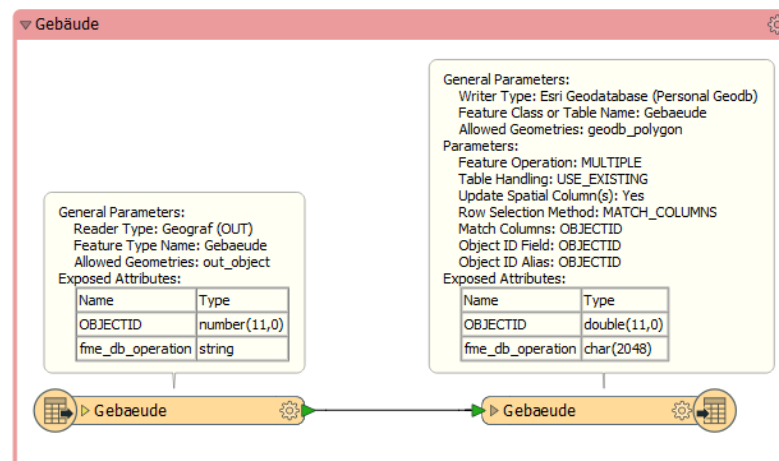
### Datenmodell übertragen

Man tut gut daran, das Datenmodell der Geodatenbank bezüglich der Geometrie und Sachdaten so originalgetreu wie möglich nach Geograf zu übertragen, um so einfacher lassen sich dann die Workbenches gestalten. Der Name des Geograf-Sachdatensatzes sollte dem FeatureType der Datenbank entsprechen, beim Reimport der Geograf-Daten mit dem Parameter `FEATURE_TYPE = SACH` stehen dann gleich die FeatureTypes der Datenbank zur Verfügung. Wenn die Geograf-Sachdaten identisch zu den Datenbankfeldern an- und möglichst in einer `Template-Datei` dauerhaft abgelegt werden, beschränkt sich die Arbeit in der Workbench auf wenige Schritte.



*Feature-Transformation von Geodatabase nach Geograf*

Vor dem Import der umgewandelten Daten nach Geograf ist zu entscheiden, ob ein Projekt im Auftragstyp 'Standard' oder 'Fortführung' angelegt wird. Der Fortführungsmodus bietet den Vorteil, dass physische Löschungen registriert werden. Dann muss aber mit externen Schlüsseln gearbeitet werden, damit die Löschatensätze ein zur Datenbank verbindendes Merkmal behalten.



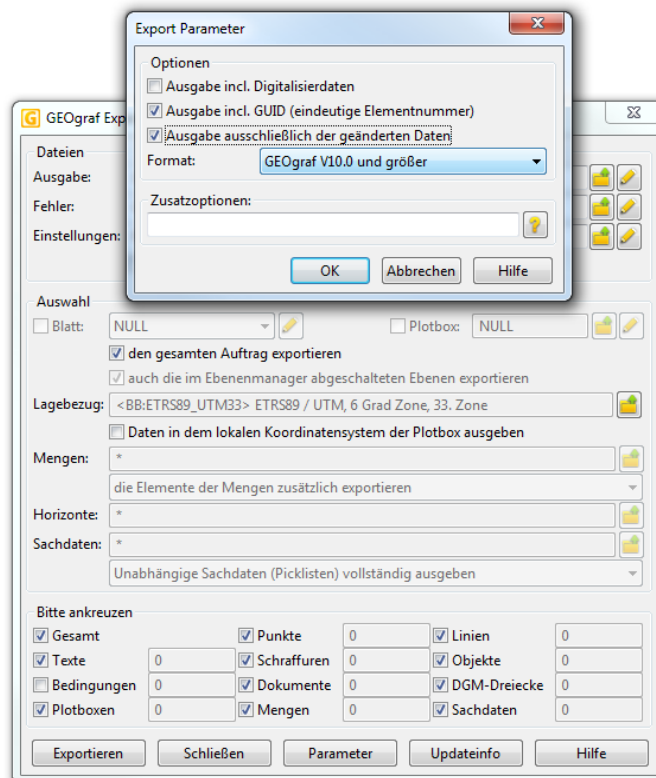
*Feature-Transformation von Geograf nach Geodatabase*

`FEATURE_TYPE = SACH,`  
`READ_OPERATION = YES`  
`READ_UNCHANGE = NO`  
`DELETE_LAYERS = 9000`

## Datenbankschlüssel in Sachdaten

Damit die Features auf dem Rückweg in die Geodatabase wieder den originalen Feature zugeordnet werden kann, muss die ObjektId auch unter Geograf gespeichert werden. Am einfachsten kann das im dem Feature zugeordneten Sachdatensatz geschehen, mit dem Nachteil, dass sie dort versehentlich änderbar ist. Abhilfe kann geschaffen werden, wenn zwei Sachdatentabellen je Geograf-Element angelegt werden, eine als als geschützt gekennzeichnete Tabelle mit der ObjektId und anderen nicht zu verändernden Attributen und eine Tabelle mit Attributen, die im Zuge der Bearbeitung fortgeführt werden sollen.

Ebenso einfach ist der Rückweg der Daten von Geograf in die Datenbank. Einzige Herausforderung ist hier zu erkennen, mit welcher Operation das einzelne Feature in die Datenbank geführt werden soll.



*Export eines Geograf-Projektes mit Fortführungsdaten*

Soweit beim Export die Option 'Ausgabe incl. GUID' aktiviert ist, schreibt Geograf die GUID und die Aktualität jedes Elements. Die GUID wird nicht weiter benötigt. Aus der in `out_akt` gespeicherten Aktualität erzeugt der Reader bei `READ_OPERATION = YES` setzt der Reader das Attribut `fme_db_operation` auf UPDATE bzw. INSERT.

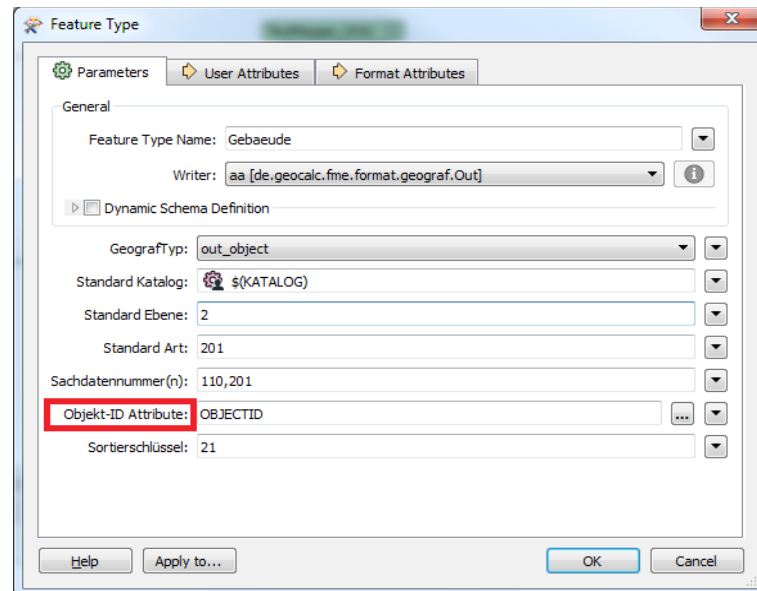
Um Features in der Datenbank zu löschen, dürfen diese unter Geograf nicht physisch gelöscht werden, da hiermit auch die Sachdaten einschließlich des FeatureTypes und der verbindenden ObjektId verloren gehen. Es bietet sich die Vereinbarung einer oder mehrerer Löschebenen an, auf die die zu löschenden Elemente geschoben werden. Bei `READ_OPERATION = YES` und gesetztem `DELETE_LAYERS` setzt der Reader das Attribut `fme_db_operation` auf DELETE.

Die Datenbank-Operation ließe sich auch mit wenigen Tests in einem `AttributCreator` erzeugen.

## Datenbankschlüssel als externer Key

Wenn Elemente bei der Geograf-Bearbeitung physisch gelöscht werden sollen, muss das Projekt im Modus 'Fortführung' vor dem Import angelegt werden. Nur dann werden `Löschungen` registriert.

Bei `Löschungen` gehen sämtliche Informationen zum Element verloren, einzig die GUID oder ein erzeugter externer Schlüssel bleiben erhalten. Damit aus einer Löschung der ursprüngliche FeatureType und die ObjektId wiederhergestellt werden kann, bietet der Writer bei der Feature-Definition mit dem Attribut `out_match_attr` eine besondere Variante der Erzeugung eines externen Schlüssels an.



### *Erzeugung eines externen Schlüssels zur Fortführung*

Wird wenigstens Objekt-ID Attribute gewählt, erzeugt der Writer zu jedem Feature einen externen Schlüssel der Form: `$FeatureType$AttName1=AttValue1<$AttName2=AttValue2$...>`

KEY: "\$Gebaeude\$OBJECTID=222", 0

Die damit im externen Schlüssel untergebrachte ObjektId braucht dann nicht in den Sachdaten gespeichert zu werden.

So erzeugte und unter Geograf bearbeitete Daten sind dann mit den Parametereinstellungen `READ_OPERATION = YES` und `RESOLVE_OPERATION = YES` zu lesen. Für alle Features werden beim Lesen die Objekt-Id-Attribute mit den im externen Schlüssel gespeicherten Inhalten erzeugt. Der gespeicherte FeatureType wird nur für die Löschungen benötigt. Die sonst ohne Informationen verbliebenen Löschungen gelangen dann über den im externen Schlüssel gespeicherten FeatureType in die Workbench, alle anderen Elemente erhalten den FeatureType, so wie er regulär über den Parameter `FEATURE_TYPE` erzeugt wird.

# Beispiele

## Geograf Mapping File Beispiel 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.ffe
#

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

## Geograf Mapping File Beispiel 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 eine Template-Datei 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 Dokumente 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

01.01.2019 Wenn `DATA_ATTRIBUTE_PREFIX` gesetzt wurde, erzeugte der Reader einen Punkt als Trennzeichen nach dem Präfix. Dies war so nicht gewollt und wurde berichtigt.

In den `Sachdaten` kann jetzt auch ein Attribut als Typ 'Datum' bzw. Datum/Uhrzeit vereinbart werden.

Das Lesen und Schreiben von undefinierten `Sachdaten` ist jetzt möglich.

Das Attribut `out_size_prop` wird jetzt auch für Linien erzeugt da sie in Geograf ab Version 10 in Meter oder als Faktor angegeben werden kann.

Der Reader erzeugt das Attribut `out_file_key`.

Der Writer gibt jetzt eine Fehlermeldung aus, wenn die Objektzuordnung von Elementen nicht erfolgt.

Die Referenzierung beim Schreiben von Objekten und Beschriftungen kann jetzt auch über alphanumerische Schlüssel erfolgen.

Der Reader hat bisher auch die auf Disabled gesetzten InputFeatures an die FME übergeben, diese wurden zwar nicht weiter verarbeitet, aber in der Statistik angezeigt. Jetzt werden die DEFs ausgewertet und nur noch die angeschalteten Fetures gelesen. Nicht gelesene Features werden mit einer Warnung protokolliert.

Sub- und Picktabellen können jetzt gelesen und geschrieben werden. Hierzu wurde der Parameter `READ_TABLE` eingeführt.

10.01.2019 Die Stiftdatei wurde nicht gelesen, wenn die Farbsysteme als String definiert waren, berichtigt.

In neueren Versionen der ArtBat-Datei sind die Dateinamen für `Stifte`, `Bildschirmfarben` und `Stufen` enthalten. Diese werden dann daraus gelesen, wenn sie in den Parametern nicht explizit ausgewählt werden.

Es können jetzt mehrere Artendateien gelesen werden. Das Multiartenkonzept wurde komplett implementiert, auch die maßstabsabhängigen `Views`. Es können automatisch die `Katalogdaten` geladen werden, die in der GrafBat-Datei angegeben sind.

Die Farbangaben aus den `Katalogdaten` können jetzt auf die Features übertragen werden.

Die Größenangaben der `Textarten` können jetzt auf die Features übertragen werden, dazu stehen die Parameter `TEXT_SCALE` und `RESET_TEXT_POSITION` zur Verfügung.

26.01.2019 Der Reader wertet jetzt auch die Kommandos APP und KATALOG aus. Diese werden in aktuellen Versionen von Geograf selbst nicht geschrieben, bei Dateien aus Fremdsystemen können sie aber auftreten.

Geograf ermöglicht auch die Speicherung von mehreren externen Schlüssel je Element. Reader und Writer benutzen jetzt dazu ein Listenattribut.

09.02.2019 Die Elemente der `Katalogdaten` können jetzt auch als Feature gelesen werden.

Das aktuelle ARTBAT-Format wurde ergänzt, einige `art_`-Attribute wurde neu strukturiert.

Der FeatureType kann jetzt auch aus der Katalognummer erzeugt werden.

Geschlossene Polylinien sind mit der Geograf-Version 10 möglich, diese werden jetzt gelesen und geschrieben.

Beim Schreiben von Polylinien und Schraffuren entstanden z.T. doppelte Koordinaten, dies wurde korrigiert.

28.02.2019 Einige Sonderformen, die Geograf nicht selbst, sondern deren Schnittstellenprogramme erzeugen wurden implementiert. Ebenso wurden Elemente der historischen Versionen 1-2 nachgeführt.

Der Reader bricht bei Lesefehlern nicht mehr unbedingt ab. Auftretende Fehler werden dokumentiert, es erfolgt dann auch eine abschließende Fehlermeldung.

Löschungen und Umnummerierungen werden gelesen und geschrieben.

Die Geograf-Fortführung wird ausgewertet, das Attribut `fme_db_operation` erzeugt. Die Parameter `READ_OPERATION`, `READ_UNCHANGE`, `DELETE_LAYERS` und `RESOLVE_OPERATION` wurden dazu angelegt.

01.04.2019 Es wurde die Möglichkeit der Sortierung der Ausgabefeatures implementiert.

Diese Dokumentation wurde komplett überarbeitet.

