

# Benutzerhandbuch für *Survey2GIS* (Version 1.5.1)

– Software zur Konvertierung von Vermessungsdaten in GIS-Datenformate –

geschrieben von der *survey-tools.org*-Entwicklergruppe (<http://www.survey-tools.org>)

13. Oktober 2019

## Inhaltsverzeichnis

<b>1 Einführung</b>	<b>4</b>
1.1 Funktionen . . . . .	5
1.2 Struktur der Eingabedaten . . . . .	6
1.3 Installation . . . . .	6
1.4 Benutzung . . . . .	7
<b>2 Programmaufruf und -optionen</b>	<b>9</b>
2.1 Ein- und Ausgabedatei(en) . . . . .	10
2.2 Spracheinstellungen . . . . .	11
2.3 Liste der Optionen . . . . .	13
<b>3 Graphische Benutzeroberfläche (GUI)</b>	<b>22</b>
3.1 Einbindung des GUI in aufrufenden Anwendungen . . . . .	23
<b>4 Das Parserschema</b>	<b>25</b>
4.1 Syntax der Datei . . . . .	25
4.2 Einstellungen für [Parser] . . . . .	27
4.3 Einstellungen für [Field] . . . . .	31
4.4 Geometriemarker und Parsermodi . . . . .	36
4.4.1 „Min“: Minimale Kodierarbeit . . . . .	37
4.4.2 „Max“: Maximale Kontrolle . . . . .	38
4.4.3 „End“: Flexible Vermessung . . . . .	39
4.4.4 „None“: Vermessung von reinen Punktfeldern . . . . .	40

<b>5</b>	<b>Etiketten (Beschriftungen/Labels)</b>	<b>41</b>
<b>6</b>	<b>Datenauswahl</b>	<b>42</b>
6.1	Syntax der Auswahlkommandos . . . . .	42
6.2	Auswahltypen . . . . .	43
6.3	Auswahlmodifizierer und -ketten . . . . .	45
6.4	Selektionskommandos mit der graphischen Benutzeroberfläche . . . . .	47
<b>7</b>	<b>Orientierungsmodi</b>	<b>49</b>
7.1	Weltorientierung (Vorgabe) . . . . .	49
7.2	Lokale X-Z-Orientierung („Profile“) . . . . .	49
<b>8</b>	<b>Bezugssysteme und Reprojektionen</b>	<b>53</b>
8.1	Ein- und Ausgabesysteme setzen . . . . .	54
8.1.1	Interne KBS-Kürzel . . . . .	54
8.1.2	Angabe von KBS per EPSG-Nummer . . . . .	55
8.1.3	Angabe von KBS als PROJ.4-Definitionen . . . . .	55
8.1.4	Web Mercator als KBS . . . . .	56
8.2	Reprojektion mittels PROJ.4 . . . . .	57
8.2.1	Datumstransformationen . . . . .	58
8.2.2	Reprojektionen mit Grids . . . . .	60
8.3	Hilfestellungen zu KBS und Reprojektion . . . . .	61
<b>9</b>	<b>Topologische Qualität und Bereinigung</b>	<b>62</b>
9.1	Probleme und Bereinigungen . . . . .	62
9.1.1	Doppelte Stützpunkte (Ausdünnung) . . . . .	63
9.1.2	Splitter . . . . .	64
9.1.3	Mehrteilige Geometrien . . . . .	64
9.1.4	Selbstüberschneidungen bei Polygonen . . . . .	64
9.1.5	Gemeinsame Grenzen von Polygonen . . . . .	65
9.1.6	Innengrenzen von Polygonen (Löcher) . . . . .	68
9.1.7	Kreuzungspunkte auf Linien und Polygongrenzen . . . . .	68
9.1.8	Hängende Linienknoten (Über- und Unterschwinger) . . . . .	70
9.1.9	Polygonrichtung . . . . .	73
9.2	Praktische Hinweise und Einschränkungen . . . . .	73

<b>10 Ausgabeformate</b>	<b>75</b>
10.1 ESRI Shapefile	75
10.1.1 ESRI Shapefiles und Etiketten	76
10.1.2 „Null“-Werte in DBF Attributtabellen	76
10.2 Drawing Exchange Format (DXF)	76
10.2.1 DXF und Datenarchivierung	78
10.2.2 DXF und topologische Daten	78
10.2.3 DXF und planare Polygone bzw. Polylinien	78
10.2.4 DXF und Attributdaten	79
10.2.5 „Null“-Werte in DXF-Attributen	80
10.2.6 DXF und Etiketten	80
10.3 GeoJSON	80
10.3.1 Geometrietypen in GeoJSON	81
10.3.2 Primärschlüssel in GeoJSON (Feld “id”)	82
10.3.3 „Null“-Werte in GeoJSON-Objekten	82
10.3.4 Etiketten in GeoJSON	82
10.4 Keyhole Markup Language (Google KML)	82
10.4.1 Geometrietypen in KML	84
10.4.2 Label layers in KML	84
<b>11 Umgang mit Sonderzeichen</b>	<b>85</b>
<b>12 Hinweise zur Vermessungspraxis</b>	<b>87</b>
<b>A Bekannte Probleme</b>	<b>89</b>
<b>B Lizenz</b>	<b>90</b>

# 1 Einführung

*Survey2GIS* dient der Verarbeitung von Daten aus einer *topographischen Vermessung*. Zweck einer solchen ist es, geometrische Aspekte der realen Welt genau so aufzunehmen, wie sie auf einer *Oberfläche sichtbar* sind. Dementsprechend geht auch dieses Programm davon aus, dass Geometrien (Punkte, Linien und Polygone) prozessiert werden, welche die *sichtbaren* Schnittflächen von Objekten mit einer Ebene (einer natürlichen oder künstlich freigelegten Oberfläche) repräsentieren. Dieses Programm eignet sich nicht als Werkzeug zur Konstruktion oder zum technischen Zeichnen von vermuteten oder interpretierten Geometrien. In der technischen Umsetzung harmonisiert dies mit den (überwiegend 2D-) GIS-Konzepten „Layer“ und „topologische Qualität“. Falls die von *Survey2GIS* erzeugten Geometrien nicht den erwarteten entsprechen, sollte unbedingt Abschnitt 9 dieses Handbuchs sorgfältig studiert werden, bevor in den Daten, den Programmeinstellungen oder dem Programm selbst nach Fehlern gesucht wird.

Die Software *Survey2GIS* ist ein flexibles Werkzeug zur Verflechtung, topologischen Bereinigung und Konvertierung von Vermessungsdaten, die in Textdateien vorliegen. Hierbei kann es sich z. B. um die Resultate von tachymetrischen oder GPS-Vermessungen handeln. Die Arbeitsweise von digitalen Totalstationen oder GPS-Empfängern besteht darin, Einzelpunktmessungen durchzuführen und im internen Speicher des Geräts abzulegen. Die vom Hersteller des jeweiligen Geräts beigelegte Treibersoftware liest den internen Speicher nach Abschluss der Feldarbeit aus und produziert einfache Textdateien, welche *eine Messung pro Zeile* enthalten. *Nur in dieser Form* sind die Daten zur Verarbeitung mittels *Survey2GIS* geeignet.

Die Benutzung von *Survey2GIS* erfordert im Anschluss zwei wesentliche Schritte:

1. Die Benutzerin erstellt eine einfache Textdatei mit einer Parserdefinition, welche die Struktur der Vermessungsdaten beschreibt.
2. Der Benutzer startet *Survey2GIS* unter Angabe der Parserdatei und einer oder mehreren Eingabedateien mit Vermessungsdaten.

Die Syntax der Parserdatei wird in Abschnitt 4 detailliert beschrieben.

Um komplexere Formen („Geometrien“), die im Feld als Linenzüge oder Polygone eingemessen wurden, aus den Messdaten zu rekonstruieren, müssen die Einzelmessungen, welche die Stützpunkte der Geometrien beschreiben, korrekt zusammengeführt werden. Gleichzeitig soll jeder Geometrie ein Datensatz mit beschreibenden Sachdaten (sog. „Attributdaten“), etwa einer laufenden Indexnummer oder einem Objektcode, zugewiesen werden.

Damit dies alles einwandfrei funktioniert, müssen Vermesser im Feld Zusatzdaten in die Messungen einfügen, welche zusammengehörige Messungen beschreiben. Die Vermessungshardware, die Feldarbeit und die Datenverarbeitung müssen entsprechend aufeinander abgestimmt werden. Die Aufgabe von *Survey2GIS* ist es dann, komplexe Geometrien aus den Rohdaten der Vermessung zu rekonstruieren und somit als Bindeglied zwischen Vermessung und GIS-gestützter Datenverarbeitung zu fungieren.

Bei der Entwicklung der Software standen Flexibilität, Robustheit und die Qualität der Ausgabedaten im Vordergrund. Anwendern wird keine bestimmte Methodik bei der Vermessung im Feld aufgezwungen, son-

dern sie können zwischen einer Reihe von grundsätzlichen Arbeitsweisen (sog. „Parsermodus“) wählen und die Verarbeitung der resultierenden Daten bis ins Detail an die eigenen Bedürfnisse anpassen.

Das GIS-Vektormodell kann zwar 3D-Koordinaten speichern, so dass 3D-Ansichten dargestellt werden können, es ist aber kein „echtes“ 3D-Modell. GIS verwenden eine 2D-Ansicht aus der Vogelperspektive um Vektorebenen darzustellen und um die topologischen Beziehungen zwischen Objekten („überlappt“, „enthält“, „grenzt an“, etc.) zu repräsentieren. Aus diesem Grund sind GIS *nicht* dazu geeignet, Objekte mit starker vertikaler Strukturierung, wie etwa Gebäude, akkurat darzustellen. Der intendierte Einsatzzweck von *Survey2GIS* ist die *topographische Vermessung* von Punkten, linearen Strukturen und Flächen auf der Geländeoberfläche. In *Survey2GIS* existieren begrenzte Möglichkeiten für die Aufbereitung von „vertikalen“ Daten für die Nutzung im 2D-GIS (s. 7).

## 1.1 Funktionen

Die folgenden Funktionen werden von *Survey2GIS* in der aktuellen Version zur Verfügung gestellt:

- Zeilenweises Einlesen von Koordinatenmessungen und kodierten Attributdaten aus beliebig vielen Eingabedateien.
- Zuordnung der Messungen zu geometrischen Objekten in Form von Punkten, Linien und Polygonen.
- Frei definierbare Datenfelder zur Speicherung von Attributdaten.
- Detaillierte Validierung und Protokollierung von Einzelmessungen und komplexen Geometrien, inkl. topologischer Bereinigung.
- Zusammenführung aller Eingabedaten und Export der Geometrie- und Attributdaten in standardisierte GIS-Datenformate.
- Neuorientierung und Reprojektion von Koordinatendaten.

Die besonderen Merkmale des Programms sind:

- Quelloffen und frei erhältlich. Keine proprietäre Technologie, keine lästigen Kopierschutzmechanismen.
- Lauffähig unter Linux, Mac OS X (macOS) und Windows (32 und 64 Bit).
- Minimale Hardwareanforderungen.
- Verwendung von der Kommandozeile oder mittels einer graphischen Benutzeroberfläche (GUI).
- Verarbeitung von 2D- und 3D-Eingabedaten.
- Einfache Übersetzbarkeit aller Programmtexte in beliebige Sprachen und Berücksichtigung unterschiedlicher Notationen von Dezimalzahlen (das Programm kann jederzeit auf Englisch „zurückgeschaltet“ werden).

- Flexible Anpassbarkeit an unterschiedlich strukturierte Eingabedateien über frei definierbares Parser-schema.
- Abarbeitung mehrerer Eingabedateien (mit derselben Struktur) und Zusammenführung der darin enthaltenen Daten in die Ausgabedatei(en).
- Vollständige Protokollierung aller vom Programm ausgegebenen Hinweise, Warnungen und Fehlermeldungen auf dem Bildschirm oder in eine Datei zum gezielten Suchen und Beseitigen von Fehlern in den Eingabedaten.
- Robuste Arbeitsweise: fehlerhafte Eingabedaten werden, sofern möglich, übersprungen ohne die Verarbeitung abubrechen.
- Topologische Datenprüfung und -bereinigung zur Erzeugung qualitativ hochwertiger GIS-Daten für räumliche Analysen.
- Flexibler Umgang mit unvollständigen Eingabedaten.

## 1.2 Struktur der Eingabedaten

Die Eingabedaten erwartet *Survey2GIS* in einer oder mehreren Textdateien (alternativ aus dem Standardeingabestrom des Betriebssystems: s. 2.1). Die Eingabedaten können aus eingemessenen Punkten (bzw. Koordinaten) und beschreibenden Sachdaten bestehen. Sie können auch beliebige weitere, z. B. von der Exportsoftware des Vermessungsgerätes hinzugefügte, Daten enthalten, die nach Bedarf übernommen oder verworfen werden können. In jedem Fall muss aber eine Punktmessung pro Zeile vorliegen, d. h. jede Zeile der Eingabedatei muss mindestens genau einen X- und einen Y-Koordinatenwert enthalten.

Von *Survey2GIS* werden keine veralteten Kodierungssysteme für Umlaute und Fremdzeichen (z. B. DOS/Windows-Codepages) unterstützt. Es wird davon ausgegangen, dass alle Eingabedaten entweder Standard-ASCII (keine erweiterten ASCII-Tabellen) oder UTF-8-Kodierung verwenden. Näheres s. 11.

## 1.3 Installation

Die Software benötigt keine aufwendige Installation, sondern kann einfach aus dem „bin“-Verzeichnis (für das jeweilige Betriebssystem) heraus gestartet werden. Weitere Hinweise finden sich in den „README“-Dateien für die einzelnen Betriebssysteme.

Eine nützliche Ergänzung zu *Survey2GIS* ist die, ebenfalls quelloffene und frei erhältliche, Software *TotalOpenStation* (<http://tops.iosa.it>). Diese dient dazu, die Messungen im internen Speicher von Totalstationen oder GPS-Empfängern direkt auszulesen und in einfache Textdateien zu konvertieren.

Windows-Benutzer sollten außerdem einen Texteditor, wie z. B. *NotePad++* (<http://notepad-plus-plus.org>) installieren, um umfangreichere Parser- und Messdateien bearbeiten zu können.

## 1.4 Benutzung

Die Software kann sowohl von der Kommandozeile als auch per graphischer Benutzeroberfläche (GUI) gesteuert werden. Letzterer Fall wird in Abschnitt 3 auf Seite 22 näher beschrieben.

Um die Software von der Kommandozeile aus zu benutzen, muss unter Windows zunächst *cmd.exe* gestartet werden und anschließend ins entsprechende Verzeichnis auf dem Installationslaufwerk gewechselt werden (die Angaben in spitzen Klammern „<>“ müssen durch die tatsächlichen Pfadnamen ersetzt werden):

```
cd /D <Laufwerk>:\<Installationsverzeichnis>
```

Für einen ersten Funktionstest kann z. B. ein neues Verzeichnis innerhalb des Programmverzeichnisses angelegt werden, in welches die Ausgabedateien geschrieben werden. Im Folgenden wird davon ausgegangen, dass dieses „output“ heißt. Außerdem müssen die mitgelieferten Dateien „parser\_desc.end.txt“ und „sample\_data.end.dat“ aus dem Unterverzeichnis „samples“ in dasselbe Verzeichnis kopiert werden, in welchem sich die Binärdateien (ausführbares Programm) von *Survey2GIS* befinden.

Sind obige Voraussetzungen erfüllt, so lässt sich ein einfacher erster Funktionstest durchführen:

```
survey2gis -p parser_desc.end.txt -o output -n test_daten sample_data.end.dat
```

Zur Erläuterung:

- Die Option *-p* (*parser*) gibt an, in welcher Datei die Strukturinformationen zu den Eingabedateien zu finden sind. Details zum Aufbau dieser Datei finden sich in 4.2 auf Seite 27.
- Mittels *-o* (*output*) wird das Verzeichnis angegeben, in welches die Ausgabedateien geschrieben werden sollen. In diesem Beispiel heißt es ebenfalls „output“. Per Voreinstellung produziert *Survey2GIS* Shapefiles.
- Die Option *-n* (*base name*) gibt den Basisnamen der Ausgabedateien an. Da z. B. Shapefiles aus mehreren einzelnen Dateien bestehen, werden die Endungen, „.shp“, „.shx“ und „.dbf“ an diesen Basisnamen gehängt (s. 10.1 auf Seite 75).
- Die Datei „sample\_data.end.dat“ enthält ein paar einfache Vermessungsdaten, welche Punkte, Linien und Polygone beschreiben. Es handelt sich um eine Textdatei, die mit einem beliebigen Texteditor betrachtet und bearbeitet werden kann. Im Prinzip könnten noch weitere solcher Eingabedateien an die Kommandozeile angehängt werden. Ihre Inhalte würden dann alle zusammen in die Ausgabedateien fließen.

Das Resultat sollten drei Shapefiles sein, welche die Punkte, Linie und Polygone aus „sample\_data.end.dat“ enthalten.

Eine Auflistung aller Funktionen steht über die Option *-h* (*help*) zur Verfügung:

```
survey2gis -h
```

Eine detaillierte Beschreibung folgt im Abschnitt [2 auf der nächsten Seite](#).



## 2 Programmaufruf und -optionen

Die Programmausführung geschieht über die Kommandozeile (in Windows per *cmd.exe* zu starten), wie hier beschrieben, oder per graphischer Benutzeroberfläche (GUI), wie in Abschnitt 3 auf Seite 22 beschrieben.

Zur Benutzung von der Kommandozeilen werden nach dem Programmnamen (*survey2gis*) eine oder mehrere Optionsnamen mit nachfolgenden, vom Benutzer zu spezifizierenden, Optionswerten und evtl. den Namen der Eingabedatei(en) angegeben werden. Die einzelnen Optionen sind dabei vom Programmnamen und voneinander durch Leerzeichen zu trennen.

Der Programmaufruf mittels (Leerzeichen zwischen Programmnamen und Option *-h* beachten!)

```
survey2gis -h
```

zeigt eine zusammenfassende Beschreibung der Software und ihrer Optionen auf dem Bildschirm an (hier stark verkürzt wiedergegeben):

```
Benutzung: survey2gis -p DATEI -o VERZEICHNIS -n NAME [OPTION]... [DATEI]...
Beschreibungen der Geometrien u. Attribute aus einer Messprotokolldatei
im ASCII-Format auslesen und diese in gängige GIS- und CAD-Ausgabeformate
umwandeln.
```

Mögliche OPTIONen sind:

```
-p, --parser=      Name der Datei mit dem Parser-Schema (erforderlich)
-o, --output=      Verzeichnisname für Ausgabedatei(en) (erforderlich)
-n, --name=        Basisname für Ausgabedatei(en) (erforderlich)
-f, --format=      Ausgabeformat (siehe Liste unten; Vorgabe: "shp")
                   "shp" (Esri Shapefile)
                   "dxf" (AutoCAD Drawing Exchange Format)
                   "geojson" (GeoJSON Object)
                   "kml" (Keyhole Markup Language)
-L, --label=       Textfeld für Etiketten auswählen (s. Handbuch)
--label-mode-point= Etikettierungsmodus f. Punkte ("center")
--label-mode-line=  Etikettierungsmodus f. Linien ("center")
--label-mode-poly=  Etikettierungsmodus f. Polygone ("center")
                   "center" ((Im Zentrum der Geometrie platzieren))
                   "first" (Auf erstem Stützpunkt platzieren)
                   "last" (Auf letztem Stützpunkt platzieren)
                   "none" (Nicht etikettieren)
...
```

Ein minimaler Programmaufruf von *Survey2GIS* zum Zweck der Datenprozessierung umfasst demnach:

- Die Angabe einer Datei mit Parserschema (s. 4 auf Seite 25) mittels der Option `-p`,
- den Pfadnamen eines *existierenden* Verzeichnisses, in welchem die Ausgabedatei/en gespeichert werden sollen (Option `-o`),
- einen Basisnamen für die Benennung der Ausgabedatei(en; Option `-n`),
- und eine oder mehrere Eingabedateien.

Das Parserschema ist eine strukturierte Textdatei, deren Aufbau in Abschnitt 4 auf Seite 25 detailliert beschrieben ist. Ihre Aufgabe ist es, den Inhalt der Eingabedatei(en) und die Form der Ausgabedaten detailliert zu beschreiben.

Die meisten Optionsnamen besitzen eine Kurz- und eine Langschreibung, welche äquivalent sind. Es spielt also z. B. keine Rolle, ob die Parserschemadatei „parser.txt“ so:

```
-p parser.txt
```

oder so:

```
--parser=parser.txt
```

angegeben wird. In der (besser lesbaren) Langform werden Optionswerte mit Gleichheitszeichen und ohne Leerzeichen nach dem Optionsnamen angegeben.

Die Reihenfolge, in welcher Optionen angegeben werden, spielt dabei keine Rolle.

Optionswerte und Dateinamen, welche Leerzeichen beinhalten, müssen in Anführungszeichen gesetzt werden, da sie dem Programm sonst irrtümlich wie mehrere Optionen erscheinen.

Die graphische Benutzeroberfläche (s. 3 auf Seite 22) wird aufgerufen, indem *Survey2GIS* ohne jegliche Optionen gestartet wird, oder indem die spezielle Option zum Anzeigen des GUIs gesetzt wird:

```
survey2gis --show-gui
```

## 2.1 Ein- und Ausgabedatei(en)

Alle auf die Programmoptionen folgenden Werte werden als Namen von Eingabedateien interpretiert. Sämtliche Eingabedateien werden unter Zuhilfenahme desselben Parserschemas (Option „-p“) abgearbeitet und als Ausgabedateien im durch die Option „-o“ angegebenen Verzeichnis gespeichert. Das Ausgabeverzeichnis muss existieren und den Schreibzugriff erlauben. Das aktuelle Verzeichnis kann als Ausgabeverzeichnis gewählt werden, indem als Name der Punkt („.“) angegeben wird.

Das Format der Ausgabedaten wird mittels Option „-f“ gewählt. Per Voreinstellung ist der Optionswert „shp“, d.h. es werden ESRI Shapefiles produziert (s. 10.1 auf Seite 75). Alle Eingabedaten werden von *Survey2GIS* zu einem gemeinsamen Ausgabedatensatz verwoben. Wieviele Ausgabedateien im Endeffekt tatsächlich produziert werden, hängt allerdings von den Eigenschaften des gewählten Ausgabeformats ab (s. Erläuterungen zu den einzelnen Formaten unter 10 auf Seite 75). Aus diesem Grunde sollte die Option „-n“ nur den Basisnamen der Ausgabedatei(en) ohne Dateierweiterung (z.B. „.shp“) enthalten. Letztere werden nach Bedarf automatisch hinzugefügt.

Wichtig: Dies bedeutet auch, dass bereits existierende Dateien automatisch und ohne Rückfrage überschrieben werden, falls ihre Namen mit den von *Survey2GIS* produzierten übereinstimmen!

Eine besondere Bedeutung hat das Minuszeichen („-“). Wird es in der Liste der Eingabedateien angegeben (dies kann auch mehrfach geschehen), so erwartet *Survey2GIS* Eingabedaten aus dem Standardeingabekanal der Systemkonsole. Unter Mac OS X (macOS) und Linux ist dieser Kanal als *stdin* bekannt. Wird der Eingabekanal nicht vom Benutzer aus einer anderen Quelle zu *Survey2GIS* umgelenkt, so bedeutet dies, dass der Benutzer Daten direkt auf der Kommandozeile eingeben kann. Die Eingabezeilen sind dabei genau in dem Format einzutippen, wie es das verwendete Parserschema verlangt. Um die Eingabe abzuschließen, muss der Benutzer unter Windows ein „^Z“ (Tasten „STRG“ und „Z“) und unter Linux und Mac OS X ein „^C“ (Tasten „STRG“ und „C“) auf einer ansonsten vollkommen leeren Zeile eingeben.

## 2.2 Spracheinstellungen

Bei *Survey2GIS* handelt es sich um eine internationalisierbare Software, deren Programmmenüs und -meldungen sich vollständig in unterschiedliche Sprachen übersetzen lassen. Bei Programmstart prüft *Survey2GIS* die aktuellen Spracheinstellungen des Betriebssystems und zeigt entsprechend übersetzte Bildschirmtexte an. Existiert keine passende Übersetzung, so wird auf Englisch zurückgeschaltet. Alternativ kann *Survey2GIS* mittels der Option „-e“ dazu gezwungen werden, stets englische Sprache zu verwenden.

Zur Internationalisierung gehören auch unterschiedliche Notationen für Dezimalzahlen. So wird die Zahl „Zehntausendkommanull“ im deutschen und niederländischen Sprachraum folgendermaßen notiert:

10.000,00

Im englischen Sprachraum jedoch so:

10,000.00

Da die meisten Geräte bzw. deren Software ihre Daten in Englischer Notation ausgeben, ist letztere die Voreinstellung in *Survey2GIS*. Bei Bedarf kann aber per Optionen „-i“ (Dezimalpunkt) und „-g“ (Tausendertrenner) explizit ein anderes Zahlenformat gesetzt werden.

Es ist sehr wichtig, dass die Spracheinstellungen für die Zahlenformatierung korrekt eingestellt sind, so dass sie den *in den Eingabedaten verwendeten* Konventionen entsprechen. Typische Symptome von falschen Einstellungen sind deplazierte Objekte in den Ausgabedaten, falsche Koordinatenbereiche und/oder stufige Linien und deformierte Polygone.

## 2.3 Liste der Optionen

Die folgende Liste enthält alle von *Survey2GIS* akzeptierten Optionen sowie deren Beschreibungen. Die Optionen stehen hier in derselben Reihenfolge, in der sie auch beim Aufruf von „survey2gis -h“ (Hilfefunktion) auf der Kommandozeile erscheinen.

---

Option: **-p**, **--parser=**

Bedeutung: Vollständiger Pfad und Name der Textdatei, welche das Parserschema zur Verarbeitung der Eingabedatei(en) enthält (s. Abschnitt **4 auf Seite 25**).

Beispiel:

```
-p C:\Users\Me\parser.txt
```

---

Option: **-o**, **--output=**

Bedeutung: Vollständiger Pfad und Name des Verzeichnisses, in welchem die Ausgabedateien gespeichert werden.

Beispiel:

```
-o c:\Data
```

---

Option: **-n**, **--name=**

Bedeutung: Basisname (d.h. ohne Erweiterung) für Ausgabedatei(en). Speicherung erfolgt im mit „-o“ (s. o.) angegebenen Verzeichnis.

Beispiel:

```
-n output
```

---

Option: **-f**, **--format=**

Bedeutung: Name des Formats für die Speicherung der Ausgabedaten (Vorgabewert: „shp“).

Hinweis: Der mit der Option „-h“ angezeigte Hilfetext enthält die vollständige Liste aller möglichen Formate (s. auch **10 auf Seite 75**).

Beispiel:

```
-f shp
```

---

Option: **-L**, --label=

Bedeutung: Der Name eines *Feldes* (see 4.3) das für die Erzeugung von Etiketten („Labels“) verwendet werden soll (Details s. 5).

Beispiel:

```
-L level
```

---

Option: --label-mode-point=

Bedeutung: Gibt an, wie die Etiketten auf Punktgeometrien platziert werden sollen. Die Möglichkeiten sind: „center“ (Voreinstellung), „first“, „last“ und „none“.

Hinweis: Nur „none“ hat tatsächlich einen Effekt auf Punktgeometrien (zur Erläuterung s. 5).

Beispiel:

```
--label-mode-point=none
```

---

Option: --label-mode-line=

Bedeutung: Gibt an, wie die Etiketten auf Liniengeometrien zu platzieren sind. Die Möglichkeiten sind: „center“ (Voreinstellung), „first“, „last“ und „none“ (zur Erläuterung s. 5)..

Beispiel:

```
--label-mode-line=center
```

---

Option: --label-mode-poly=

Bedeutung: Gibt an, wie die Etiketten auf Polygoneometrien zu platzieren sind. Die Möglichkeiten sind: „center“ (Voreinstellung), „first“, „last“ und „none“ (zur Erläuterung s. 5)..

Beispiel:

```
--label-mode-line=center
```

---

Option: **-O**, --orientation=

Bedeutung: Bestimmt den *Orientierungsmodus* der Axen des (evtl. synthetischen) Koordinatensystems für die Ausgabedaten (s. 7). Der Vorgabemodus ist „world“, d. h. es findet *keine* Veränderung der Axenorientierung gegenüber den originalen Eingabedaten statt.

Beispiel:

`-L localxz`

---

Option: **-S**, `--selection=`

Bedeutung: Gibt ein *Auswahlkommando* an, mittels dessen sich die zu verarbeitenden Eingabedaten auf eine Untermenge (Selektion), die bestimmte Kriterien erfüllt, beschränken lässt. Es ist möglich, mehr als eine Auswahl zu treffen, wobei jeweils eine separate „--selection“-Option anzugeben ist. Die Bearbeitung der Auswahlkommandos erfolgt in der Reihenfolge, in welcher sie auf der Kommandozeile angegeben werden. Weitere Details finden sich im Abschnitt [6](#).

Hinweis: Die Abkürzung für diese Option ist ein *großes* „-S“, während die Abkürzung für die Option „--snapping=“ (s. u.) ein *kleines* „-s“ ist.

Beispiel:

`-S objdesc:all:EQ:Boundary`

---

Option: **-l**, `--log=`

Bedeutung: Bewirkt, dass sämtliche Programm Meldungen in der angegebenen Protokolldatei abgespeichert werden.

Beispiel:

`-l C:\Users\Me\protocol.txt`

---

Option: **-t**, `--tolerance=`

Bedeutung: Setzt den Schwellenwert für die Distanz von unterscheidbaren Messungen. Messungen, deren räumlicher Abstand voneinander den hier gesetzten nicht überschreiten, werden als identische Messungen (z. B. doppelte Stützpunkte) betrachtet (s. [9.1.1 auf Seite 63](#)).

Beispiel:

`-t 0,05`

---

Option: **-s**, `--snapping=`

Bedeutung: Setzt den Distanz-Schwellenwert für das „Snapping“ (Einrasten) auf die Grenzen von Polygonen. Stützpunkte rasten auf den nächsten Stützpunkt einer Polygongrenze ein, der weniger weit entfernt liegt als der hier angegebene Wert (s. [9.1.5 auf Seite 65](#)).

Beispiel:

-s 0,05

---

Option: **-D**, --dangling=

Bedeutung: Setzt den Schwellenwert für das „Snapping“ (Einrasten) der ersten und letzten Stützpunkte von Liniengeometrien. Eingerastete Linienstützpunkte werden auf das nächste Segment bewegt, welches Teil einer Linien oder einer Polygongrenze ist. Man bezeichnet dies auch als Korrektur von Über- (Stützpunkt liegt knapp hinter dem Segment, auf dem er eigentlich liegen sollte) bzw. Unterschwingern (Stützpunkt liegt knapp vor dem Liniensegment, auf dem er eigentlich liegen sollte); auch engl. „Overshoots“ bzw. „Undershoots“ (Details s. 9.1.8). Wird diese Option auf „0“ gesetzt (Vorgabewert), dann wird keine derartige Korrektur durchgeführt.

Hinweis: Hinweis: Die Abkürzung für diese Option ist ein *großes* „-D“, während die Abkürzung für die Option „--decimal-places=“ (s. u.) ein kleines „-d“ ist.

Beispiel:

-D 0,05

---

Option: **-x**, --x-offset=

Bedeutung: Konstanter Aufschlag auf alle eingelesenen X-Koordinaten.

Hinweis: Nützlich wenn z. B. UTM-Koordinaten ohne die ersten beiden Stellen eingemessen worden sind. Zu beachten ist, dass die Attributtabelle der Ausgabedaten immer die *originalen Messkoordinaten*, ohne Offset, enthält! Falls die transformierten Koordinaten ebenfalls in der Attributtabelle abgelegt werden sollen, so sollten die entsprechenden Werkzeuge des GIS („Feldrechnern“, o. ä.) direkt nach dem Datenimport genutzt werden, um den Offset auf die Feldwerte aufzuschlagen.

Beispiel:

-x 100000

---

Option: **-y**, --y-offset=

Bedeutung: s.o. (-x, --x-offset=).

Beispiel:

-y 100000

---

Option: **-z**, --z-offset=

Bedeutung: s.o. (-x, --x-offset=).

Beispiel:



-z 100000

---

Option: **-d**, --decimal-places=

Bedeutung: Gibt an, mit wievielen Nachkommastellen (Genauigkeit) numerische Werte in den entsprechenden Attributfeldern des Ausgabedatensatzes gespeichert werden sollen. Für Vermessungsdaten in Metern bewirkt z. B. eine Einstellung von „2“ eine Zentimetergenauigkeit. Die Genauigkeit der Koordinatenspeicherung ist hiervon nicht betroffen (Vorgabe: „3“).

Beispiel:

-d 3

---

Option: **-i**, --decimal-point=

Bedeutung: Gibt an, welches Zeichen zur Darstellung des Dezimalpunkts in den Eingabedaten verwendet wird (s. [2.2 auf Seite 11](#)).

Beispiel:

-i .

---

Option: **-g**, --decimal-group=

Bedeutung: Gibt an, welches Zeichen zur Gruppierung von Dezimalzahlen (z. B. Tausendertrennzeichen) in den Eingabedaten verwendet wird (s. [2.2 auf Seite 11](#)).

Beispiel:

-g ,

---

Option: **-r**, --raw-data

Bedeutung: Erzeugt eine zusätzliche Ausgabedatei mit Punkten, welche sämtliche individuellen Messungen der Eingabedaten, inkl. der Stützpunkte aller Linien und Polygone als Einzelpunkte, enthält. Auf diese Weise ist es möglich, die gesamte Information der Vermessungsdaten, insbes. individuelle Z-Koordinaten und Attributwerte von Stützpunkten, zu konservieren. Wenn das Ausgabeformat es bedingt, unterschiedliche Geometrien in einzelnen Dateien zu speichern (z. B. Esri Shapefile), so wird die zusätzliche Ausgabedatei durch ein angehängtes „\_raw“ am Dateinamen gekennzeichnet.

Hinweis: Die Ausgabe *enthält nicht* diejenigen (Stütz-)Punkte, die während der topologischen Bereinigung (s. [9](#)) gelöscht wurden, da sie zu dicht an benachbarten Punkten standen. Um derlei Punkte ebenfalls zu

erhalten, muss die Option „--tolerance=” (s. o.) auf einen negativen Wert gesetzt werden. Im Parsermodus „None” (s. 4.4) wird die Ausgabe der „rohen” Punktdaten der regulären Ausgabe entsprechen.

Beispiel:

```
-r
```

---

Option: -2, --force-2d

Bedeutung: Erzwingt zweidimensionale Ausgabedaten, unabhängig davon, ob die Eingabedaten dreidimensional sind. Diese Einstellung beeinflusst nur die Speicherung der Koordinatenwerte in der Ausgabedatei. Interne Berechnungen, wie evtl. topologische Bereinigungen, sind hiervon nicht betroffen. Für eine vollständig zweidimensionale Verarbeitung von Daten sollte stattdessen das Parserschema so angepasst werden, dass nur zwei Koordinatenfelder definiert werden.

Beispiel:

```
-s
```

---

Option: -c, --strict

Bedeutung: Schaltet den Parser in einen „strengen” Modus welcher ihn dazu veranlasst, Eingabedaten ohne Geometriemarker zu verwerfen, anstatt sie als Punktmessungen zu interpretieren (s. 4.4 auf Seite 36).

Beispiel:

```
-c
```

---

Option: -v, --validate

Bedeutung: Führt eine Validierung des mit der Option „-p” angegeben Parserschemas durch. Der Inhalt der Schemadatei wird auf syntaktische Korrektheit und Vollständigkeit geprüft und eventuelle Fehler werden gemeldet.

Beispiel:

```
-p parser.txt -v
```

---

Option: -e, --english

Bedeutung: Stellt alle Programmmeldungen und Dezimalnotation der Eingabedaten auf Englisch um.

Beispiel:

-e

---

Option: **-h**, --help

Bedeutung: Gibt Hinweise zur Programmbenutzung, inkl. kurzer Beschreibung aller Optionen, auf dem Bildschirm aus.

Beispiel:

-h

---

Option: --proj-in=

Bedeutung: Bestimmt das *vorgegebene* Koordinatenbezugssystem (Quell-KBS) der *Eingabedaten* (per internem Kürzel, EPSG-Eintrag oder vollständiger PROJ.4-Definition). Bei der Angabe einer PROJ.4-Definition ist auf korrekte Groß- und Kleinschreibung zu achten. Details hierzu s. Abschnitt 8.

Beispiel:

--proj-in=epsg:31467

---

Option: --proj-out=

Bedeutung: Bestimmt das *beabsichtigte* Koordinatenbezugssystem (Ziel-KBS) der *Ausgabedaten* (per internem Kürzel, EPSG-Eintrag oder vollständiger PROJ.4-Definition). Bei der Angabe einer PROJ.4-Definition ist auf korrekte Groß- und Kleinschreibung zu achten. Details hierzu s. Abschnitt 8.

Beispiel:

--proj-out=epsg:wgs84

---

Option: --proj-dx=

Bedeutung: Gibt die Verschiebung entlang der geographischen X-Achse für eine Transformation des geodätischen Datum des *Quell-KBS* nach WGS 84 an. Details s. Abschnitt 8.2.1. Die Voreinstellung ist "0", also "keine X-Verschiebung".

Beispiel:

--proj-dx=598,1

---

Option: --proj-dy=

Bedeutung: Gibt die Verschiebung entlang der geographischen Y-Achse für eine Transformation des geodätischen Datum des *Quell-KBS nach* WGS 84 an. Details s. Abschnitt 8.2.1. Die Voreinstellung ist "0", also "keine Y-Verschiebung".

Beispiel:

--proj-dy=73,7

---

Option: --proj-dz=

Bedeutung: Gibt die Verschiebung entlang der geographischen Z-Achse für eine Transformation des geodätischen Datum des *Quell-KBS nach* WGS 84 an. Details s. Abschnitt 8.2.1. Die Voreinstellung ist "0", also "keine Z-Verschiebung".

Beispiel:

--proj-dz=418,2

---

Option: --proj-rx=

Bedeutung: Gibt die Rotation um die geographische X-Achse für eine Transformation des geodätischen Datum des *Quell-KBS nach* WGS 84 an. Details s. Abschnitt 8.2.1. Die Voreinstellung ist "0", also "keine X-Rotation".

Beispiel:

--proj-rx=0,202

---

Option: --proj-ry=

Bedeutung: Gibt die Rotation um die geographische Y-Achse für eine Transformation des geodätischen Datum des *Quell-KBS nach* WGS 84 an. Details s. Abschnitt 8.2.1. Die Voreinstellung ist "0", also "keine Y-Rotation".

Beispiel:

--proj-ry=0,045

---

Option: --proj-rz=

Bedeutung: Gibt die Rotation um die geographische Z-Achse für eine Transformation des geodätischen Datum des *Quell-KBS nach* WGS 84 an. Details s. Abschnitt 8.2.1. Die Voreinstellung ist "0", also "keine Z-Rotation".

Beispiel:

```
--proj-rz=-2
```

---

Option: --proj-ds=

Bedeutung: Gibt den Skalierungsfaktor für eine Transformation des geodätischen Datum des *Quell-KBS nach* WGS 84 an. Details s. Abschnitt 8.2.1. Der Vorgabewert ist "1", also "keine Skalierungsänderung".

Beispiel:

```
--proj-ds=6,7
```

---

Option: --proj-grid=

Bedeutung: Gibt den vollständigen Pfad und Namen einer Grid-Datei für eine hochgenaue lokale Transformation des geodätischen Datums des *Quell-KBS nach* WGS 84 an. Details s. 8.2.2. Per Voreinstellung wird keine Grid-Datei verwendet.

Beispiel:

```
--proj-grid=BETA2007.gsb
```

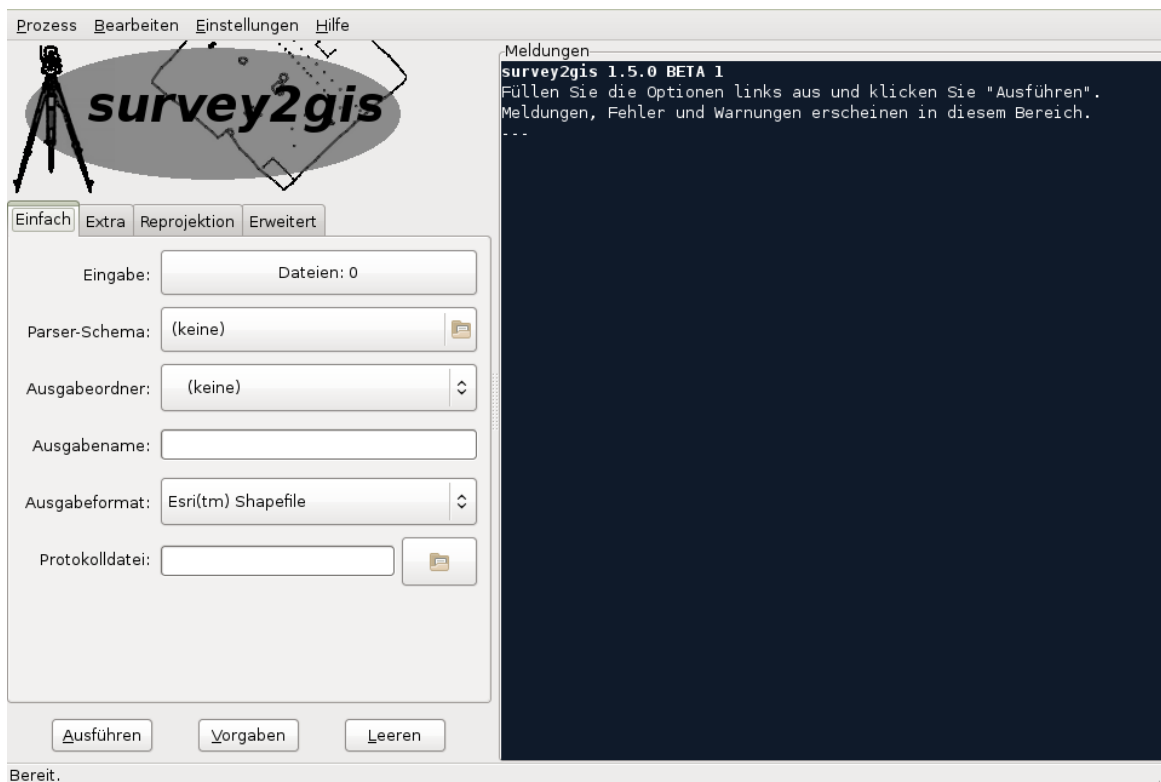
### 3 Graphische Benutzeroberfläche (GUI)

Alternativ zum Aufruf über die Kommandozeile lässt sich *Survey2GIS* auch mittels einer graphischen Benutzeroberfläche (Graphical User Interface, GUI) steuern. Um das GUI zu starten, existieren zwei Möglichkeiten:

1. Aufruf von „survey2gis“ ohne Angabe irgendwelcher Optionen: das GUI startet mit den Voreinstellungen.
2. Aufruf von „survey2gis“
3. mit der zusätzlichen Option „--show-gui“ : das GUI übernimmt alle auf der Kommandozeile übergebenen Optionswerte. Auf diese Weise lassen sich auch Sammlungen von Voreinstellungen, z. B. als DOS-Batchdateien oder Unix-Shellskripte, realisieren.

Das GUI gliedert sich vertikal in zwei wesentliche Bereiche (s. u.). Auf der linken Seite finden sich die Optionen, welche denen der Kommandozeile (s. 2 auf Seite 9) entsprechen. Diese sind aber in mehrere Kategorien/Seiten unterteilt, welche über Karteireiter gewählt werden können. Auf der Seite („Einfach“) finden sich die wesentlichen Optionen, die benötigt werden, um überhaupt Ausgabedateien erzeugen zu können; unter „Extra“ finden sich weitere Möglichkeiten, u. a. zur Etikettierung und Auswahl von Daten (s. 5 und 6); die Seite „Reprojektion“ steuert geographische Koordinatentransformationen (s. 8); auf der Seite „Erweitert“ finden sich die weitere Optionen zur Feinsteuerung des Programmablaufs.

Das Bildschirmfoto unten zeigt den Aufbau der GUI (Betriebssystem GNU/Linux, unter anderen Systemen kann *Survey2GIS* abweichend aussehen):



Nachdem alle notwendigen Angaben auf der Seite „Einfach“ gemacht wurden, kann die Schaltfläche „Ausführen“ betätigt werden. Dies startet einen Programmdurchlauf von *Survey2GIS*. Während der Verarbeitung werden Fehlermeldungen, Warn- und Statushinweise im rechten Bereich (dunkelblaues Konsolefenster) angezeigt. Die Meldungen entsprechen exakt denjenigen der Kommandozeilenversion und dem Inhalt der optionalen Logdatei.

Warnungen erscheinen gelb. Sie führen nicht zum Programmabbruch, können aber wichtige Hinweise zu Problemen mit den Eingabedaten enthalten.

Fehlermeldungen erscheinen rot und führen zum Abbruch der Verarbeitung. In diesem Fall werden keine Ausgabedaten erzeugt.

Um Optionen nicht immer wieder neu einstellen zu müssen, können die aktuellen Einstellungen per „Einstellungen▷Speichern unter“ in eine Datei (Endung „.s2g“) gespeichert und per „Einstellungen▷Öffnen“ jederzeit wieder hergestellt werden. Gespeichert werden nur diejenigen Optionswerte, die auch tatsächlich eingestellt worden, nicht der Inhalt leerer Felder. So lassen sich z. B. alle Einstellungen bis auf die der Eingabedateien abspeichern, um diese bequem auf unterschiedliche Dateien anwenden zu können.

Die Funktion „Einstellungen▷Vorgaben“ (bzw. der Knopf gleichen Namens am unteren Rand des GUI) liest die Einstellungen aus der Datei namens „default.s2g“ im Programmverzeichnis von *Survey2GIS* – insofern eine solche Datei existiert.

Achtung: Im Ausgabeverzeichnis bereits existierende Ausgabedateien werden *ohne weitere Rückfrage überschrieben!* Dies gilt auch für eine evtl. existierende Logdatei. Wird eine Einstellungsdatei geöffnet, so wird deren Inhalt direkt und ohne weitere Rückfrage in das GUI übernommen.

### 3.1 Einbindung des GUI in aufrufenden Anwendungen

Die Einbindung in eine aufrufende Anwendung (z. B. ein GIS) wird von *Survey2GIS* auf zwei Wegen unterstützt:

1. Die aufrufenden Anwendungen kann sinnvolle Voreinstellungen als Optionswerte übergeben und das GUI dann mittels „--show-gui“ starten.
2. Während *Survey2GIS* läuft, werden auf dem Standardausgabekanal des Betriebssystems (der Konsole) speziell formatierte Meldungen ausgegeben. Die aufrufende Anwendung kann diese sammeln und, nachdem *Survey2GIS* vom Benutzer beendet wurde, analysieren und entsprechend reagieren (s. u.).

Sofern *Survey2GIS* Ausgabedaten erzeugt hat, werden die folgenden Tags in der Ausgabe zu finden sein:

```
<OUTPUT_FORMAT>format</OUTPUT_FORMAT>
```

Dabei steht *format* für die Bezeichnung des Ausgabeformats, z. B. „Esri Shapefile“.

Darauf folgen die Namen der einzelnen Ausgabedatensätze, nach Geometrien getrennt:

```
<OUTPUT_POINTS>data</OUTPUT_POINTS>  
<OUTPUT_LINES>data</OUTPUT_LINES>  
<OUTPUT_POLYGONS>data</OUTPUT_POLYGONS>
```

Wobei *data* der (Datei-)Name des zugehörigen Datensatzes ist.

Sollte der Benutzer die Option „--raw-data“ verwendet haben, so erscheint zusätzlich:

```
<OUTPUT_POINTS_RAW>data</OUTPUT_POINTS_RAW>
```

Da es möglich ist, dass der Benutzer mit *Survey2GIS* in mehreren Durchläufen unterschiedliche Ausgabeformate und -dateien erzeugt oder auch wieder überschrieben hat, muss die aufrufende Anwendung stets die gesamte Ausgabe prüfen.



## 4 Das Parserschema

Unter einem „Parser“ versteht man denjenigen Teil eines Programms, welcher die Eingabedaten nach einem vorgegeben Format („Syntax“) einliest und in ihre einzelnen Bestandteil aufbricht, so dass diese vom Programm verarbeitet werden können.

Da es sich bei den Eingabedaten, die von *Survey2GIS* verarbeitet werden, um reine Messdaten handelt, muss die Struktur der Daten zunächst beschrieben werden. Diese Aufgabe fällt dem Parserschema zu, welches mittels der Option *-p* an das Programm übergeben wird. Dabei handelt es sich um eine einfache Textdatei, in der Angaben zum Format der Eingabedateien gemacht werden können. Wie bereits erwähnt, kann *Survey2GIS* mehrere Eingabedateien gleichzeitig abarbeiten. Sie alle werden aber nach einem einzigen Parserschema abgearbeitet.

### 4.1 Syntax der Datei

Hinweis: Zur Veranschaulichung der hier beschriebenen Syntax empfiehlt es sich, die Datei „parser.txt“ im Verzeichnis „docs“ sowie den Inhalt der Dateien im Verzeichnis „samples“ zu betrachten.

Zeilen, die mit einem „#“ („Gitter“) beginnen, sind Kommentarzeilen. Sie werden vom Programm ignoriert, ebenso wie leere Zeilen.

Die Datei muss aus zwei grundsätzlichen Typen von Abschnitten bestehen, die mit einem Schlüsselwort in eckigen Klammern („[ ]“) eingeleitet werden. Die beiden Typen sind „[Parser]“ und „[Field]“. Der Abschnitt „[Parser]“ darf nur einmal auftauchen und enthält allgemeine Parser-Einstellungen. Danach können bis zu 251 Abschnitte mit „[Field]“ eingeleitet werden. Sie enthalten Einstellungen zu den einzelnen Datenfeldern der Eingabedatei(en). Die Datenfelder müssen in derselben Reihenfolge definiert werden, in der sie (von links nach rechts) in den Eingabedateien stehen.

Unter jedem Abschnitt macht der Benutzer Angaben in Form von Optionsnamen und Optionswerten (Einstellungen), die durch ein Gleichheitszeichen getrennt sind (*Option=Wert*). Ein solches Paar steht allein in einer Zeile, z. B.:

```
[Field]
Name=Indexfeld
```

Der gesamte Text rechts von „=“ wird als Optionswert interpretiert. Leere Optionswerte sind nicht erlaubt. Wenn eine Option nicht eingestellt werden soll, dann kann sie ganz weggelassen werden.

Bei der Eingabe von Schlüsselwörtern, Optionsnamen und Einstellungen innerhalb der Parser-Schemadatei werden Groß- und Kleinschreibung ignoriert. Die Namen von Optionen und die Werte ihrer Einstellungen können auch in Anführungszeichen gesetzt werden, zusätzliche Leerzeichen zwischen den beiden werden ignoriert:

```
[Field]
''Name'' = ''Indexfeld''
```

Die Namen von Optionen enthalten hingegen niemals Leerzeichen. Optionsnamen, die aus mehr als einem Wort bestehen, enthalten stattdessen den Unterstrich „\_“:

```
[Field]
Name = Indexfeld
empty_allowed = No
```

Einige Einstellungen, z. B. „empty\_allowed“ im obigen Beispiel, sind Wahrheitswerte oder „Schalter“, d. h. sie lassen sich nur an- oder ausschalten. Die folgenden Werte sind äquivalente Möglichkeiten, um eine Option einzuschalten: „Y“, „Yes“, „On“, „1“, „Enable“, oder „True“. Um eine Option abzuschalten, lassen sich verwenden: „N“, „No“, „Off“, „0“, „Disable“, oder „False“.

Die Korrektheit eines Parserschemas kann überprüft werden, indem *survey2gis* mit der Option „-v“ und „-p“, gefolgt vom Dateinamen des zu prüfenden Schemas, gestartet wird. Sollte das Schema Fehler enthalten, so wird eine entsprechende Meldung ausgegeben. Ist es fehlerfrei, dann wird eine Zusammenfassung des Schemas ausgegeben.

Im Folgenden werden alle Einstellungen für die Abschnitte „[Parser]“ und „[Field]“ besprochen.

## 4.2 Einstellungen für [Parser]

Die folgenden Einstellungen (hier in alphabetischer Reihenfolge) können oder müssen im Abschnitt „[Parser]“ gemacht werden. Sie beschreiben allgemeine Einstellungen des Parsers. Zu *Survey2GIS* gehört die mitgelieferte Datei „parser.txt“ im Verzeichnis „docs“. Sie enthält ein (nicht für tatsächliche Eingabedateien verwendbares) Parserschema, das sämtliche Einstellungen und kurze Erklärungen enthält.

---

Option: **comment\_mark**

Bedeutung: Zeichen oder Zeichenkette(n), welche in den Eingabedaten Kommentarzeilen einleiten. Kommentarzeilen werden bei der Verarbeitung übersprungen.

Bemerkungen: Optional. Es ist möglich, mehrere Zeichen(ketten) anzugeben. Darunter dürfen sich jedoch keine Zeichen befinden, welche auch als Feldtrenner oder Anführungszeichen verwendet werden (s. „quotation“ und „separator“ in 4.3).

Beispiel:

```
comment_mark = #  
comment_mark = //
```

---

Option: **coor\_x**, **coor\_y**, **coor\_z**

Bedeutung: Diese drei Einstellungen geben die Namen der Felder an, in welchen sich die Messwerte der X-, Y- und Z-Koordinaten befinden.

Bemerkungen: Die Einstellungen für „coor\_x“ und „coor\_y“ müssen stets gemacht werden und die Namen gültiger, definierter Felder (s. nächster Abschnitt) in der/n Eingabedatei(en) enthalten. Die Felder müssen numerisch und vom Typ „Double“ sein. Wenn „coor\_z“ nicht angegeben wird, so werden alle Z-Koordinaten per Voreinstellung als „0“ gespeichert.

Beispiel:

```
coor_x = xfield  
coor_y = yfield  
coor_z = zfield
```

---

Option: **geom\_tag\_point**, **geom\_tag\_line**, **geom\_tag\_poly**

Bedeutung: Angabe der Geometriemarker für die drei Grundtypen „Punkt“, „Linie“ und „Polygon“. Diese Marker versetzen den Parser in die Lage, Messungen zu komplexeren Geometrien zusammenzufügen. Sie müssen als Teil der Messdaten eingegeben werden.

Bemerkungen: Die spezifische Art und Weise, auf welche die Geometriemarker vom Parser verarbeitet werden, hängt vom gewählten Parsermodus (s. 4.4 auf Seite 36) ab. Die Angabe für „geom\_tag\_point“ ist für

bestimmte Parsermodi optional. Wird sie nicht gemacht, dann versucht das Programm, Punktmessungen automatisch als solche zu erkennen. Dies funktioniert jedoch nur unter bestimmten Voraussetzungen zuverlässig (beachte Hinweise in Abschnitt 12). Für alle Parsermodi außer „None“ müssen „geom\_tag\_line“ und „geom\_tag\_poly“ auf unterschiedliche Zeichen gesetzt werden. Im Modus „None“ bedarf es keinerlei Definitionen für die Geometriemarker; evtl. dennoch definierte Marker werden in diesem Modus ignoriert. Es ist möglich, mehr als einen Geometriemarker (bis zu 32 verschiedene) für jeden Geometrietyp zu definieren. Dabei dürfen die Marker jedoch in ihren Zeichenketten nicht überlappen (es wird zwischen Groß- und Kleinschreibung unterschieden). Die Zeichenketten von Geometriemarkern dürfen auch keine gemeinsamen Zeichen mit denen von Feldtrennern oder Anführungszeichen in Felddefinitionen (s. 4.3) oder Kommentarzeichen (s. „comment\_mark“, oben) besitzen. Weiterhin dürfen die Definitionen von Geometriemarkern keine Zahlen, Plus („+“) oder Minus („-“) enthalten, wenn das Feld, welches den Marker enthält ein numerisches ist. Handelt es sich um den Feldtyp „Double“, dann sind auch der Dezimalpunkt und das Tausendertrennzeichen nicht erlaubt (s. „tag\_field“, unten).

Beispiel:

```
geom_tag_point = "*"
geom_tag_line = "-"
geom_tag_poly = "@"
geom_tag_poly = "@_"
```

---

Option: **info**

Bedeutung: Kurze Beschreibung des Parsers.

Bemerkungen: Optional. Kann weggelassen werden.

Beispiel:

```
info = This parser has been designed to test Survey2GIS
```

---

Option: **key\_field**

Bedeutung: Name des einen Feldes, welches für jede komplexe Geometrie (Linienzug oder Polygon) einen Schlüsselwert enthält, der für alle Datensätze welche zu der Geometrie gehören, denselben Wert enthält. In den Parsermodi „End“ und „Max“ werden aufeinanderfolgende Messungen mit demselben Wert in „key\_field“ zu Geometrien zusammengefügt. In diesem Zusammenhang bedeutet „aufeinanderfolgend“ dass Datensätze bis zu und inklusive dem nächsten, der einen Linien- oder Polygonmarker (s. „tag\_field“, unten) enthält, zusammengeführt werden. Die Option „key\_unique“ (s. unten) kann genutzt werden um mehrteilige Geometrien zu erzeugen.

Bemerkungen: Muss für die Parsermodi „End“ und „Max“ angegeben werden.

Beispiel:

```
key_field = key
```

---

Option: **key\_unique**

Bedeutung: Gibt an, ob das Schlüsselfeld auch für disjunkte Geometrien eindeutig ist.

Bemerkungen: Wenn diese Option eingeschaltet ist, dann geht *survey2gis* davon aus, dass jede Geometrie einen global einmaligen (engl. *unique*) Wert besitzt. Dies kann dazu genutzt werden, komplexe Geometrien (Mengen von Linien oder Polygonen) aus *disjunkten* Teilen als zusammengehörige Objekte zu speichern. Alle Geometrien, welche denselben Wert im *key\_field* (s. o.) besitzen werden dann als ein einziges Objekt in der Ausgabedatei abgelegt. Dies hat zur Folge, dass ihnen nur ein gemeinsamer Attributdatensatz zugewiesen wird und dass sie z. B. bei einer Selektion oder Editierung im GIS als Einheit behandelt werden. Wenn diese Option nicht gesetzt wird, dann ist es *survey2gis* nicht möglich, disjunkte Teile von Geometrien automatisch zusammenzuführen. In diesem Fall muss die Zusammenführung im GIS, bswp. über ein Attributfeld mit entsprechendem Schlüsselwert, erfolgen.

Beispiel:

```
key_unique = yes
```

---

Option: **name**

Bedeutung: Name des Parsers.

Bemerkungen: Optional. Kann weggelassen werden.

Beispiel:

```
name = survey2gis test parser
```

---

Option: **no\_data**

Bedeutung: Ein *Integerwert* (Ganzzahl), der für „Null“ oder „keine Daten“ (engl. „no data“) steht. Hierfür sollte ein Wert gewählt werden, der in den tatsächlichen Messdaten niemals vorkommt. Der gewählte Null-Wert wird immer dann in die Ausgabedaten geschrieben, wenn ein Datensatz keinen Wert für ein Feld enthält. Siehe auch „empty\_allowed“ in den Einstellungen für Felder ([4.3](#)).

Hinweis: Optional. Siehe auch Hinweise zu den einzelnen Ausgabeformaten in [10](#), welche evtl. automatische „Null“-Werte wählen, wenn der Benutzer keinen vorgibt.

Example:

```
no_data = -99999
```

---

Option: **tag\_field**

Bedeutung: Name des einen Feldes, welches die Geometriemarker enthält (s. [4.4 auf Seite 36](#)). Kann mit „key\_field“ (s. o.) identisch sein, allerdings nicht im Parsermodus „Max“. Muss den Namen eines per „[Field]“ deklarierten Feldes enthalten (s. [4.3 auf der nächsten Seite](#)).

Bemerkungen: Muss für alle Parsermodi außer „None“ angegeben werden. Geometriemarker können, um die Tipparbeit im Feld zu reduzieren, während der Messung als Teil des Inhalts regulärer Attributfelder kodiert werden (Details s. [4.4](#)). In diesem Fall dürfen die Attributwerte allerdings nicht mit den Zeichenketten der Geometriemarker überlappen. Koordinatenfelder (s. „coord\_x“, „coord\_y“, „coord\_z“, oben) können ebenfalls nicht hierfür verwendet werden. Im Parser-Modus „Min“ macht es üblicherweise Sinn, ein separates Feld für „tag\_field“ zu wählen, da in diesem Modus pro Geometrie nur eine vollständige Zeile mit Attributdaten eingelesen wird.

Beispiel:

```
tag_field = tag
```

---

Option: **tagging\_mode**

Bedeutung: Bestimmt die grundsätzliche Arbeitsweise des Parsers (*Parsermodus*). Mögliche Werte sind „min“, „max“, „end“ und „none“. Weitere Erklärungen hierzu in [4.4 auf Seite 36](#).

Bemerkungen: Muss angegeben werden.

Beispiel:

```
tagging_mode = end
```

---

Option: **tag\_strict**

Bedeutung: Wenn diese Option eingeschaltet ist, dann werden Geometriemarker from Parser strikt behandelt, d. h. jede Messung (Zeile) in den Eingabedaten muss einen Geometriemarker enthalten. Messungen mit fehlenden Markern werden nicht automatisch als Punkte betrachtet, sondern verworfen.

Bemerkungen: Optional. Wenn nicht angegeben, dann ausgeschaltet.

Beispiel:

```
tag_strict = yes
```

### 4.3 Einstellungen für [Field]

Auf die allgemeinen Einstellungen unter „[Parser]“ (s. vorheriger Abschnitt) folgen zwei oder mehr (bis zu 251) Felddefinitionen, die *jeweils* mit der Zeile

[Field]

eingeleitet werden. Die hier definierten Felder können zusätzlich zu den eingemessenen Koordinaten weitere Daten (sog. „Attributwerte“) aufnehmen, welche vom Vermessungsgerät oder dem Benutzer selbst den Eingabedaten hinzugefügt werden. Für die Vermessung und automatische Rekonstruktion komplexerer Geometrien (z.B. Polygone) durch *survey2gis* wird zumindest ein zusätzliches Feld benötigt, welches die Geometriemarkierung und den eindeutigen Schlüsselwert der Geometrie enthält (s. „key field“ und „tag field“ im vorigen Abschnitt).

Die definierten Attributfelder werden in den Ausgabedaten gespeichert, sofern das gewählte Ausgabeformat dies unterstützt. Im Falle von Shapefiles werden sie in einer Attributtabelle im *DBase*-Format (Dateiendung „.dbf“) gespeichert.

Die Definition von Attributfeldern unterliegt gewissen Einschränkungen, welche u.a. vom Ausgabeformat abhängig sind. Weitere Informationen sind im Abschnitt [10 auf Seite 75](#) zu finden.

---

Option: **change\_case**

Bedeutung: Die Option „change\_case“ kann gesetzt werden, um den Inhalt eines Textfeldes direkt nach dem Einlesen vollständig in Groß- oder Kleinbuchstaben umzuwandeln. Die möglichen Werte sind „lower“ (in Kleinbuchstaben umwandeln), „upper“ (in Großbuchstaben umwandeln) und „none“ (keine Umwandlung).

Bemerkungen: Optional. Kann nur für Textfelder gesetzt werden.

Beispiel:

```
change_case = lower
```

---

Option: **empty\_allowed**

Bedeutung: Gibt an, ob der Inhalt eines Feldes vollständig leer sein darf. Dies ist z. B. der Fall, wenn in einer Zeile mit Eingabedaten zwei Feldtrenner direkt aufeinander folgen oder das Feld nur Leerzeichen enthält. In diesem Fall wird der Feldinhalt als Nullwert gespeichert, wenn die Option „empty\_allowed“ angeschaltet ist. Andernfalls wird die gesamte Messung (Zeile) als ungültig verworfen.

Bemerkungen: Optional. Wenn die Einstellung nicht explizit angegeben wird, so ist sie ausgeschaltet. Diese Einstellungen kann für die Felder, welche als „key\_field“ oder „tag\_field“ oder als Koordinatenfelder dienen, nicht angeschaltet werden. Sie kann ebenfalls nicht angeschaltet werden, wenn „merge\_separators“ (s.u.) angeschaltet ist.

*Wichtig:* Wenn der Feldtrenner für das betreffende Feld auf ein „Leerzeichen“ („space“ oder „tab“) eingestellt wird, dann kann die Option „empty\_allowed“ nicht korrekt verarbeitet werden: Da Leerzeichen links und rechts vom Feldinhalt vor dessen Speicherung automatisch entfernt werden, kann der Parser in diesem Fall nicht zwischen leerem Feldinhalt und Feldtrenner unterscheiden.

Beispiel:

```
empty_allowed = yes
```

---

Option: **info**

Bedeutung: Einzeilige Beschreibung des Feldinhalts.

Bemerkungen: Optional. Dient der Dokumentation der Feldbedeutung.

Beispiel:

```
info = Field with unique object index.
```

---

Option: **merge\_separators**

Bedeutung: Gibt an, ob mehrere aufeinander folgenden Feldtrenner wie ein einziger behandelt werden sollen.

Bemerkungen: Optional. Wenn ausgeschaltet (Voreinstellung), dann bedeuten zwei direkt aufeinander folgende Feldtrenner, dass sich dazwischen ein leerer Feldinhalt befindet (s. auch „empty allowed“ oben).

Beispiel:

```
merge_separators = yes
```

---

Option: **name**

Bedeutung: Der Name des Feldes, welcher maximal 10 Zeichen umfassen darf. Erlaubt sind nur Buchstaben des lateinischen Alphabets, der Unterstrich „\_“, sowie die Ziffern „0“ bis „9“. Es wird nicht nach Groß- und Kleinschreibung unterschieden. Alle Feldnamen werden automatisch in *Kleinschreibung* abgespeichert (Hinweis: dies wird seit Version 1.4.0 korrekt umgesetzt).

Bemerkungen: Muss angegeben werden.

Beispiel:

```
field = index_fld
```

---

Option: **persistent**



Bedeutung: Diese Option hat nur im Parsermodus „Min“ (s. 4.4 auf Seite 36) eine Bedeutung. Ist sie für ein Feld eingeschaltet, so bedeutet dies, dass der Parser auch in solchen Messungen (Zeilen), die keine Attributdaten enthalten, das betreffende Feld erwartet. Diese Einstellung ist z.B. dann nützlich, wenn die herstellerspezifische Software, welche den Speicher des Vermessungsgeräts ausliest, zusätzliche Daten in jede Zeile schreibt.

Bemerkungen: Optional. Koordinatenfelder sind im Parsermodus „Min“ stets persistent.

Beispiel:

```
persistent = yes
```

---

Option: **quotation**

Bedeutung: Falls der Inhalt eines Textfeldes in Anführungszeichen steht und diese nicht mit dem Feldwert abgespeichert werden sollen, dann muss das Zeichen, welches als Anführungszeichen dient, hier angegeben werden.

Bemerkungen: Optional. Es kann nur ein Anführungszeichen pro Feld angegeben werden. Dieses darf kein Zeichen sein, welches auch als Feldtrenner (s. u., „separator“) oder Kommentarzeichen (s. „comment.mark“ in 4.2) verwendet wird. Nur auf Textfelder anwendbar.

```
quotation = ' 
```

---

Option: **skip**

Bedeutung: Mit der Einstellung „skip“ lässt sich festlegen, dass das betreffende Feld zwar vom Parser verarbeitet, aber nicht in der Ausgabedatei abgespeichert wird. Nützlich für Felder deren Inhalt für die weitere Datenverarbeitung nicht wichtig ist.

Bemerkungen: Optional. Per Voreinstellung wird jedes definierte Feld auch in der Ausgabedatei gespeichert.

Beispiel:

```
skip = yes
```

---

Option: **separator**

Bedeutung: Für jedes Feld können ein oder mehrere Feldtrenner angegeben werden, welche den Inhalt des Feld zu dem des nächsten Feldes abgrenzen. Feldtrenner können auch aus Zeichenketten mit mehreren Buchstaben bestehen. Die speziellen Feldtrennerwerte „tab“ und „space“ stehen für das Tabulator- und Leerzeichen.

Bemerkungen: Muss für jedes Feld mit Ausnahme des letzten angegeben werden. Darf kein(e) Zeichen sein, welche(s) auch als Anführungszeichen (s. o., „quotation“) oder Kommentarzeichen (s. „comment\_mark“ in 4.2) verwendet wird/werden. S. auch die Option „merge\_separators“ (oben).

Beispiel:

```
separator = ;  
separator = tab  
separator = space
```

---

Option: **type**

Bedeutung: Diese Option bestimmt den Datentyp des Für jedes Feldes. Zur Auswahl stehen die Typen „integer“ (für Ganzzahlen: -1, 0, 100, etc.), „double“ (für Realzahlen: -10,05, 0.0, 1200,12, etc.) und „text“ (für Text aller Art; einige Ausgabeformate begrenzen die Länge von Textfeldern allerdings stark: s. jeweils den Eintrag in 10). Bei der Verarbeitung von Feldern mit Realzahlen ist unbedingt darauf zu achten, die mathematische Notation, d. h. Dezimaltrennzeichen und Tausendertrennzeichen, korrekt einzustellen (s. 2.2 auf Seite 11).

Bemerkungen: Muss für jedes Feld angegeben werden.

Beispiel:

```
type = integer
```

---

Option: **unique**

Bedeutung: Wenn diese Option angeschaltet ist, dann wird während der Verarbeitung der Eingabedaten getestet, ob der Inhalt des betroffenen Feldes *pro rekonstruierter Geometrie* einen einmaligen Wert besitzt. Wenn dem nicht so ist, wird von *Survey2GIS* eine entsprechende Warnung ausgegeben. Die betroffene(n) Geometrie(n) wird/werden aber nicht verworfen oder anderweitig verändert. Der Benutzer muss selbst entscheiden, wie mit mehrfach vorhandenen Attributwerten in den Ausgabedaten zu verfahren ist.

Bemerkungen: Optional. Wenn die Einstellung nicht explizit angegeben wird, so ist sie ausgeschaltet. Die Einmaligkeit des Feldwerts wird über alle Eingabedateien geprüft.

Beispiel:

```
unique = yes
```

---

Option: **value**

Bedeutung: Mit der Option „value“ kann ein „Pseudofeld“ definiert werden, welches nicht in den Eingabedaten existiert, aber in den Ausgabedaten mit einem konstantem Wert, der durch „value“ angegeben wird,

erscheinen soll. Solche Felder werden vereinfacht definiert. Es werden nur die Optionen „info“, „name“, „type“ und „value“ verarbeitet. Andere Feldoptionen führen zu einer Fehlermeldung.

Bemerkungen: Der Datentyp des Feldes (s. Option „type“) muss zum gewählten Wert passen. Bei Realzahlen sind außerdem evtl. die Spracheinstellungen zu beachten (s. [2.2 auf Seite 11](#)).

Beispiel (für ein Feld vom Typ „double“):

```
value = 1000,05
```

---

Option: @ (Ersetzung)

Bedeutung: Mit dem Zeichen „@“ wird eine Ersetzung von Textfeldinhalten eingeleitet. Diese Funktion dient dazu, wiederkehrende kurze Textfeldinhalte (Kürzel) durch längere Beschreibungen zu ersetzen und die Langform in der Attributtabelle zu speichern. Die Syntax für die Ersetzung lautet:

```
@alt=neu
```

Damit wird die Zeichenfolge „alt“ durch „neu“ ersetzt, wann immer erstere in den Daten des Textfeldes erscheint. Dabei muss „alt“ der *gesamte* Inhalt des Textfeldes sein. Ersetzungen von Teilinhalten werden nicht durchgeführt. Es ist möglich, mehr als nur eine Ersetzung pro Textfeld zu definieren. Zwischen „@“ und „alt“ (der zu ersetzenden Zeichenfolge) dürfen keine Leerzeichen, Tabulatoren oder sonstige Zeichen stehen.

Bemerkungen: Diese Option ist nur für Textfelder gültig. Beim Vergleich zwischen altem und neuem Inhalt wird die Groß- und Kleinschreibung *nicht* beachtet. Der neue Textfeldinhalt darf eine Länge von 254 ASCII-Zeichen nicht überschreiten.

Beispiel (für ein Feld vom Typ „text“):

```
@ABK = Abkürzung
```

```
@P1 = Punktyp 1
```

```
...
```

## 4.4 Geometriemarker und Parsermodi

Um komplexe Geometrien wie Linien(-züge) und Polygone aus den Messdaten rekonstruieren zu können, muss der Parser wissen, wie er die einzelnen Zeilen mit den Messdaten der Stützpunkte zu verflechten hat. Grundsätzlich sind hierbei folgende Einschränkungen zu beachten:

- Obwohl *Survey2GIS* beliebig viele Eingabedateien verarbeiten kann, ist es nicht möglich, Stützpunkte für eine Geometrie über mehrere Eingabedateien zu verteilen. Alle Messdaten, die zu einer Geometrie gehören, müssen in ein und derselben Eingabedatei stehen. Allerdings können disjunkte Teile einer komplexen Geometrie (s. Beschreibung der Parser-Option „key\_unique“) über mehrere Dateien verteilt werden, so dass es z. B. möglich ist, eine sehr komplexe Geometrie in mehreren Teilen und mit unterschiedlichen Geräten einzumessen.
- Die Liste der einzelnen Stützpunkte in der Eingabedatei darf nur von Kommentarzeilen oder leeren Zeilen unterbrochen werden. Es ist somit nicht möglich, während der Einmessung einer Geometrie im Feld die Messung zu unterbrechen, eine andere Geometrie einzumessen und dann die Messungen zur ursprünglichen Geometrie fortzusetzen. Ist ein solcher Arbeitsablauf im Feld unvermeidlich, so muss die resultierende Datei manuell bereinigt werden, bevor *Survey2GIS* sie erfolgreich verarbeiten kann.
- Es können nur die drei Geometrien „Punkt“, „Linie“, und „Polygon“ verarbeitet. Andere, insbesondere gerundete Typen wie „Ellipsen“ müssen mit der weiter verarbeitenden Software daraus erzeugt werden.

Hinweis: Weitere Hinweise zu komplexeren Geometrien wie Polygone mit Löchern oder separaten Teilen finden sich im Abschnitt „Topologische Bereinigung“ ( 9 auf Seite 62).

Jeder Geometrietyt wird dem Parser durch einen frei definierbaren Geometriemarker (s. 4.2 auf Seite 27) angezeigt. Bei diesen Markern handelt es sich normalerweise um sonst nicht benötigte Zeichen wie „@“ oder „\$“. Im Feld werden sie den Messungen hinzugefügt, entweder als separates Attributfeld oder am Ende des Inhalts eines bestehenden Feldes (je nach gewähltem Parsermodus). Das Feld, welches die Geometriemarker der Messungen enthält, wird als „Markierungsfeld“ bezeichnet. Es wird vom Benutzer als Teil des Parserschemas (Option „tag\_field“, s. 4.2 auf Seite 27) definiert. Wenn dies unterbleibt, so interpretiert *Survey2GIS* jede Messung/Zeile in den Eingabedaten automatisch als einzelne Punktmessung (also nicht als Teil einer Linie oder eines Polygons).

Welche Punkte zu ein und derselben Geometrie gehören, wird dem Parser in einigen Modi zusätzlich durch ein Schlüsselfeld („key\_field“) mitgeteilt. Die Prüfung des Inhalts des Schlüsselfelds berücksichtigt Groß- und Kleinschreibung. Wird ein Textfeld als Schlüsselfeld verwendet, so können einzelne Stützpunkte also nur dann korrekt einer gemeinsamen Geometrie zugeordnet werden, wenn sich auch die Groß- und Kleinschreibung ihrer Schlüsselwerte nicht unterscheidet. Um dieses Problem zu vermeiden, kann die Feldoption „change\_case“ gesetzt werden, welche eine einheitliche Schreibung des Feldinhalts erzwingt.

Der letzte Stützpunkt eines Polygons muss nicht dem ersten entsprechen, d. h. eine doppelte Einmessung braucht nicht vorgenommen werden. Polygone werden beim Datenexport automatisch geschlossen.

Tatsächlich stellt die doppelte Einmessung eines Stützpunktes einen topologischen Fehler dar, welcher von *Survey2GIS* automatisch bereinigt werden kann. Mehr zu diesem Thema im Abschnitt [9 auf Seite 62](#).

Je nach Natur der Feldarbeiten kann es sinnvoll sein, unterschiedliche Arbeitsabläufe zur Markierung von Geometrien zu verwenden. Daher stellt *Survey2GIS* eine Reihe alternativer, hier als *Parsermodi* bezeichneter, Vermessungsmethodiken zur Verfügung. Sie werden im Parserschema mit der Option „tagging\_mode“ gesetzt:

```
[Parser]
...
tagging_mode = min
...
```

Pro Parserschema kann nur jeweils ein Parsermodus angegeben werden. Dies bedeutet auch, dass *sämtliche Eingabedateien* im selben Modus abgearbeitet werden. Die vier möglichen Parsermodi heißen „Min“, „Max“, „End“ und „None“.

Hinweis: Wenngleich es möglich ist, jedes Feld vom Typ „Text“ zum Speichern von Geometriemarkern zu verwenden (in einigen Parsermodi, s. unten, kann dasselbe Feld sogar als „tag\_field“ und „key\_field“ deklariert werden), wird dies nicht empfohlen. Stattdessen sollte ein eigenes Feld für die Marker verwendet werden. Sollten Geometriemarken dennoch als Teil eines anderen Textfelds gespeichert sein, dann *müssen* diese entweder als erstes oder letztes Zeichen des Feldinhalts stehen.

#### 4.4.1 „Min“: Minimale Kodierarbeit

Dieser Modus ist die Voreinstellung, die verwendet wird, wenn „tagging\_mode“ nicht explizit vom Benutzer als Teil des Parserschemas gesetzt wird. Er eignet sich zur Minimierung des Messaufwandes für Linienzüge und Polygone. In diesem Modus *muss* die Messung des ersten Stützpunktes eines Linienzugs oder Polygons mit dem entsprechenden Geometriemarken versehen werden.

Das Auftauchen eines Geometriemarkers leitet die Liste der weiteren Stützpunkte ein. Nur die erste Stützpunktmessung enthält Attributdaten. Alle weiteren Messungen/Zeilen sind *reduziert*; sie enthalten lediglich Koordinatenmessungen und unvermeidbare, evtl. vom Vermessungsgerät oder seiner Treibersoftware eingefügte, Felder. Letztere *müssen* vom Benutzer, als Teil der Felddefinition im Parserschema, mittels der Option „persistent“ (zu deutsch etwa „hartnäckig“; s. Beschreibung der Einstellungen für Felder in [4.3](#)) gekennzeichnet werden, so dass der Parser erkennen kann, dass es sich um reduzierte Stützpunktmessungen handelt.

Der Vorteil dieses Modus liegt darin, dass die Tipparbeit zur Attributdatenkodierung im Feld auf ein Minimum reduziert wird. Die Eingaben sind nur einmal, bei der Einmessung des ersten Stützpunktes der Geometrie, vorzunehmen. Es ist nicht notwendig, ein „key\_field“ im Parserschema anzugeben, es sei denn disjunkte Geometries sollen verarbeitet werden.

Beispiel einer Eingabedatei für den Modus „Min“ (Auszug):

```

# Sample file structure for a polygon (square) in mode „Min‘‘.
# Hashmark introduces a comment line that will not be parsed.
# Fields: index, label, geometry tag (@=polygon).
1 polygon_1 @ 10.00 10.00 1.00
# The first line had all the attributes and the geometry tag.
# The following lines will only have the x/y/z vertex coordinates.
10.00 20.00 1.05
20.00 20.00 1.10
20.00 10.00 1.00
# The next occurrence of a full record terminates the polygon.
2 polygon_2 @ 30.00 30.00 2.50
...

```

Um Konsistenz zu wahren, müssen persistente und Koordinatenfelder sowohl in attribuierten als auch reduzierten Messungen in derselben Reihenfolge stehen, in der sie ursprünglich definiert wurden. Angenommen, im Beispiel oben wäre das Feld „label“ als „persistent“ gesetzt worden, dann hätte ein gültiger Datensatz folgende Struktur:

```

1 polygon_1 @ 10.00 10.00 1.00
polygon_1 10.00 20.00 1.05
polygon_1 20.00 20.00 1.10
polygon_1 20.00 10.00 1.00

```

Die minimale Zahl der Felder,  $Min(f)$ , einer reduzierten Messung im Modus „Min“ ergibt sich demnach als:

$$Min(f) = \text{Anzahl Koordinatenfelder (2 oder 3)} + \text{Anzahl persistenter Felder}.$$

#### 4.4.2 „Max“: Maximale Kontrolle

Im Modus „Max“ muss jede einzelne Messung/Zeile mit einem Geometriemarker gekennzeichnet werden (gleichzeitig muss im Parserschema für jeden Geometriety ein Geometriemarker definiert sein). Dieser Modus ist vor allem dazu gedacht, die kontrollierte Aufbereitung von Daten durch *Survey2GIS* zu ermöglichen, welche nicht ausschließlich von Vermessungsgeräten produziert wurden. So lassen sich mithilfe eines Texteditors bequem und kontrolliert Geometrien konstruieren und im Modus „Max“ konsistente GIS-Daten daraus produzieren. Der Modus kann auch nützlich sein, wenn Datensätze (z. B. aus einer Datenbank) als Zeilen exportiert und mit *Survey2GIS* aufbereitet werden sollen.

Es ist notwendig, sowohl ein „tag\_field“ als auch ein „key\_field“ als Teil des Parserschemas zu definieren. Dabei muss es sich um separate Felder handeln. Prinzipiell reicht es aus, wenn der Inhalt von „key\_field“

sich von einer Geometrie zur nächsten ändert. Im Interesse der Datenintegrität ist es aber ratsam, den Inhalt von „key\_field“ so zu setzen, dass er für jede Geometrie (über alle Eingabedaten) eindeutig ist.

Beispiel einer Eingabedatei für den Modus „Max“ (Auszug):

```
# Sample file structure for a polygon (square) in mode „Max''.
# Hashmark introduces a comment line that will not be parsed.
# Fields: index, label, geometry tag (@=polygon).
# The second field (label) is the key field.
1 polygon_1 @ 10.00 10.00 1.00
2 polygon_1 @ 10.00 20.00 1.05
3 polygon_1 @ 20.00 20.00 1.10
4 polygon_1 @ 20.00 10.00 1.00
# A change in the value of the key field terminates the polygon.
5 polygon_2 @ 30.00 30.00 2.50
...
```

Hinweis: Die Stützpunkte von Linien und Polygonen, die im Modus „Max“ kodiert werden, müssen *direkt aufeinanderfolgen*. Es ist nicht möglich, z. B. Einzelpunktmessungen zwischen die Stützpunkte eines Polygons zu einzustreuen!

#### 4.4.3 „End“: Flexible Vermessung

In diesem Modus müssen alle Einmessungen von Linien und Polygonen mit einem Geometriemarker *abgeschlossen* werden, d. h. der Marker muss als Teil des letzten Stützpunktes eingemessen werden.

Für die Feldarbeit besteht der Vorteil im Modus „End“ darin, dass erst mit der abschließenden Messung festgelegt werden muss, ob ein Objekt als Linienzug oder Polygon repräsentiert werden soll. Damit der Parser die der Abschlussmessung vorangehenden Stützpunkte korrekt zusammenstellen kann, müssen alle *zusammengehörigen* Messungen mit einem identischen Wert im Schlüsselfeld („key\_field“) versehen werden, was entsprechenden Mehraufwand bei der Kodierung im Rahmen der Vermessung erfordert.

Der Modus „End“ erfordert vom Benutzer die Definition von Schlüsselfeld („key\_field“) und Markierungsfeld („tag\_field“) als Teil des Parserschemas. Es kann sich dabei um dasselbe Feld handeln. Der Geometriemarker wird nicht als Teil der Ausgabedaten gespeichert. Prinzipiell reicht es aus, wenn der Inhalt von „key\_field“ sich innerhalb derselben Geometrie *nicht* ändert, der der Geometriemarker das Ende einer Geometrie zuverlässig anzeigt. Im Interesse der Datenintegrität ist es aber ratsam, den Inhalt von „key\_field“ so zu setzen, dass er für jede Geometrie (über alle Eingabedaten) eindeutig ist.

Beispiel einer Eingabedatei für den Modus „End“ (Auszug):

```
# Sample file structure for a polygon (square) in mode „End''.
# Hashmark introduces a comment line that will not be parsed.
```

```

# Fields: index, label with embedded geometry tag (@=polygon).
# The second field (label) is also the key field.
1 polygon_1 10.00 10.00 1.00
2 polygon_1 10.00 20.00 1.05
3 polygon_1 20.00 20.00 1.10
4 polygon_1@ 20.00 10.00 1.00
# After the occurrence of „@’’ (geometry tag), another object follows.
5 polygon_2 30.00 30.00 2.50
...

```

#### 4.4.4 „None”: Vermessung von reinen Punktfeldern

Im Modus „None” erfolgt keinerlei Markierung von Geometrien. Dieser Modus ist für Vermessungen gedacht, welche lediglich Punktmessungen umfassen. Ein Beispiel wäre das Einmessen eines Höhenmodells in Form von 3D-Höhenpunkten, welche keinerlei zusätzliche Attributfelder benötigen.

Dieser Parsermodus erlaubt es, *Survey2GIS* als flexibles Werkzeug für die Verarbeitung von einfachen Punktlis-ten zu verwenden.

In den Modi „Min,” „Max” und „End” müssen einfache Punktmessungen nur dann mit einem Geometriemarker versehen werden, wenn die Parseroption „tag\_strict” eingestellt ist. Ansonsten werden einfache Punktmessungen automatisch als solche erkannt.

Beispiel einer Eingabedatei für den Modus „None” (Auszug):

```

# Sample file structure for a polygon (square) in mode „None’’.
# Hashmark introduces a comment line that will not be parsed.
# Fields: index, X, Y, Z.
1 10.00 10.00 1.00
2 10.00 20.00 1.05
3 20.00 20.00 1.10
4 20.00 10.00 1.00
5 30.00 30.00 2.50
...

```



## 5 Etiketten (Beschriftungen/Labels)

Zusätzlich zu den eigentlichen Messdaten lassen sich mit *Survey2GIS* auch *Etiketten* (Beschriftungen, engl. „labels“) für die Produktion von Plänen und Karten erzeugen. Etiketten sind 2D-Punkte (Z-Werte werden bei ihrer Platzierung nicht berücksichtigt) mit speziellen Attributfeldern. Das tatsächliche Ausgabeformat und das Schema der Attributdaten sind vom gewählten Ausgabeformat der Messdaten abhängig (Details s. 10).

Vorsicht: Ausgabeformate, welche zusätzliche Ausgabedateien für Etiketten erzeugen (z. B. Shapefiles: s. 10.1), überschreiben ohne Rückfrage evtl. vorhandene Dateien. Da für die Benennung von Dateien mit Etiketten automatisch erzeugte Namen verwendet werden, kann dies unerwünschte Ergebnisse produzieren. Die erzeugten Namen verwenden „\_labels“ als Suffix des Dateinamens. Bitte prüfen Sie vorher, ob Dateien mit demselben Namensschema bereits existieren und beibehalten werden sollen!

Die Erzeugung von Etiketten erfordert die Angabe eines Feldes in den Eingabedaten. Der Inhalt dieses Feldes wird für die Etikettierung der jeweiligen Geometrien verwendet. Es ist möglich, ein Feld zu wählen, das in der Parserdefinition als „skip=true“ angegeben ist (s. 4.3). Das Feld kann jeden Typ haben, der von *survey2gis* unterstützt wird (d. h. „Text“, „Integer“ oder „Double“); sein Inhalt wird ohne weitere Konvertierung für die Etikettierung übernommen.

Der Benutzer hat ein gewisses Maß an Kontrolle über die Platzierung der Etiketten auf den Geometrien. Die wählbaren Modi sind: „center“ (Voreinstellung), „first“, „last“ und „none“. Das Resultat hängt vom Geometrietyp ab:

- Punkte: Der gewählte Modus wird weitgehend ignoriert; es wird jeweils ein Etikett pro Punkt exakt auf letzterem platziert. Sollte allerdings der Modus „none“ gewählt sein, so werden keine Etiketten für Punkte erzeugt. Rohdaten werden niemals etikettiert (s. Option „--raw-data“ in Abschnitt 2).
- Linien: Im Modus „center“, werden die Etiketten jeweils auf halber Strecke entlang der Linien platziert. Im Modus „first“ werden die Etiketten jeweils auf den ersten Stützpunkten der Linien platziert und im Modus „last“ auf den letzten. In allen drei Fällen ist garantiert, dass die Etiketten exakt auf den Linien platziert werden.
- Polygone: Im Modus „center“, werden die Etiketten jeweils im geometrischen Schwerpunkt der Polygone platziert. Im Modus „first“ werden die Etiketten jeweils auf den ersten Stützpunkten der Polygone platziert und im Modus „last“ auf den letzten. Im Modus center kann *nicht* garantiert werden, dass die Etiketten innerhalb der jeweiligen Polygone liegen, wenn letztere Löcher enthalten oder stark konkav geformt sind.

Der Modus „none“ hat in allen Fällen denselben Effekt und sorgt dafür, dass für den jeweiligen Geometrietyp keine Etiketten erzeugt werden.

Bei mehrteiligen Geometrien wird ein Etikett pro Teil erzeugt.

Achtung: Falls das Etikettierungsfeld *zusätzlich* auch Geometriemarker enthält, wird *Survey2GIS* versuchen, diese aus dem Text der Etiketten zu entfernen. Dies kann in einigen Fällen fehlschlagen. Um sicherzustellen, dass Etikettentexte keine unnötigen Geometriemarker enthalten, sollte ein eigenes Feld für letztere verwendet werden (siehe „tag.field“ in 4.2).

## 6 Datenauswahl

Unter Umständen ist es wünschenswert, die zu prozessierenden Datensätze auf eine Untermenge zu reduzieren, etwa wenn kleinere Teilflächen eines Vermessungsgebiets separat verarbeitet werden sollen. Zu diesem Zweck ist es möglich, eines oder mehrere *Auswahlkommandos* mittels der Option „-S“ anzugeben (Erläuterungen zu deren Syntax folgen in Abschnitt 6.1):

```
-S seltype:geomtype:field:expression -S seltype:geomtype:field:expression [-S ...]
```

Die Wirkung dieser Kommandos besteht darin, dass nur diejenigen Datensätze, deren zugeordnete *Attributfeldinhalte und Geometrietypen* einem oder mehreren Auswahlkriterien (sog. *Ausdrücke*) entsprechen, in der Ausgabedatei ankommen. Die Auswahl wird unmittelbar nach der Zusammstellung der Geometrien, noch vor weitergehender Verarbeitung, wie z. B. topologischer Bereinigung (s. 9), vorgenommen. Eine Auswahl kann sich auf einen bestimmten oder alle Geometrietypen (Punkte, Linien, Polygone) beziehen.

Komplexe Geometrien (Linien und Polygone) können aus zahlreichen Messpunkten (Stützpunkten) bestehen, haben aber nur noch einen Datensatz in der Attributtabelle, nachdem sie aus den Rohdaten zusammengestellt worden sind. Eine Auswahl über Attributwerte solcher Geometrien kann daher vom Ergebnis einer Auswahl über dieselben Attribute der ursprünglichen Messpunkte abweichen. Insbesondere bleiben die Koordinatenwerte der einzelnen Stützpunkte (mit Ausnahme des ersten) nicht im Attributdatensatz eines Polygons oder einer Linie erhalten. Eine Auswahl nach einem bestimmten Wert der „Z“-Koordinate wird daher mit großer Wahrscheinlichkeit nicht das gewünschte Ergebnis liefern. In solchen Fällen bietet sich eine Auswahl mittels eines Wertebereichs oder eines regulären Ausdrucks (s. 6.2) an.

### 6.1 Syntax der Auswahlkommandos

Jede Auswahl muss in Form eines Kommandos mit einer bestimmten Syntax vorgenommen werden. Dabei sind bis zu vier Angaben, die durch einen Doppelpunkt („:“) getrennt werden, zu machen; außerdem können *Modifizierer* angegeben werden („\*“ und/oder „+“ oder „-“; nicht immer benötigte Angaben stehen hier in eckigen Klammern):

```
[*]seltype[+|-]:geomtype[:field:expression]
```

1. Die erste Angabe („seltype“ oben) ist der *Auswahltyp* (s. 6.2), welcher optional noch den Modifizierer „umkehren“ („\*“) vorangestellt, bzw. einen der Modifizierer „hinzufügen“ („+“) oder „abziehen“ („-“) angehängt bekommt (Details hierzu s. 6.3).

2. Die zweite Angabe („geomtype“) gibt den *Geometrietyp* an, auf welchen sich die Auswahl bezieht. Dieser kann als „point“ (Punkt), „raw“ (unverarb. Messpunkt), „line“ (Linie), „poly“ (Polygon) oder „all“ (sämtliche Typen) angegeben werden.
3. Die dritte Angabe enthält den *Namen des Attributfelds* (s. auch 4.3), auf das sich die Auswahl bezieht. Bei Feldnamen wird nicht zwischen Groß- und Kleinschreibung unterschieden, so dass z. B. „feature“ und „FEATURE“ sich beide auf dasselbe Feld beziehen (s. Hinweise zu Einstellung „name“ in 4.3).
4. Die vierte Angabe enthält den *Auswahlausdruck*. In den meisten Fällen wird dies einfach der Text sein, gegen den der Wert des angegebenen Attributfelds geprüft werden soll. Komplexere Ausdrücke ergeben sich bei der Verwendung von sog. „regulären Ausdrücken“ (s. Details in 6.2).

Die dritte und vierte Angabe *entfallen*, wenn als Auswahltyp „all“ (wähle alle Datensätze, unabhängig von Attributfeldwerten) angegeben wird.

Wenn der „@“-Operator zur Ersetzung von Textfeldinhalten verwendet wird (s. Erläuterungen in 4.3), dann wird der Auswahlausdruck auf den ersetzten, *nicht* den originalen, Feldinhalt, angewendet.

Um Probleme mit Leerzeichen in Auswahlausdrücken zu vermeiden, sollten Auswahlkommandos auf der *Kommandozeile* stets in Anführungszeichen eingeschlossen werden (s. die folgenden Beispiele). Dieses Problem besteht nicht bei der Verwendung über die graphische Benutzeroberfläche.

## 6.2 Auswahltypen

Die folgenden vergleichenden Auswahltypen können auf Attributfelder jeglichen Typs (numerisch oder Text) angewendet werden. Der Vergleich wird gegen den Text im „Ausdruck“-Teil des Auswahlkommandos vorgenommen.

**eq** Der Typ „equal“ („gleich“) wählt diejenigen Datensätze aus, deren Attributfeldwerte identisch mit dem angegebenen Ausdruck sind.

**neq** Der Typ „not equal“ („ungleich“) wählt diejenigen Datensätze aus, deren Attributfeldwerte *nicht* identisch mit dem angegebenen Ausdruck sind.

**lt** Der Typ „less than“ („kleiner als“) wählt diejenigen Datensätze aus, deren Attributfeldwerte „kleiner“ als der angegebene Ausdruck sind. Im Fall von numerischen Feldern ist die Bedeutung von „kleiner“ intuitiv verständlich. Bei Textfeldern funktioniert der Vergleich lexikographisch, d. h. die zu vergleichenden Werte werden in alphabetische Reihenfolge gebracht, und derjenige, der danach zuerst steht, wird als „kleiner“ betrachtet.

**gt** Der Typ „greater than“ („größer als“) wählt diejenigen Datensätze aus, deren Attributfeldwerte „größer“ als der angegebene Ausdruck sind. Im Fall von numerischen Feldern ist die Bedeutung von „größer“ intuitiv verständlich. Bei Textfeldern funktioniert der Vergleich lexikographisch, d. h. die zu vergleichenden Werte werden in alphabetische Reihenfolge gebracht, und derjenige, der danach zuletzt steht, wird als „größer“ betrachtet.

**lte** Der Typ „less than or equal“ („kleiner oder gleich“) funktioniert prinzipiell wie „kleiner als“ (s. o.), wählt aber zusätzlich diejenigen Datensätze aus, deren Feldwerte „gleich“ (s. o.) dem angegebenen Ausdruck sind.

**gte** Der Typ „greater than or equal“ („größer oder gleich“) funktioniert prinzipiell wie „größer als“ (s. o.), wählt aber zusätzlich diejenigen Datensätze aus, deren Feldwerte „gleich“ (s. o.) dem angegebenen Ausdruck sind.

Beispiele (beachte die Verwendung von Anführungszeichen, um Probleme mit Leerzeichen auf der Kommandozeile zu vermeiden):

```
-S "eq:poly:objname:boundary"  
-S "eq:poly:objname:boundary type A"  
-S "gte:all:elevation:12.00"
```

Ein weiterer Auswahltyp kann nur auf numerische Felder angewendet werden:

**range** Der Typ „range“ („Wertebereich“) wählt diejenigen Datensätze aus, deren numerische Feldwerte sich innerhalb des als „min;max“ angegebenen Intervals (Intervallgrenzen inklusive) befinden, z. B.:

```
-S "range:point:elevation:100.0;120.0"
```

Ebenso existieren Auswahltypen, die sich nur auf Felder vom Typ Text anwenden lassen:

**sub** Der Typ „Teiltext“ („Substring“) wählt diejenigen Datensätze aus, deren Textfeldwerte die Zeichenfolge im Ausdruck als Untermenge enthalten. Das Beispiel unten wählt „boundary“, „Boundary“, „boundaries“, „closed boundary“, etc. aus:

```
-S "sub:poly:objname:oundar"
```

**regexp** Dieser Typ wählt Datensätze anhand eines *regulären Ausdrucks* ([https://de.wikipedia.org/wiki/Regul%C3%A4rer\\_Ausdruck](https://de.wikipedia.org/wiki/Regul%C3%A4rer_Ausdruck)) aus. Folgende Suchmuster werden in regulären Ausdrücken unterstützt (s. auch <https://github.com/cesanta/slre>):

<code>^</code>	Match beginning of field content.
<code>\$</code>	Match end of field content.
<code>()</code>	Grouping and substring capturing
<code>\s</code>	Match whitespace.
<code>\S</code>	Match non-whitespace.
<code>\d</code>	Match decimal digit.
<code>\n</code>	Match new line character.

<code>\r</code>	Match line feed character.
<code>\f</code>	Match form feed character.
<code>\v</code>	Match vertical tab character.
<code>\t</code>	Match horizontal tab character.
<code>\b</code>	Match backspace character.
<code>+</code>	Match one or more times (greedy).
<code>+</code> <code>?</code>	Match one or more times (non-greedy).
<code>*</code>	Match zero or more times (greedy).
<code>*</code> <code>?</code>	Match zero or more times (non-greedy).
<code>?</code>	Match zero or once (non-greedy).
<code>x y</code>	Match x or y (alternation operator).
<code>\meta</code>	Match one of the meta character: <code>^\$().[*+? \]</code> .
<code>\xHH</code>	Match byte with hex value 0xHH, e.g. <code>\x4a</code> .
<code>[...]</code>	Match any character from set. Ranges like <code>[a-z]</code> are supported.
<code>[^...]</code>	Match any character but ones from set.

Beispiel:

```
-S "regexp:all:objname:type 1|type 2"
```

Dies wählt alle Datensätze, die „type 1“ oder „type 2“ in „objname“ führen.

Reguläre Ausdrücke in *survey2gis* unterstützen *keine* Unicode-Zeichenkodierung.

Schließlich existiert noch ein besonderer Auswahltyp:

**all** Der Typ „all“ („alle“) wählt alle Datensätze aus, unabhängig davon, welche Werte sich in ihren Attributfeldern befinden. Sein Zweck besteht v. a. in der Auswahl von Datensätzen nur nach Geometrietyp. In diesem Fall sind weder die „Feld“- noch „Ausdruck“-Teile des Selektionskommandos anzugeben. Um bspw. alle Linien („line“) zu wählen:

```
-S "all:line"
```

### 6.3 Auswahlmodifizierer und -ketten

Die Wirkung jedes Auswahlkommandos (hinsichtlich Attributfeldern und Geometrietypen) kann *umgekehrt* werden, indem der Angabe des Selektionstyps ein „\*“ vorangestellt wird. Bspw. wählt das folgende Kommando diejenigen Datensätze, deren Attributfeldwerte die Zeichenfolge „boundary“ *nicht* im Feld „objname“ führen:

```
-S "*eq:all:objname:boundary"
```

Zu jedem Selektionstyp existiert außerdem eine alternative Schreibweise in Großbuchstaben:

EQ,LT,GT,LTE,GTE,SUB,REGEXP,RANGE,ALL

Wird eine dieser Alternativformen auf ein Feld vom Typ „Text“ angewendet, dann wird der Auswahl Ausdruck ohne Berücksichtigung von Groß- und Kleinschreibung verarbeitet. Der Effekt ist derselbe, als würde man den Feldwert und den Inhalt des Ausdrucks vor der Verarbeitung vollständig in Großbuchstaben umwandeln, z. B. würden ...

```
-S "EQ:poly:objname:Boundary"  
-S "EQ:poly:objname:boundary"  
-S "EQ:poly:objname:BOUNDARY"
```

... gleichermaßen auf die Feldwerte „Boundary“, „BoUNDary“, etc. passen.

Bei numerischen Feldtypen besteht kein Unterschied zwischen den groß- und kleingeschriebenen Typennamen, ebenso verhält es sich bei „ALL“.

Mehrere Auswahlkommandos können als *Kette* ausgeführt werden, um das Auswahlergebnis schrittweise zu verfeinern. Dies erreicht man durch Voranstellen von „+“ (zum bisherigen Auswahlergebnis hinzufügen) bzw. „-“ (vom bisherigen Auswahlergebnis abziehen).

Dabei gelten, zusammengefasst, folgende Regeln:

1. Anfangs sind per Vorgabe alle Datensätze Teil der aktuellen Auswahl (dies ist das Äquivalent eines Auswahlkommandos der Form „all:all“).
2. Ein oder mehrere Auswahlkommandos können angegeben werden. Sie werden in der Reihenfolge verarbeitet, in der sie auf der Kommandozeile stehen.
3. Der Modifizierer „+“ fügt das Result eines Auswahlkommandos zur bestehenden Auswahl hinzu.
4. Der Modifizierer „-“ zieht das Result eines Auswahlkommandos von der bestehenden Auswahl ab.
5. Wenn keiner dieser Modifizierer angegeben wird, dann wird die Auswahl auf das Resultat des Auswahlkommandos gesetzt.
6. Nachdem alle Auswahlkommandos verarbeitet wurden, enthält die Ausgabedatei diejenigen Datensätze, welche dann noch Teil der Auswahl sind.

Die Auswahlkommandos werden in der Reihenfolge verarbeitet (und Datensätze dem aktuellen Resultat hinzugefügt oder von diesem abgezogen), in der sie vom Benutzer angegeben werden. Das Beispiel unten (alle drei „-S“-Optionen müssen in einer Kommandozeile stehen) wählt zunächst alle Datensätze, die im Feld „objname“ („Objektnamen“) die Zeichenfolge „oundar“ enthalten, wie etwa „Boundary“. Das zweite Kommando entfernt dann diejenigen Datensätze vom Ergebnis, die in besagtem Feld „closed boundary“

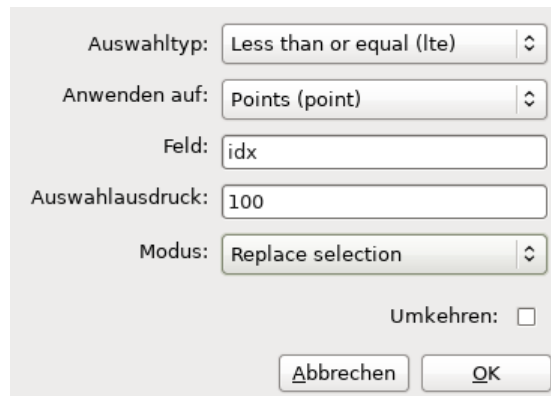
enthalten. Das dritte Kommando sorgt schließlich dafür, dass im Endresultat nur noch diejenigen Datensätze enthalten sind, bei denen der Inhalt des Felds „objtype“ („Objekttyp“) nicht auf „breakline“ gesetzt ist. Alle drei Kommandos beachten die Groß- und Kleinschreibung der Feldinhalte nicht:

```
-S "SUB:poly:objname:oundar"  
-S "EQ-poly:objname:closed boundary"  
-S "EQ-poly:objtype:breakline"
```

Details zur Wirkung jedes Auswahlkommandos auf die Datensätze sowie evtl. Warnungen und Fehlermeldungen werden während der Prozessierung in Form von Meldungen ausgegeben.

## 6.4 Selektionskommandos mit der graphischen Benutzeroberfläche

Auswahlkommandos lassen sich auch über die graphische Benutzeroberfläche (GUI: s. 3) zusammenstellen. Dazu betätigt man den Knopf „Selection“. Dies bringt eine Liste der momentan definierten Auswahlkommandos hervor. Mittels „Add“ oder „Edit“ lassen sich neue Kommandos erstellen bzw. vorhandene verändern. In beiden Fällen erscheint ein weiteres Fenster, wie unten abgebildet:



Kommandos lassen sich definieren, indem alle benötigten Felder des Formulars ausgefüllt werden. Im Falle des Auswahltyps „All“ werden Eingaben in den Feldern „Field:“ und „Expression:“ verworfen, sobald die Definition des Kommandos mit „OK“ abgeschlossen wird. Die Bestätigung per „OK“ löst auch eine einfache Syntaxprüfung des definierten Kommandos aus. Diese Prüfung bezieht sich jedoch nur auf die formale Korrektheit der Eingaben. Weder wird geprüft, ob das Resultat der Auswahl leer wäre, noch ob ein angegebenes Feld existiert und den richtigen Typ besitzt. Da dies von den konkreten Daten abhängt, kann es erst während der Prozessierung geprüft werden. Aus diesem Grund sollten die Statusmeldungen von *Survey2GIS* gut beobachtet werden, um die Korrektheit der Auswahlresultate sicherzustellen.

Die Auswahlkommandos werden in der Reihenfolge (von oben nach unten) verarbeitet, wie sie in der Liste erscheinen.

Beim Speichern der GUI-Einstellungen in eine „.s2g“-Einstellungsdatei werden alle momentan definierten Auswahlkommandos ebenfalls gesichert.

Technischer Hinweis: Das Zeichen „|“ („Pipe“) wird in regulären Ausdrücken dazu verwendet Alternativen auszudrücken (logisches „oder“). Dasselbe Zeichen wird allerdings auch in *survey2gis* Einstellungsdateien („s2g“) als Listentrenner verwendet. Um Konflikte beim Speichern/Laden von regulären Ausdrücken in/aus Einstellungsdateien zu vermeiden und gleichzeitig mit älteren Programmversionen kompatibel zu bleiben, wird „|“ in regulären Ausdrücken als ASCII-Zeichen Nr. 30 (RS, „record separator“) gespeichert und automatisch konvertiert. Dies muss berücksichtigt werden, falls mit anderen Programmen Einstellungsdateien von *Survey2GIS* bearbeitet werden soll.



## 7 Orientierungsmodi

Ab Version 1.5.0 unterstützt *Survey2GIS* unterschiedliche Modi um die geographischen Achsen des Koordinatensystems der Ausgabedaten neu zu orientieren. Dies macht es möglich, „Vertikaldaten“, d. h. Vermessungsdaten mit dominanter Z-Komponente (Bodenaufschlüsse, „Profile“ in der Archäologie, etc.) für eine 2D-GIS-Umgebung aufzubereiten.

Die Verarbeitung von 3D-Daten ist komplex und umfasst weit mehr als das „Durchreichen“ von zusätzlichen Z-Koordinaten. GIS-Konzepte und Funktionen wie Topologie und Georeferenzierung lassen sich nicht geradlinig in die dritte Dimension erweitern.

Es ist daher nicht trivial, Daten mit starker Z-Komponente in einen Arbeitsfluss im 2D-GIS zu integrieren. *Survey2GIS* bietet einen Ausweg, indem es die Eingabedaten auf ein neues, *synthetisches* Koordinatensystem abbildet, dessen Achsen derart neu orientiert sind, dass sie sich in ein 2D-GIS laden lassen und sich dort wie reguläre X-Y-Daten darstellen. Eine solche Transformation von 3D nach 2D beinhaltet notwendigerweise einen Informationsverlust.

Per Voreinstellung behält *Survey2GIS* das originale (Welt-)Koordinatensystem der Eingabedaten bei, d. h. es wird keine Transformation vorgenommen. Dies entspricht dem Modus „Weltorientierung“, bzw. dem Optionswert „world“ für die Option „--orientation“ (s. 7.1).

Falls gewünscht, wird eine Neuorientierung der Achsen und Transformation der Koordinaten in ein synthetisches Benutzerkoordinatensystem vorgenommen. Dies geschieht unmittelbar nach dem Lesen der Eingabedaten und vor einer weiteren Verarbeitung der Geometrien, bspw. zum Zwecke der topologischen Bereinigung (s. 9). Dies erlaubt es *Survey2GIS*, die transformierten Daten wie gehabt weiter zu verarbeiten, als handelte es sich um reguläre 2D-Daten. Die Koordinaten von Etikettierungspunkten (soweit vorhanden: s. 5) werden ebenfalls transformiert.

Die Programmoption „--orientation=“ (s. 2) wird verwendet um die Daten wie unten beschrieben neu zu orientieren. Evtl. sind nicht alle Modi für alle Typen von Eingabedaten verfügbar.

Achtung: Daten, die in einem anderen Orientierungsmodus als „world“ ausgegeben werden, unterliegen Transformationen die i. d. R. *nicht reversibel* sind. Daher sollten Rohdaten entsprechend archiviert und neu orientierte Daten nicht als primäres Ausgabeprodukt betrachtet werden.

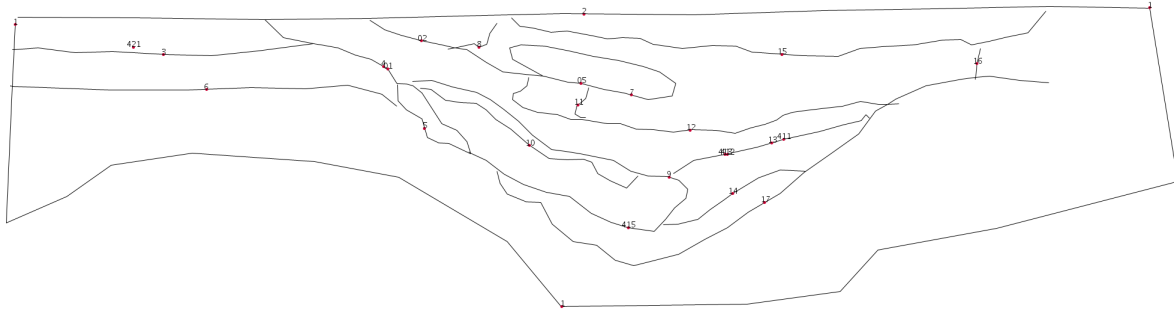
### 7.1 Weltorientierung (Vorgabe)

Der Orientierungsmodus „Weltorientierung“ ist der Vorgabemodus für *Survey2GIS*. Er modifiziert die Eingabedaten in keiner Weise. Der zugehörige Optionswert „world“ existiert lediglich um Konsistenz bei der Benutzung des Programms zu wahren.

### 7.2 Lokale X-Z-Orientierung („Profile“)

Die „lokale X-Z-Orientierung“ (Optionswert „localxz“) bietet eine einfache Möglichkeit 3D-Vermessungsdaten in 2D-Schnittebenen („Profile“) abzubilden, die sich zur Visualisierung im 2D-GIS eignen. Hierbei wird ein

neues, synthetisches Koordinatensystem für die Ausgabedaten produziert. Dieser Modus eignet sich nicht für große Areale mit vielen disjunkten Geometrien. Sein Einsatzzweck ist die 2D-Abbildung von einzelnen „Profilen“ (Bodenaufschlüssen, „Schnitten“, etc.):



Die neu orientierten Ausgabedaten stehen in einem synthetischen Koordinatensystem mit vertauschten Z- und Y-Achsen sowie neu berechneten X-Koordinaten. Da die transformierten X-Koordinaten stets by „0“ beginnen und die Z-Daten erhalten bleiben, lassen sich unterschiedliche Pofilvermessungen gut miteinander vergleichen (sofern das verwendete GIS mehr als eine Datensicht gleichzeitig erlaubt).

Diese Transformationsmethode ist nichtparametrisch und vollautomatisch (insbesondere muss keine Projektionsebene festgelegt werden). Die transformierten Daten haben ansonsten die folgenden Eigenschaften:

- Die neuen Y-Koordinaten sind mit den ursprünglichen Z-Daten identisch.
- Die Koordinaten entlang der neuen, synthetischen X-Achse entsprechen nicht länger irgendwelchen Originaldaten.
- Die kleinste X-Koordinate in den Eingabedaten wird zum Ursprung ( $X = 0$ ) der neuen, synthetischen X-Achse.
- Die größte X-Koordinate auf der synthetischen X-Achse entspricht dem maximalen Unterschied zwischen den X-Koordinaten der Eingabedaten.
- Alle originalen Y-Koordinaten werden verworfen; benutzerdefinierte Offsets auf allen Achsen werden verworfen.

Diese Transformationsmethode hat die folgenden Einschränkungen:

- Eingabedaten in Längen- und Breitengraden können nicht verarbeitet werden.
- Die Eingabedaten sollten nicht mehr als die Ausdehnung eines Profils umfassen.
- Alle Eingabedaten müssen Punktmessungen repräsentieren, die von derselben Stelle aus eingemessen wurden. Dies bedeutet zugleich, dass komplexere 3D-Strukturen mit stärkerer Konkavität („Rücksprünge“) zu schlechteren Ergebnissen führen werden.

- Daten mit starker Varianz der Koordinaten entlang der Hauptachse des Profils (in der  $X$ - $Y$ -Ebene betrachtet) führen zu entsprechend starker Verzerrung der transformierten Darstellung.
- Daten mit lokaler  $X$ - $Z$ -Orientierung können im Anschluss nicht in ein anderes KBS reprojiziert werden (s. 8).
- Daten die eine deutlich stärkere Ausdehnung entlang der  $Z$ -Achse als innerhalb der  $X$ - $Y$ -Ebene haben, werden mit höherer Wahrscheinlichkeit zu einem seitenverkehrt ausgerichteten Resultat führen (s. Beschreibung weiter unten).
- Es ist nicht möglich, Daten aus mehr als einer Eingabedatei in einem Programmdurchlauf zu transformieren.

Das neu orientierte Ergebnis wird umso weniger verzerrt, je perfekter die Messungen bei Betrachtung der Daten in der  $X$ - $Y$ -Ebene auf einer Linie liegen. Je größer die Abweichung von diesem Idealfall, desto stärker die Verzerrung im Resultat. In ähnlicher Weise sind transformierte lokale Profildaten besser zu verstehen, wenn die originalen  $Z$ -Koordinaten sich „einigermaßen orthogonal“ zur  $X$ - $Y$ -Ebene verteilen (denn die neue  $Y$ -Achse behält die geographische Ausrichtung der ursprünglichen  $Z$ -Achse bei).

Die Transformation in lokale  $X$ - $Z$ -Orientierung funktioniert folgendermaßen:

1. Wähle einen Referenzpunkt  $RP$  und finde die Orientierung des Profils (s. u.).
2. Für alle Messpunkte ( $p_{i=1,2,\dots,n}$ ) in den Eingabedaten  $P$ :
  - (a) Berechne die  $X$ -Koordinate der transformierten Version von  $p_i$  als geradlinige Distanz zwischen der originalen  $X$ / $Y$ -Position von  $p_i$  und  $RP$ .
  - (b) Setze  $Y$  für den transformierten Punkt  $p_i$  auf den Wert der originalen  $Z$ -Koordinate.
  - (c) Setze neue  $Z$ -Koordinate auf „0“.

Die Voraussetzung für brauchbare Resultate sind zwei Parameter, welche automatisch aus den Daten abgeleitet werden:

1. Die allgemeine  $W$ - $E$ -Richtung des Profils, d. h. die Entscheidung, auf welcher Seite im synthetischen Koordinatensystem sich der linke Rand befinden wird. Dieser soll mit der linken Seite des Profils während der Vermessung übereinstimmen. Wenn dies falsch berechnet wird, erscheint das Profil im Resultat „gespiegelt“.
2. Der „extremste“ Messpunkt, d. h. der Punkt mit der kleinsten  $X$ -Koordinate im synthetischen  $X$ - $Z$ -Koordinatensystem. Wird der falsche Punkt identifiziert, dann erscheint der linke Rand des Profils verzerrt.

Der erste Parameter, „Profilrichtung“, wird automatisch bestimmt, indem die gewichtete Summe der Differenzen zwischen den *X*-Koordinaten eines *Referenzpunkts* und denen aller anderen Messpunkte betrachtet wird. Alle *X*-Differenzen werden *invers* mittels der Differenzen in den *Z*-Koordinaten zwischen ihnen und dem Referenzpunkt *gewichtet*. Dies hat die Wirkung, dass Messpunkte, die auf der *Z*-Achse weiter vom Referenzpunkt entfernt liegen, das gewichtete Ergebnis weniger stark beeinflussen. So wird das Praxisproblem kompensiert, dass der tatsächliche „linke“ Rand des Profils signifikant gegenüber dem Referenzpunkt „zurückspringen“ kann. Ist die gewichtete Summe der *X*-Unterschiede positiv, dann wird die „Richtung“ des Profils als „ostwärts“ bestimmt, andernfalls als „westwärts“. Dabei handelt es sich nicht um geographisch akkurate Bestimmungen der Richtung, sondern lediglich eine Entscheidung, auf welcher Seite des transformierten Systems der Ursprung der Profildaten liegen wird.

Im (eher theoretischen) Extremfall, bei dem sich die gewichteten *X*-Differenzen exakt zu „0“ summieren (dies könnte z. B. theoretisch der Fall sein, wenn alle Messpunkte exakt auf einer Parallelen zur geographischen *S-N*-Achse lägen), kann die Richtung nicht berechnet werden und wird als „ostwärts“ angenommen.

In der Praxis hängt alles davon ab, dass ein geeigneter Referenzpunkt vorhanden ist. *Survey2GIS* geht davon aus, dass es sich dabei um den *ersten Punkt* in den Eingabedaten handelt. Dies führt am ehesten dann zu guten Ergebnissen, wenn die folgenden Regeln bei der Vermessungsarbeit berücksichtigt werden:

- Messungen sollten an oder in direkter Nähe der linken oberen Ecke (vom Standpunkt des Vermessungsgeräts aus gesehen) des Profils starten.
- Weitere Messungen sollten dann annähernd nach rechts und unten erfolgen, mit der letzten Messung nahe der rechten unteren Ecke.

Die obigen Voraussetzungen müssen (und können in der Praxis verm. auch) nicht exakt eingehalten werden. Solange eine ausreichende Zahl von Messpunkten vorliegt, kann das Programm Abweichungen kompensieren. *Survey2GIS* erstattet detailliert Bericht über die Koordinaten des Referenzpunktes, die ermittelte Profilrichtung und sonstige, wesentliche Parameter der Transformation.

## 8 Bezugssysteme und Reprojektionen

Seit Version 1.5.0 unterstützt *Survey2GIS* die Angabe von Koordinatenbezugssystemen (KBS) für Ein- und Ausgabedaten sowie die Reprojektion zwischen verschiedenen Systemen. Dies zog signifikante Änderungen an der Benutzerschnittstelle und der Programmlogik nach sich. Von besonderer Wichtigkeit ist, dass Eingabedaten in Längen- und Breitengraden seitdem explizit als solche behandelt werden, was das Verhalten des Programms bei topologischer Bereinigung (s. 9) und einigen Ausgabeformaten (s. 10) im Vergleich zu älteren Versionen verändert.

*Survey2GIS* ermöglicht es dem Benutzer, Koordinatenbezugssysteme (KBS) und verwandte Einstellungen entweder nur für die Eingabedaten oder sowohl für Ein- als auch Ausgabedaten anzugeben. Die Angabe des *Quell-KBS* für die Eingabedaten hilft die Korrektheit der Datenprozessierung sicherzustellen. Wird außerdem das *Ziel-KBS* für die Ausgabedaten angegeben, dann reprojiziert *Survey2GIS* alle Koordinaten automatisch vom Quell- ins Ziel-KBS.

KBS und verwandte Parameter zur Datumstransformation (für höhere Genauigkeit bei bestimmten Reprojektionen: s. 8.2.1) können über eine handvoll von Programmoptionen eingestellt werden, die alle das Präfix „proj-“ im Optionsnamen tragen (es gibt keine Einbuchstabenkürzel für diese Optionen):

```
--proj-in  
--proj-out  
--proj-dx  
--proj-dy  
--proj-dz  
--proj-rx  
--proj-ry  
--proj-rz  
--proj-ds  
--proj-grid
```

Die ersten beiden Optionen („--proj-in“ und „--proj-out“) dienen dazu, Quell- und Ziel-KBS anzugeben, die übrigen werden im Kontext von geodätischen Datumstransformationen verwendet (s. 8.2.1). Weitere Details lassen sich den folgenden Ausführungen entnehmen (s. auch die Kurzbeschreibungen in 2).

Eine Reprojektion wird erst durchgeführt, *nachdem* die sonstige Geometrieverarbeitung abgeschlossen ist. Dadurch ist es bspw. möglich, UTM-Eingabedaten topologisch zu bereinigen und anschließend in ein System aus Längen- und Breitengraden zu reprojizieren.

Zur Berechnung von Koordinatenreprojektionen verwendet *Survey2GIS* eine eingebaute Kopie der Open-Source-Software PROJ.4. Demzufolge werden intern alle KBS-Angaben in das PROJ.4-Format konvertiert, und die Möglichkeiten der Reprojektion entsprechen exakt denen der eingebauten Version von PROJ.4. Weitere Details s. 8.2.

Bestehen die Eingabedaten aus Längen- und Breitengraden, dann wird angenommen, dass diese im Format „Dezimalgrad“ (ggfs. mit negativem Vorzeichen) vorliegen ([https://en.wikipedia.org/wiki/Decimal\\_degrees](https://en.wikipedia.org/wiki/Decimal_degrees)). *Survey2GIS* bietet keine Unterstützung für die Schreibung als „Grad, Minuten, Sekunden“, „E/W“-Präfixe oder ähnliche Notationen.

## 8.1 Ein- und Ausgabesysteme setzen

Die Benutzerin kann das (gegebene) KBS der Eingabedaten (Quell-KBS) sowie das (gewünschte) KBS der Ausgabedaten (Ziel-KBS) setzen. Hierfür existieren entsprechend die Optionen „--proj-in“ und „--proj-out“. Für sämtliche Eingabedaten kann nur ein KBS gesetzt werden. Es ist somit nicht möglich, Eingabedaten mit unterschiedlichen KBS gleichzeitig zu verarbeiten.

Die Angabe *lediglich* des Quell-KBS ermöglicht es *Survey2GIS*, die Datenverarbeitung wo benötigt anzupassen. Dies ist von besonderer Bedeutung bei der Verarbeitung von Eingabedaten in Längen- und Breitengraden, da letztere eine andere mathematische Behandlung erfordern als planimetrische X-Y-Daten. In Situationen, in denen keine (korrekte) Verarbeitung solcher Daten erfolgen kann, gibt *Survey2GIS* eine Warnung aus oder bricht die Verarbeitung komplett ab.

Werden Quell- und Ziel-KBS angegeben und handelt es sich dabei um *unterschiedliche* Systeme, dann reprojiziert *Survey2GIS* die Daten vom Quell- (es sei denn, es handelt sich um ein lokales System: s. u.) ins Ziel-KBS. Eine Reprojektion bringt normalerweise einen Genauigkeitsverlust mit sich. Um letzteren möglichst klein zu halten, können weitere Optionen zur Datumstransformation (s. 8.2.1) gesetzt werden.

Die Angabe lediglich des Ziel-KBS führt zu einer Fehlermeldung.

KBS können per EPSG-Nummer, PROJ.4-Definition oder (in einigen wenigen Fällen) einfach per Kürzel gesetzt werden.

### 8.1.1 Interne KBS-Kürzel

Für eine kleine Zahl von KBS kennt *Survey2GIS* einfache Kürzel (Groß- und Kleinschreibung spielt keine Rolle):

**local** Das Setzen des Eingabe-KBS auf „local“ teilt *Survey2GIS* mit, dass die Vermessungsdaten in einem lokalen, kartesischen (planimetrischen) System mit eigenem Ursprung und metrischen Entfernungs-einheiten (Metern) vorliegen. Derartige Daten können nicht in ein anderes KBS reprojiziert werden. Das Kürzel „local“ existiert hauptsächlich um Konsistenz bei der Datenverarbeitung zu erreichen. Ist das Eingabe-KBS „local“ und das Ausgabe-KBS ein anderes, so führt dies zu einer Fehlermeldung.

**wgs84** Die Angabe „wgs84“ ist äquivalent zu „EPSG:4326“ (d. h. Längen- und Breitengrade mit Datum „WGS 84“). Achtung: *Survey2GIS* verarbeitet nur Daten im Format *Dezimalgrad* (s. 8.3).

**web** Dieses Kürzel steht für das KBS, welches von *Google Earth/Maps* und anderen Internet-Kartendiensten verwendet wird. Bitte beachten Sie die Hinweise in 8.1.4. Der entsprechende EPSG-Eintrag ist „EPSG:3857“.

**utmXXN/S** Eine UTM-Zone (mit WSG-84-Datum) wird mittels des Kürzels „utm“ und angehängter Zonennummer, sowie „N“ (Ord) oder „S“ (üd)-Suffix angegeben. Zum Beispiel steht „utm32s“ für „UTM mit WGS-84-Datum, Zone 32, Nord“. Die entsprechenden EPSG-Einträge haben die Nummern „32601“ bis einschließlich „32760“.

**dhdnX** Dieses Kürzel steht für das Gauß-Krüger-System auf dem Gebiet der Bundesrepublik Deutschland. Für „X“ wird die Zonennummer „2“ bis „5“ eingesetzt. Die äquivalenten EPSG-Nummern sind „31466“ bis einschließlich „31469“.

**osgb** Dieses Kürzel steht für das traditionelle (immer noch in Benutzung befindliche) System des British Ordnance Survey (OSGB). Sein Äquivalent ist „EPSG:27700“.

### 8.1.2 Angabe von KBS per EPSG-Nummer

Sofern für das zu verwendene KBS kein Kürzel zur Verfügung steht (s. 8.1.1), besteht die zweiteinfachste Möglichkeit zur Angabe eines KBS in der Verwendung einer EPSG-Nummer (alle Kürzel werden von *Survey2GIS* ohnehin intern in ihre äquivalenten EPSG-Einträge konvertiert).

Die vollständige EPSG-Datenbank kann online hier eingesehen werden: <http://www.epsg-registry.org/>.

Die Eingabe eines KBS per EPSG-Nummer geschieht über das Präfix „epsg:“, gefolgt von der ID des Eintrags in der EPSG-Datenbank (Groß- und Kleinschreibung spielt keine Rolle), z. B.:

```
--proj-in=EPSG:27700
--proj-out=epsg:4326
```

Beachten Sie, dass *Survey2GIS* nur eine Untermenge der EPSG-Definitionen verwenden kann, nämlich dieselben, welche auch vom eingebauten PROJ.4 unterstützt werden (s. 8.2). PROJ.4 wiederum führt eine, nicht unbedingt perfekte, Übersetzung von EPSG in sein eigenes Format durch, was in seltenen Fällen zu einem signifikanten Informationsverlust führen kann (s. <http://proj4.org/faq.html#how-do-i-use-epsg-coordinate-system-codes-with-proj>). Sollte dies zu Problemen führen, kann alternativ eine vollständige PROJ.4-Definition als KBS eingegeben werden (s. 8.1.3).

### 8.1.3 Angabe von KBS als PROJ.4-Definitionen

In manchen Fällen kann weder die Verwendung eines Kürzels (s. 8.1.1) noch eines EPSG-Eintrags (s. 8.1.2) ausreichend sein um ein KBS vollständig zu beschreiben und akzeptable Reprojektionsergebnisse zu erhalten. In solchen Fällen können alternativ vollständige PROJ.4-Definitionen verwendet werden.

Eine PROJ.4-Definition besteht aus mehreren *Elementen*, die jeweils mit einem vorangestellten „+“ eingeleitet werden und mit einem Leerzeichen vom Folgeelement getrennt werden. Groß- und Kleinschreibung von Elementnamen spielt keine Rolle, da *Survey2GIS* diese ohnehin intern in Kleinschreibung umsetzt. Rechts

des Elementnamens folgt der Elementwert, vom Namen durch ein Gleichheitszeichen („=“) *ohne weitere Leerzeichen* getrennt.

So lautet die vollständige PROJ.4-Definition von „EPSG:4326“ (Längen- und Breitengrade mit WGS-84-Datum):

```
+proj=longlat +ellps=WGS84 +datum=WGS84
```

Und „EPSG:32601“ (UTM-Zone 1, Nord) schreibt sich:

```
+proj=utm +zone=1 +ellps=WGS84 +datum=WGS84 +units=m
```

Bitte konsultieren Sie die Online-Hilfe von PROJ.4 auf <http://proj4.org/> für alle Details, und beachten Sie auch die Informationen unter 8.2.

Außerdem können folgende Quellen dabei helfen, das Format gültiger PROJ.4-Definitionen zu verstehen:

- *Survey2GIS* gibt die vollständigen PROJ.4-Definitionen aller verwendeten KBS als Teil seiner Programmausgabe (Log) aus.
- Die Internet-Seiten auf <http://www.spatialreference.org/> können verwendet werden um zwischen verschiedenen KBS-Formaten, darunter auch EPSG und PROJ.4, zu konvertieren.

Ein zusätzliches „+no\_defs“-Element wird von *Survey2GIS* stets automatisch der PROJ.4-Definition hinzugefügt. Dieses verhindert, dass PROJ.4 (potentiell unerwünschte) eigene Vorgaben anwendet.

Die Parameter für genauere Datumstransformationen (s. 8.2.1) bzw. eine oder mehrere Grid-Dateien (s. 8.2.2) können über die Elemente „+towgs84“ bzw. „+nadgrid“ in die PROJ.4-Definition eingefügt werden. Allerdings werden diese Elemente automatisch verworfen und durch die entsprechenden Optionswerte von *Survey2GIS* ersetzt, sofern letztere angegeben worden sind.

Achtung: Bei den Elementen in PROJ.4-Definitionen wird zwischen Groß- und Kleinschreibung unterschieden. Bspw. führt die Angabe von „+ellps=wgs84“ zu einer Fehlermeldung (die korrekte Form wäre „+ellps=WGS84“).

Achtung: PROJ.4 verarbeitet keine internationalen Zahlenformate! Daher ist es nicht möglich, in Elementwerten von PROJ.4-Definitionen einen anderen Dezimaltrenner als den einfachen Punkt („.“) zu verwenden!

#### 8.1.4 Web Mercator als KBS

Populäre Internet-Dienste zur Navigation und Kartenvisualisierung, wie etwa Google Earth/Maps, OpenStreetMap, etc. verwenden eine perfekte Kugel als Erdmodell (d. h. die Länge des Äquators entspricht exakt der jedes Meridians). Dies ist keine akzeptable geodätische Annahme, aber sie hilft, die Performanz von 3D-Anzeigen zu erhöhen. Dieses für die 3D-Darstellung optimiert Modell ist mittlerweile als „Web Mercator“



bekannt (s. [https://en.wikipedia.org/wiki/Web\\_Mercator](https://en.wikipedia.org/wiki/Web_Mercator)). Das Problem besteht hier darin, dass oben genannte Dienste *vorgeben*, Daten mit (geodätisch weit korrekterem) WGS-84-Datum zu verwenden. Paradoxerweise kommt es dadurch, wenn eine geodätisch korrekte Reprojektion mit Datumstranformation über WGS 84 durchgeführt wird (s. 8.2.1) zu falschen Y-Koordinaten im Ergebnis. Dies liegt daran, dass PROJ.4 automatisch den Unterschied in den Längen der N-S-Radien der beiden Erdmodelle kompensiert. Laut <http://spatialreference.org/ref/sr-org/6864/> kann der Fehler durchaus signifikant sein:

“Relative to an ellipsoidal development errors of up to 800 meters in position and 0.7 percent in scale may arise.”

Der entsprechende EPSG-Eintrag besagt:

“Relative to WGS 84 / World Mercator (CRS code 3395) errors of 0.7 percent in scale and differences in northing of up to 43km in the map (equivalent to 21km on the ground) may arise.”

Um diesen Effekt zu umgehen und die gängige Annahme, dass es sich z. B. bei aus Google Earth abgegriffenen Koordinaten um WGS-84-Daten handelt, aufrecht zu erhalten, führt *Survey2GIS* eine zweistufige Reprojektion durch, wenn Web-Mercator-Daten vorliegen:

1. Die Web-Mercator-Koordinaten werden *ohne* Reprojektion in WGS-84-Daten umdeklariert.
2. Die neuen „WGS-84-Koordinaten“ werden dann „ordentlich“ in das gewünschte Ziel-KBS reprojiziert (es sei denn, letzteres ist ohnehin „WGS 84“, in welchem Fall keine weitere Transformation notwendig ist).

Es muss jedoch angemerkt werden, dass dieses Verfahren zur Umgehung des Problems streng genommen kein geodätisch korrektes ist und dass das Web-Mercator-Modell allgemein nicht für Anwendungen geeignet sind, die auf geodätisch akkurate Messungen angewiesen sind.

Der obige Mechanismus *greift nur dann*, wenn entweder „EPSG:3857“ oder das Kürzel „web“ für die Angabe des KBS verwendet wird. Wird eine PROJ.4-Definition verwendet, so muss diese entsprechend ergänzt werden (s. <http://proj4.org/faq.html#changing-ellipsoid-why-can-t-i-convert-from-wgs84-to-google-earth>).

Die Angabe von Parametern zur Datumstranformation oder einer Grid-Datei führt unweigerlich zu einer Fehlermeldung und einem Programmabbruch, wenn „Web Mercator“ das Quell- oder Ziel-KBS ist!

## 8.2 Reprojektion mittels PROJ.4

Wichtig: Diese Version von *Survey2GIS* verwendet PROJ.4 in Version 4. Zum Zeitpunkt der Erstellung dieses Handbuchs liefen die letzten Vorbereitungen zur Freigabe von PROJ.4 Version 5, mit zahlreichen Änderungen bzgl. der Benutzung und des Anwendungsbereichs von PROJ.4. Bei der Konsultation von Online-Quellen zu PROJ.4 muss daher beachtet werden, dass diese sich auf die von *Survey2GIS* verwendete Version beziehen.

Werden sowohl Quell-, als auch Ziel-KBS in gültiger Form angegeben (s. 8.1) und ist keines der beiden vom Typ „local“ (s. 8.1.1), dann reprojiziert *Survey2GIS* die Koordinaten vom Quell- in das Ziel-KBS. Die Reprojektion schließt alle (Stütz-)Punktkoordinaten und (sofern vorhanden) Etikettierungspunkte (s. 5) ein. *Survey2GIS* greift zur Berechnung der Reprojektion auf eine eingebaute Kopie von PROJ.4 (<http://proj4.org/>) zurück. Daher sind seine Fähigkeiten bzgl. Reprojektion, Datumstransformation, etc. identisch mit denen von PROJ.4. Im Wesentlichen bedeutet dies, dass Reprojektionen nur für 2D geographische (Längen- und Breitengrade) und 2D planimetrische (X/Y) KBS berechnet werden können.

Die genaue, von *Survey2GIS* verwendete, Version von PROJ.4 wird in der Programmausgabe (Log) vermerkt. Das mitgelieferte Unterverzeichnis `names „proj“` enthält die von PROJ.4 unterstützten EPSG-Einträge (in der Datei `„epsg“`: s. 8.1.2) und einige Grid-Dateien (eigene können hinzugefügt werden: s. 8.2.2). Es handelt sich um ein wichtiges PROJ.4-Datenverzeichnis, das in dem Verzeichnis liegen sollte, in welchem sich auch die ausführbare Datei `„survey2gis“` befindet (im Falle der Windows-Version ist dies `„survey2gis.exe“`, *nicht* `„survey2gis.bat“`):

Der `„proj“`-Ordner und die `„epsg“`-Datei darin sind Bestandteil der Distribution von *Survey2GIS*, so dass PROJ.4 problemlos funktionieren sollte.

Als Alternative lässt sich die Umgebungsvariable `„PROJ_LIB“` auf den vollständigen Pfad zu `„proj“` setzen. Als dritte und letzte Möglichkeit sucht PROJ.4 automatisch nach einer systemweiten Kopie seines Datenordners (der genaue Pfad unterscheidet sich, je nach Betriebssystem).

Existieren das Verzeichnis `„proj“` nicht an einem der genannten Orte oder ist es nicht lesbar, dann wird *Survey2GIS* weiterhin funktionieren, aber die Funktionalität zur Reprojektion wird eingeschränkt und bestimmte Operationen (z. B. solche die EPSG-Nummern als KBS-Angaben beinhalten) führen zu Fehlermeldungen.

### 8.2.1 Datumstransformationen

Bei Reprojektionen die unterschiedliche *Kartendatums* beinhalten (z. B. „WGS 84“ nach „ED50“), besteht der Schlüssel zur Minimierung des Transformationsfehlers in der Verwendung der besten verfügbaren *Datumstransformation*.

Ein geodätisches (Karten-)Datum repräsentiert essentielle Informationen, die beschreiben wie (wo) ein KBS auf dem Globus „verankert“ und orientiert ist. Für gewöhnlich zählt man das verwendete Erdmodell (bspw. ein geodätisches Ellipsoid) ebenfalls zum (erweiterten) Datum.

Im Verlauf der Geschichte wurden zahlreiche unterschiedliche Datums an unterschiedlichen Stellen der Welt etabliert. In jüngerer Zeit hat sich eine kleinere Zahl von Datums in der internationalen Kartographie durchgesetzt. Das häufigste Datum, welches auch dem modernen UTM-System zugrundeliegt, ist das „World Geodetic Datum“ von 1984 („WGS 84“), aber andere, wie etwa „ETRS89“ sind ebenfalls weiterhin in Gebrauch.

Zur Berechnung einer Reprojektion verwendet *Survey2GIS* (oder genauer: die eingebaute Kopie von PROJ.4, die es verwendet), WGS 84 als „Schlüsseldatum“: Wenn eine Transformation zwischen verschiedenen Datums notwendig ist, dann werden die Daten stets über WGS 84 als gemeinsamem Bezugspunkt transfor-

miert. Um diesen Prozess steuerbar zu machen, lassen sich die Parameter einer Datumstransformation *vom Quell-KBS nach WGS 84* setzen.

Die Verwendung von KBS-Kürzeln und EPSG-Nummern für die Angabe von Quell- und Ziel-KBS (s. 8.1) führt i. d. R. dazu, dass Vorgabewerte für eine Transformation nach WGS 84 bereits als Teil der KBS-Definition enthalten sind und entsprechend bei der Reprojektion angewendet werden. Es gibt jedoch keine Garantie, dass diese Vorgabewerte das bestmögliche Resultat erzielen. Aus diesem Grund können alle Transformationsparameter auch vom Benutzer gesetzt werden.

Man unterscheidet hierbei zwischen zwei Typen von Datumstransformationen: 3-Parameter-Transformationen beinhalten Translations- (Verschiebungs-) Faktoren nach WGS 84, entlang der geographischen *X*, *Y* und *Z*-Achsen, während 7-Parameter-Transformationen zusätzlich drei Rotationsfaktoren (um dieselben drei Achsen) und einen Skalierungsfaktor enthalten.

Die Werte für die Transformationsparameter können über die entsprechenden Optionen von *Survey2GIS* gesetzt werden:

```
--proj-dx  
--proj-dy  
--proj-dz  
--proj-rx  
--proj-ry  
--proj-rz  
--proj-ds
```

Die ersten drei Optionen („dx“, „dy“ und „dz“) der obigen Liste entsprechen den Translationsfaktoren einer 3-Parameter-Transformation. Die nächsten drei („rx“, „ry“ und „rz“) und der letzte Faktor („ds“) entsprechen den drei Rotations- und dem Skalierungsfaktor einer 7-Parameter-Transformation.

Beachten Sie, dass diese Parameterwerte auch als Teil einer PROJ.4-Definition (Element „towgs84“: s. 8.1.3) stehen können. Optionen, die an *Survey2GIS* übergeben werden, haben aber stets Vorrang und ersetzen/überschreiben die entsprechenden Elemente einer PROJ.4-Definition (sofern vorhanden) automatisch. Es ist auch möglich, nur einige der obigen Parameter anzugeben (d. h. weniger als für eine vollständige 3- oder 7-Parameter-Transformation notwendig wären). In diesem Fall ergänzt *Survey2GIS* automatisch die fehlenden Parameterwerte um Vorgabewerte. Der Vorgabewert für den Skalierungsfaktor („--proj-ds“) ist „1“, alle anderen Werte (also Translations- und Rotationsfaktoren) sind per Vorgabe „0“.

Benutzerdefinierte Transformationsparameter werden von *Survey2GIS* nur dann angewendet, wenn mindestens ein Parameterwert von seinem Vorgabewert abweicht.

Wichtig: Wenn durch den Benutzer keine Transformationsparameter angegeben werden, bedeutet dies nicht unbedingt, dass während der Reprojektion keine Datumstransformation durchgeführt wird. So enthalten bspw. die meisten EPSG-Einträge (s. 8.1.2) Datumstransformationen, die automatisch angewendet werden, sofern keine vom Benutzer angegebenen Parameterwerte vorliegen. In jedem Fall gibt *Survey2GIS* sämtliche Details zu einer aktiven Datumstransformation als Teil seines Logs aus.

Unter den folgenden Bedingungen wird eine Fehlermeldung ausgegeben und das Programm abgebrochen:

- Sowohl Transformationsparameter als auch eine Grid-Datei (s. 8.2.2) wurden angegeben.
- Transformationsparameter oder eine Grid-Datei wurden angegeben, aber „--proj-in“ oder „--proj-out“ fehlen (beide müssen angegeben werden um eine Reprojektion berechnen zu können).

Wenn das *Ziel-KBS* keine Angaben zu einer Datumstransformation nach WGS 84 enthält, wird ihm automatisch eine synthetische 3-Parameter-Transformation mit allen Translationsfaktoren als „0“ hinzugefügt. Der Grund hierfür ist, dass die Transformationsparameter sowohl im Quell- als auch im Ziel-KBS vorhanden sein *müssen*, wenn eine Datumstransformation erforderlich ist (s. <http://proj4.org/faq.html#why-do-i-get-different-results-with-4-5-0-and-4-6-0>).

Wichtig: Aufgrund der bereits erwähnten Einschränkungen von PROJ.4 (s. 8.2) ist es *nicht* möglich, ein anderes Zeichen als den einfachen Punkt („.“) als Dezimaltrenner in den Werten von Transformationsparametern zu verwenden!

### 8.2.2 Reprojektionen mit Grids

In einigen Anwendungsfällen, etwa bei der Katastervermessung, sind die Genauigkeitsanforderungen an eine Reprojektion derart hoch, dass sie auch von den besten 7-Parameter-Transformationen nicht immer erfüllt werden können. In solchen Fällen besteht die Alternative in der Verwendung eines dichten Netzes von lokalen Ausgleichsvektoren, die speziell für das Ziel-KBS entwickelt wurden.

Eine Datei („Grid-Datei“) mit derartigen Korrekturdaten kann *Survey2GIS* übergeben werden, welches diese wiederum an PROJ.4 durchreicht.

Die Ausgleichsdaten in der Grid-Datei müssen der Ausdehnung der Daten im *Ziel-KBS* entsprechen, andernfalls bleiben sie wirkungslos.

Vollständiger Pfad und Name der Grid-Datei können *Survey2GIS* über die entsprechende Option mitgeteilt werden:

```
--proj-grid=
```

Weitere Informationen zum Thema finden sich auf der entsprechenden PROJ.4-Hilfeseite: [http://proj.maptools.org/gen\\_parms.html](http://proj.maptools.org/gen_parms.html).

Beachten Sie, dass PROJ.4 auch die Verwendung von mehr als einer Grid-Datei erlaubt, *Survey2GIS* aber nur eine einzige Datei per Option „--proj-grid“ akzeptiert. Alternativ ist es daher möglich, als Teil einer PROJ.4-Definition (s. 8.1.3) mittels des Elements „+nadgrids“ mehrere Grid-Dateien (Liste durch Kommata getrennt) anzugeben – dann kommt automatisch die erste Grid-Datei zur Anwendung, welche die Ausdehnung der zu reprojizierenden Daten im Ziel-KBS abdeckt. In diesem Fall sollte die Option „--proj-grid“ allerdings nicht verwendet werden, da deren Optionswert ein entsprechende PROJ.4-Element automatisch überschreibt.

Hinweis: Eine relative Pfadangabe im Optionswert von „--proj-grid=“ wird von *Survey2GIS* automatisch zu einem vollständigen, absoluten Pfad ergänzt, so dass PROJ.4 die Grid-Datei findet. Wird stattdessen das PROJ.4-Element „+nadgrids“ verwendet, um (eine) Grid-Datei(en) an PROJ.4 zu übergeben, so muss darauf geachtet werden, dass alle Pfadangaben vollständig sind.

### 8.3 Hilfestellungen zu KBS und Reprojektion

Die Reprojektion von Koordinaten aus einem Quell- in ein Ziel-KBS kann ein komplexer Vorgang mit vielen Tücken sein. In diesem Abschnitt werden einige Hinweise gegeben, was hierbei zu beachten ist und wie sich die Resultate evtl. optimieren lassen.

**Bewahrung geometrischer Details bei Daten mit großer Ausdehnung** Bei einigen Reprojektionen, insbesondere solchen, die eine Transformation von planimetrischen Koordinaten in Längen- und Breitengrade beinhalten, kann es zu signifikanten Einbußen geometrischer Details kommen. In solchen Fällen können u. U. bessere Ergebnisse produziert werden, wenn während der Vermessung zusätzliche Stützpunkte auf Linien und Polygongrenzen aufgenommen werden.

**Qualität von Datumstransformationen** Die Verwendung der besten verfügbaren Datumstransformation ist der Schlüssel zur Fehlerminimierung wenn zwischen KBS mit unterschiedlichen geodätischen Datums reprojiziert wird (s. auch 8.2.1). Insbesondere bei der Verwendung von EPSG-Einträgen empfiehlt es sich, die Datenbank auf <http://www.epsg-registry.org/> nach alternativen, evtl. für die Zielregion besser geeigneten Datumstransformationen zu durchsuchen.

**Konvertierung zwischen KBS-Formaten:** Die Seiten unter <http://www.spatialreference.org/> stellen eine nützliche Online-Datenbank für die Suche nach KBS in zahlreichen Formaten zur Verfügung, darunter EPSG, PROJ.4 und WKT (Well Known Text). Außerdem ist es u. a. möglich, KBS-Informationen als „.prj“-Dateien herunterzuladen und zu existierenden Shapefiles hinzuzufügen.

**Koordinaten als Längen-/und Breitengrade:** *Survey2GIS* erwartet alle Längen- und Breitengrade in Dezimalschreibweise. Für eine ausreichende Genauigkeit sind u. U. viele Dezimalstellen notwendig. Hilfreiche Informationen und Empfehlungen zu diesem Thema finden sich unter [https://en.wikipedia.org/wiki/Decimal\\_degrees](https://en.wikipedia.org/wiki/Decimal_degrees).

**Einheiten für vertikale (Z-) Koordinaten:** Für Reprojektionen bestimmt *Survey2GIS* (bzw. PROJ.4) automatisch die korrekte Einheit für Z-Koordinaten aus der KBS-Definition. Bei 3D-Transformationen zur Neuorientierung der Achsen (s. etwa 7.2) geht *Survey2GIS* dagegen *immer* davon aus, dass es sich um Koordinaten in der metrischen Einheit „Meter“ handelt. In einigen wenigen Szenarien können diese beiden Annahmen sich widersprechen und zu unerwünschten Resultaten führen.

## 9 Topologische Qualität und Bereinigung

Hinweis: Die hier beschriebene Funktionalität hat zur Prämisse, dass mit den Ursprungsdaten „etwas nicht stimmt“ und diese bereinigt werden müssen. Genauer formuliert, wird davon ausgegangen, dass die Eingabedaten (idealerweise nur leicht) von den eigentlich *intendierten* Geometrien soweit abweichen, dass eine gewisse Bereinigung sinnvoll ist, welche die Punkte, Linien und Polygone dahingehend justiert, dass sie besser für die Weiterverarbeitung im 2D GIS geeignet sind.

Dabei geht *Survey2GIS* außerdem davon aus, dass die Eingabedaten Geometrien auf einer relativ planaren Oberfläche repräsentieren, so dass sie ihren 2D-Abbildern (d. h. konstantes  $Z = 0$  in den Koordinatenwerten) stark ähneln und nur wenige Über- oder Unterschneidungen im 3D-Raum zu erwarten sind. Falls dies nicht der Fall ist und die Daten lieber so erhalten werden sollen, wie eingemessen, dann sollten die Optionen „--tolerance=“, „--snapping=“ und „--dangling=“ jeweils auf „0“ gesetzt werden, um die topologische Bereinigung zu minimieren (die in 9.1.5 und 9.1.6 beschriebenen Funktionen laufen aber in jedem Fall automatisch ab).

Die GIS-gestützte Verarbeitung von räumlichen Daten stellt (im Gegensatz zu bspw. CAD) besondere Anforderungen an deren Qualität. Hierzu gehört die *topologische* Qualität der Daten. Darunter sind geometrische Beziehungen zwischen Objekten wie Polygonen zu verstehen, welche im Rahmen des GIS-Datenmodells zu gewährleisten sind.

In *Survey2GIS* laufen einige Funktionen nach dem Datenimport automatisch ab, die gewährleisten sollen, dass die topologische Qualität der Daten erhöht wird (beachte, dass dies nur zum Export ausgewählte Geometrien betrifft, s. 6). Hierbei ist einschränkend zu beachten, dass das GIS-Topologiemodell ein zweidimensionales ist. Ebenso sind die meisten Topologiefunktionen von *Survey2GIS* nur in 2D implementiert. Dies bedeutet, dass die Daten so betrachtet werden, als würde  $Z$  konstant „0“ betragen. Je nachdem, wieviel Varianz die  $Z$ -Daten enthalten, kann dies zu mehr oder weniger signifikanten Abweichungen von der Realität führen. Die topologische Bereinigung funktioniert daher besser, je planarer die Daten sind.

Obwohl einige GIS eigene, teils sehr leistungsfähige, Werkzeuge zur topologischen Bereinigung zur Verfügung stellen, sollte die grundsätzliche Arbeit besser von *Survey2GIS* verrichtet werden, da ihm die vollen Eingabedaten zur Verfügung stehen, während das GIS nur auf die bereits rekonstruierten Geometrien zurückgreifen kann. Topologische Probleme können an zwei Punkten der Verarbeitung auftreten bzw. behandelt werden: direkt beim Einlesen der Eingabedaten und nachdem die Daten zu kompletten Geometrien verflochten worden sind. Für beide Fälle gibt *Survey2GIS* separate Hinweise und Statistiken aus.

### 9.1 Probleme und Bereinigungen

Im Rahmen der Vermessungsarbeit ist eine überschaubare Menge topologischer Problemquellen und Fehler von Bedeutung, welche von *Survey2GIS* zum Teil automatisch bereinigt werden können. In diesem Abschnitt werden die entsprechenden Funktionen in derselben Reihenfolge besprochen, in der sie von *Survey2GIS* ausgeführt werden.

In der Praxis ist es unwahrscheinlich, dass eine automatische Bereinigung zu perfekten Ergebnissen führt. Echte Vermessungsdaten enthalten fehlerhafte Messungen und Extremdaten. Außerdem sind die Geometrien, so wie sie von *Survey2GIS* verarbeitet werden, nur schwach strukturiert (im Prinzip ist nur die Reihenfolge ausschlaggebend, in der die Daten in den Eingabedateien stehen) und ohne inhärente Hierarchie, welche bei der Bereinigung von topologischen Fehlern helfen könnte. Aus diesem Grund sollte das Ergebnis immer gründlich geprüft und ggfs. manuell verbessert werden. In einigen Fällen erlaubt die getrennte Verarbeitung von Linien und Polygonen mittels entsprechender Auswahlkommandos (s. 6) eine bessere Kontrolle über die Qualität des Ergebnisses. Es kann auch hilfreich sein, eine zusätzliche Version der Ausgabedaten mit minimaler topologischer Bereinigung (d. h. Programmoptionen „--tolerance=“, „--snapping=“ und „--dangling=“ jeweils auf „0“ gesetzt) zu erzeugen, um die Auswirkungen der automatischen Bereinigung im Vergleich beurteilen zu können.

### 9.1.1 Doppelte Stützpunkte (Ausdünnung)

Doppelte Stützpunkte sind solche, die „sehr eng“ zusammenstehen, also aus praktischer Sicht denselben Punkt darstellen. Solche Punkte repräsentieren redundante Messdaten, die gerade bei komplexen Polygonen das Risiko von Selbstüberschneidungen (s. 9.1.4 auf der nächsten Seite) erhöhen. Das Problem der doppelten Punktmessungen schließt auch doppelte Einzelmessungen, z. B. bei der Vermessung von Höhenmodellen, ein. Solche Messungen können bei späteren Interpolationen von Rastermodellen im GIS zur Übergewichtung des Messwerts an der betreffenden Stelle führen.

In *Survey2GIS* dient daher die Option „--tolerance“ dazu, pro Programmablauf den Abstand zu definieren, oberhalb dessen zwei Punkte als separate Messungen akzeptiert werden. Zwei Punkte, deren Abstand dem Wert von „--tolerance“ entspricht oder darunter liegt, gelten als doppelte Messungen. Bei solchen Messungen wird nur der erste Punkt in den Ausgabedaten erhalten bleiben. Alle folgenden redundanten Messungen werden automatisch verworfen.

Dieser Prozess wird auch als „Ausdünnung“ bezeichnet und wirkt sich je nach Geometrietyp unterschiedlich aus:

- Punkte werden ausgedünnt, indem alle Punkte, die näher als der Schwellenwert an *irgendeinem* anderen Punkt aus *derselben Eingabedatei* liegen, gelöscht werden.
- Stützpunkte auf Linien werden ausgedünnt, indem alle Stützpunkte entfernt werden, die näher als der Schwellenwert an einem direkt nachfolgenden Stützpunkt auf *derselben* Linie liegen.
- Ebenso werden Stützpunkte auf Polygongrenzen ausgedünnt, indem alle Stützpunkte entfernt werden, die näher als der Schwellenwert am direkt nachfolgenden Stützpunkt *derselben* Polygongrenze liegen.

Der Abstand zweier Punkte wird im dreidimensionalen Raum berechnet, wenn es sich um 3D-Punkte handelt. Damit bleiben absichtlich verdichtete Messungen an Geländekanten besser erhalten.

Bei Bedarf lässt sich die Punktausdünnung *vollständig abschalten*, indem für „--tolerance=“ ein Wert kleiner als „0“ gesetzt wird.

Achtung: Große Werte für den Schwellenwert (Option „--tolerance=“) können die Form von Linien und Polygonen signifikant verändern und unerwartete Ergebnisse produzieren. In den meisten Fällen sollte der Schwellenwert so gewählt werden, dass er der (geschätzten) Genauigkeit der Vermessung entspricht. Wenn Punktdaten aggressiver ausgedünnt werden sollen, empfiehlt es sich, diese in einem getrennten Durchgang, mittels eines entsprechenden Auswahlkommandos (s. 6) von *Survey2GIS* verarbeiten zu lassen.

### 9.1.2 Splitter

Splitter sind Geometrien, die Linien darstellen sollen, aber aus weniger als zwei Stützpunkten bestehen, oder Polygone, die aus weniger als drei Stützpunkten bestehen. In beiden Fällen werden die betroffenen Geometrien automatisch gelöscht.

### 9.1.3 Mehrteilige Geometrien

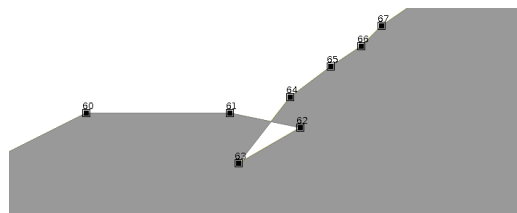
Wenngleich mehrteilige Geometrien kein topologisches Problem i. e. S. darstellen, müssen sie in den Geometriedaten korrekt repräsentiert sein, um redundante Daten zu vermeiden.

In machen Fällen müssen Geometrien in separaten, d. h. räumlich disjunkten Teilen, eingemessen werden. Dies kann der Fall sein, wenn sich ein Polygon im Planum in zwei getrennten Hälften abzeichnet oder dasselbe Objekt an unterschiedlichen Stellen in Ausschnitten sichtbar wird.

Damit dies funktioniert, muss jede Teilgeometrie denselben Wert im Feld „key\_field“ besitzen und die Parser-Option „key\_unique“ gesetzt sein (s. Beschreibung in 4 auf Seite 25). Ist dies der Fall, so wird *Survey2GIS* automatisch alle Teile einer Geometrie mit demselben Wert im Schlüsselwert als Einheit exportieren. Im GIS wird einem solchen Verbundobjekt dann nur ein Attributdatensatz zugewiesen.

### 9.1.4 Selbstüberschneidungen bei Polygonen

Bei der Digitalisierung der Grenzen komplexer Polygone kommt es, gerade bei dicht gesetzten Stützpunkten, manchmal zu *Selbstüberschneidungen*. Dabei wird ein Stützpunkt (oft auf den ersten Blick schwer erkennbar), hinter die Linie zwischen zwei existierenden Stützpunkten gesetzt. Das Resultat ist eine Überschneidung der Polygongrenze mit sich selbst an der entsprechenden Stelle. Bei der Visualisierung im GIS wird dies bei genauerem Hinschauen z. B. durch das Auftauchen von nicht gefüllten Flächen innerhalb eines Polygons sichtbar:



Selbstüberschneidungen gehören zu den hartnäckigsten topologischen Fehlern. Sind sie vorhanden, können selbst einfachste Operationen auf den Daten, wie z. B. eine korrekte Flächenberechnung, nicht durchgeführt



werden. Sie können nicht automatisch bereinigt werden, da *Survey2GIS* nicht wissen kann, welches die korrekte, intendierte Platzierung der Stützpunkte wäre. Das Risiko des Auftretens von Selbstüberschneidungen lässt sich aber reduzieren, indem die Option „--tolerance“ den Anforderungen an die Genauigkeit entsprechend gesetzt wird (s. [9.1.1 auf Seite 63](#)).

*Survey2GIS* prüft bereits beim Zusammenfügen von Linien und Polygonen aus den Messdaten, ob Selbstüberschneidungen vorliegen. Ist dies der Fall, so wird eine Warnung ausgegeben. Um die manuelle Bereinigung betroffener Geometrien zu erleichtern, fügt *Survey2GIS* am Punkt der Überschneidung einen neuen Stützpunkt zur betroffenen Geometrie hinzu.

Einige topologische Bereinigungsfunktionen können nicht korrekt arbeiten, wenn Selbstüberschneidungen vorliegen. In diesen Fällen wird *Survey2GIS* betroffene Objekte überspringen (d.h. unverändert lassen) und eine Warnung ausgeben. Prüfen Sie auch die Statusmeldung am Ende der Verarbeitung auf die Zahl erkannter Selbstüberschneidungen!

### 9.1.5 Gemeinsame Grenzen von Polygonen

Hinweis: Das automatische Entfernen von Überlappungsbereichen benachbarter Polygone steht seit Version 1.5.1 zur Verfügung.

Zwei benachbarte Polygone teilen sich entlang ihrer Kontaktlinie eine gemeinsame Grenze. Während der Datenaufnahme (Vermessung) müssen gemeinsame Grenzen aus praktischen Gründen jedoch doppelt eingemessen werden. Das Problem dabei ist, dass es kaum möglich sein wird, sämtliche betroffenen Stützpunkte zweimal exakt zu treffen. Aus diesen Gründen wird es in der Praxis unvermeidlich zu zwei Grundproblemen bei den produzierten Daten kommen:

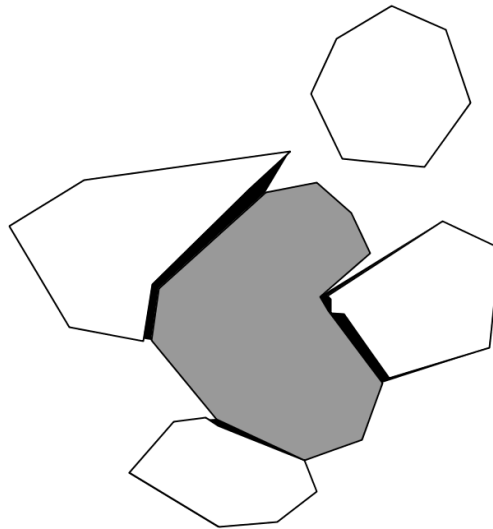
1. Zwischen zwei benachbarten Polygonen verbleibt ein Freiraum, obwohl sie eigentlich direkt aneinanderstoßen.
2. Die Grenzen zweier benachbarter Polygone überlappen.

Aus Sicht des 2D-GIS-Datenmodells stellt der zweite Fall einen topologischen Fehler dar, insofern die beteiligten Polygone im selben Layer gespeichert sind. Bei der Aufbereitung zu topologisch korrekten Daten ist es also wichtig, sowohl nicht intendierte Lücken zwischen den Polygongrenzen zu schließen, als auch Überlappungszonen zu entfernen. Hierfür bietet *Survey2GIS* zwei sich gut ergänzende Funktionen:

1. Kleine Lagedifferenzen zwischen den Stützpunkten gemeinsamer Grenzen können ausgeglichen werden, indem die Stützpunkte automatisch exakt übereinander gebracht werden.
2. Überlappende Bereiche von benachbarten Polygonen werden automatisch entfernt und Stützpunkte bei Bedarf neu gesetzt, so dass die benachbarten Polygone entlang der gemeinsamen Grenze über exakt dieselben Stützpunkte verfügen.

Nur die erste der obigen Funktion (der sogenannte „Stützpunktfang“) muss vom Benutzer gesteuert werden. Sie bewirkt, dass ein Stützpunkt, welcher zu einer *Außengrenze* von Polygon *B* gehört, automatisch exakt

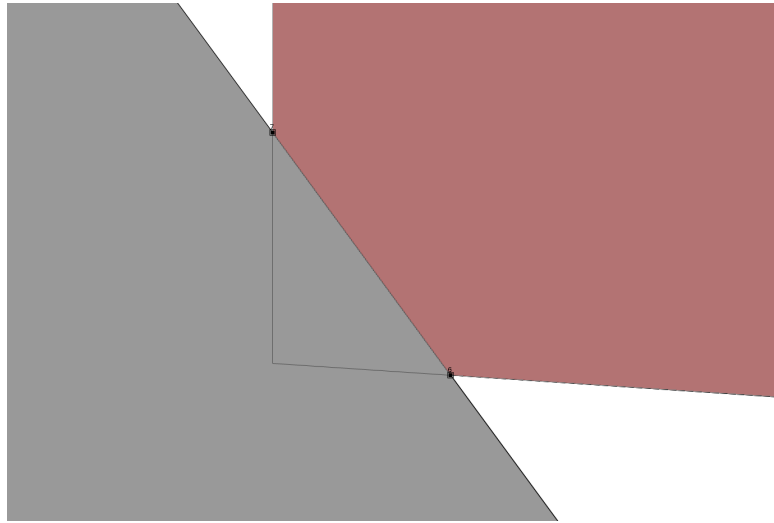
auf den Stützpunkt einer *Außengrenze* von Polygon *A* bewegt wird (einrastet; engl. *snapping*), sofern ein vom Benutzer zu wählender Abstand unterschritten wird und kein weiterer Stützpunkt von *A* noch näher gelegen ist.



Die Abbildung oben zeigt ein (graues) Polygon mit vier weiteren, benachbarten (weißen) Polygonen. Nach dem Snapping werden die Stützpunkte der drei nächsten Nachbarn derart auf der Außengrenze des zentralen Polygons eingerastet, dass ihre Flächen um die schwarzen Anteile erweitert werden.

Der Wert für die Distanz (d. h. Option „--snapping“) sollte so gewählt sein, dass er unterhalb des typischen Abstands separater Messungen, aber oberhalb der Genauigkeit mit der einmal eingemessene Punkte ein zweites Mal getroffen werden können, liegt.

Die zweite Funktion zur Bereinigung von Grenzen benachbarter Polygone arbeitet vollautomatisch: Sofern nach dem Stützpunktfang noch eine Überlappung zwischen zwei benachbarten Polygonen besteht, wird der betroffene Bereich automatisch aus dem *in den Messdaten später* stehenden Polygon ausgeschnitten. Außerdem werden automatisch zusätzliche Stützpunkte auf der gemeinsamen Grenze eingefügt, wo diese benötigt werden:



Damit die automatische Korrektur von Polygongrenzen gute Resultate liefert, empfiehlt es sich:

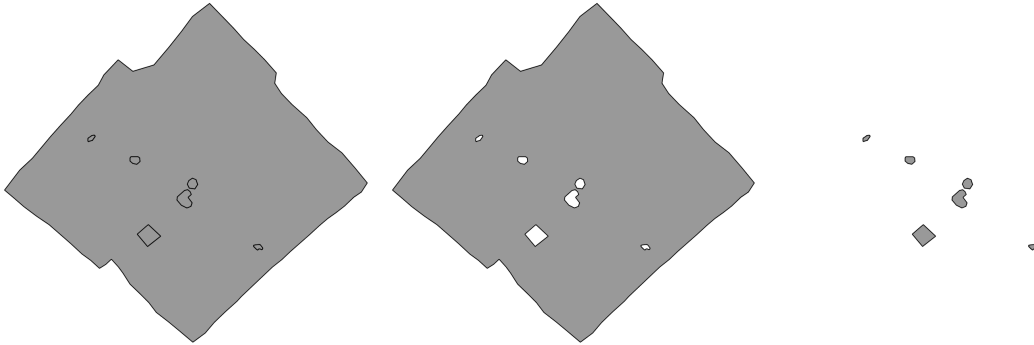
- Größere („wichtigere“) Befunde als Basisobjekte zuerst einzumessen.
- Kleinere, daran anstoßende Objekte im Anschluss einzumessen und deren Grenzen absichtlich ein gutes Stück (weiter als der Distanzschwellenwert den Stützpunktfang, s. o.!) *innerhalb* der Flächen des Basisobjekts mit wenigen Punkten einzumessen.

Die automatische Korrektur gemeinsamer Polygongrenzen wird in der Praxis in Extremsituationen an ihre Grenzen stoßen. Folgendes sollte beachtet werden:

- Relativ kleine Überlappungsbereiche lassen sich mit größter Wahrscheinlichkeit vollständig automatisch korrigieren.
- Diese automatische Korrektur ist eine reine 2D-Operation.
- Polygone mit Innengrenzen („Löchern“: s. 9.1.6) können nicht automatisch angepasst werden, wenn sich dadurch die Zahl der Löcher ändern würde (ansonsten werden Polygone mit Löchern aber korrekt verarbeitet).
- Bei der Überprüfung auf Überlappung werden Teile eines mehrteiligen Polygons (s. 9.1.3) wie eigenständige Polygone behandelt, um ein korrektes Ergebnis sicherzustellen.
- Eine extreme Überlappung, die dazu führen würde, dass ein Polygon in mehrere disjunkte Teile zerschnitten werden müsste, kann nicht automatisch korrigiert werden.
- Wenn die Programmoption „--force-2d“ (zum Erzwingen von 2D-Koordinaten in der Ausgabe) *nicht* gesetzt ist: Beim automatischen Stutzen von Polygonen entstehen i. d. R. neue Stützpunkte. Die Z-Werte für diese werden per linearer Interpolation aus den benachbarten Stützpunkten auf dem anderen Polygon berechnet.

Zu beachten ist außerdem, dass die eben beschriebenen Korrekturen mit hoher Wahrscheinlichkeit nicht wirken, wenn sie auf Polygone mit anderen topologischen Problemen, z. B. Selbstüberschneidungen (s. [9.1.4](#)), angewendet werden.

#### 9.1.6 Innengrenzen von Polygonen (Löcher)



Das 2D-Datenmodell des GIS bedingt, dass sich Polygone innerhalb eines Layers in ihren Außengrenzen nicht überlappen dürfen, da dies ein topologischer Fehler wäre.

Was aber, wenn im Planum ein Polygon *B* sichtbar ist, welches *vollständig* innerhalb der Außengrenzen eines anderen Polygons *A* liegt (eingebettet ist)? In diesem Fall sollte *B* als Innengrenze („Loch“) von *A* modelliert werden. In *Survey2GIS* existieren hierfür zwei verschiedenen Möglichkeiten:

1. Das Polygon wird als mehrteilige (mindestens zweiteilige) Geometrie eingemessen: zunächst das Basispolygon, dann eine beliebige Anzahl von darin liegenden weiteren Polygonen. Letztere werden von *Survey2GIS* dann automatisch zu Löchern konvertiert. Damit dies funktioniert, muss jede beteiligte Geometrie denselben Wert im Feld „key\_field“ besitzen und die Parser-Option „key\_unique“ gesetzt sein (s. Beschreibung in [4 auf Seite 25](#)).
2. Sofern es sich bei dem eingebetteten Polygon um ein eigenständiges Objekt handelt, kann dieses als separates Polygon digitalisiert werden, indem ihm ein eigener Schlüsselwert zugewiesen wird. In diesem Fall wird *Survey2GIS* automatisch alle notwendigen Löcher in das darunter liegende Polygone „stanzen“, so dass alle darüber liegenden Polygone perfekt und überlappungsfrei eingepasst werden (s. Abbildung oben).

Damit die Modellierung von Löchern in Polygonen einwandfrei gelingt, ist es sehr wichtig, die Polygone in der richtigen Reihenfolge einzumessen (s. auch Hinweise unter [12 auf Seite 87](#))!

#### 9.1.7 Kreuzungspunkte auf Linien und Polygongrenzen

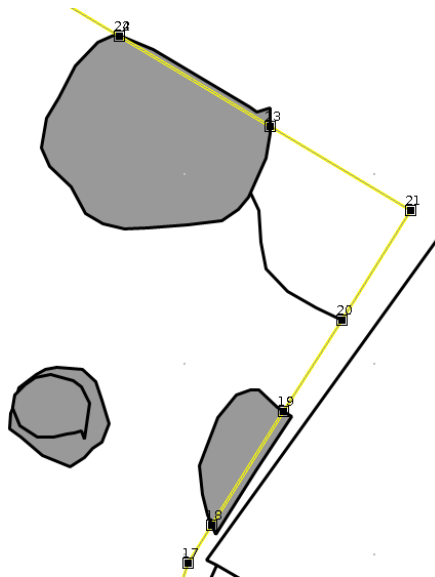
Umfangreichere Vermessungen produzieren häufig Geometrien, die sich gegenseitig überschneiden und kreuzen. Den Kreuzungspunkten kommt bei der manuellen Nachbearbeitung von Geometrien (z. B. im GIS) eine

besondere Bedeutung zu. Sich kreuzende Linien sind nur dann als topologisch korrekt anzusehen, wenn beide Linien am Kreuzungspunkt über einen Stützpunkt verfügen. Indem man Kreuzungspunkte bewegt oder löscht, kann man häufig die topologische Qualität der Daten verbessern. Kreuzungspunkte ermöglichen außerdem das automatische Löschen von „hängenden“ Linienenden (s. 9.1.8).

Daher markiert *Survey2GIS* Kreuzungspunkte automatisch und fügt diese als zusätzliche Stützpunkte den Geometrien hinzu. Dies geschieht in drei Fällen:

1. Linien kreuzen andere Linien (Stützpunkte werden allen beteiligten Linien hinzugefügt).
2. Linien kreuzen Polygongrenzen (Stützpunkte werden nur den Linien hinzugefügt, *nicht* den Polygonen).
3. Polygongrenzen kreuzen andere Polygongrenzen (Stützpunkte werden allen beteiligten Polygonen hinzugefügt).

(Letzterer Fall sollte seltener eintreten, wenn die automatische Korrektur von überlappenden Grenzen effektiv zum Einsatz kommt (s. 9.1.5).



Das Programm gibt zwei unterschiedliche Kennzahlen zu Kreuzungspunkten aus:

Erkannte Linie/Polygon-Kreuzungen: 12

Zu Linie/Polygon-Kreuzungen hinzugefügte Stützpunkte: 8

Manchmal wird die Zahl der hinzugefügten Stützpunkte kleiner sein, als die der erkannten Kreuzungen. Dies kann der Fall sein, wenn z. B. eine Linie von mehreren anderen an exakt demselben Punkt gekreuzt wird. Derartige, redundante Kreuzungspunkte werden automatisch reduziert.

Hinweis: Da *Survey2GIS* nicht wissen kann, welche Geometrie der Benutzer u. U. wie manuell korrigieren möchte, fügt es zu allen beteiligten Geometrien desselben Typs Stützpunkte hinzu.

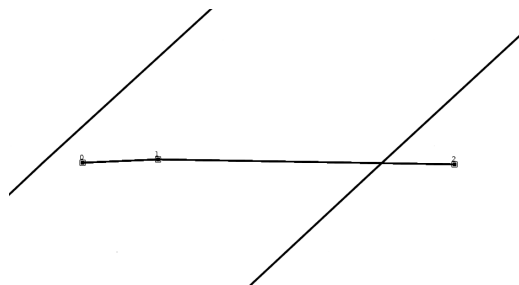
Selbstüberschneidungen von Linien werden nicht als topologische Fehler gewertet, aber ein neuer Stützpunkt wird am Kreuzungspunkt eingefügt. Demgegenüber *sind* Selbstüberschneidungen von Polygongrenzen topologische Fehler (s. 9.1.5). Wenn eine solche erkannt wird, dann gibt das Programm eine Warnung aus und ein Topologiefehler wird vermerkt, aber es werden *keine* neuen Stützpunkte eingefügt.

### 9.1.8 Hängende Linienknoten (Über- und Unterschwinger)

Bedingt durch die beschränkte Genauigkeit der Vermessung im Feld, werden in vielen Fällen die Start- und Endpunkte (im Folgenden: „Knoten“) nicht dort eingemessen, wo sie eigentlich liegen sollten (etwa genau auf einer anderen Linie oder Polygongrenze). Solche Fälle werden als „hängende“ (engl. *dangling*) Linienknoten bezeichnet, wobei zwischen zwei Ausprägungen unterschieden wird:

1. Linienknoten die vor ihrem intendierten Ort liegen, bezeichnet man als Unterschwinger.
2. Linienknoten die hinter ihrem intendierten Ort liegen, bezeichnet man als Überschwinger.

Die Abbildung unten zeigt eine Linie, die sowohl einen Unter- (Stützpunkt „0“) als auch einen Überschwinger (Stützpunkt „2“) enthält.

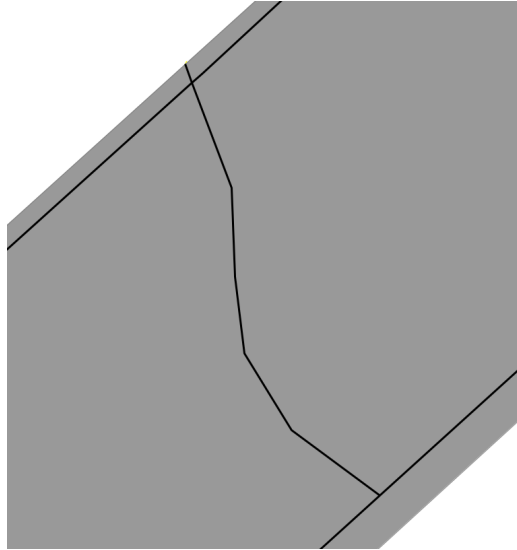


Das Programm besitzt deshalb eine Option „--dangling=“, mittels derer sich ein Schwellenwert für das Einfangen von Linienknoten auf das nächste Liniensegment/Polygongrenze festlegen lässt. Eine separate Steuerung ist deshalb notwendig, da das Einfangen von Polygonstützpunkten entlang gemeinsamer Grenzen (s. 9.1.5) zuerst durchgeführt wird und dabei die Polygongrenzen soweit verschoben werden können, dass Über- und Unterschwinger nicht mehr automatisch mit demselben Schwellenwert korrigiert werden können. In der Praxis sollte der Wert für „--dangling=“ daher höher gewählt werden, als derjenige für „--snapping=“.

Nur der erste und letzte Stützpunkt einer Linie können als Über- oder Unterschwinger gelten, und dies auch nur dann, wenn der nächste bzw. vorhergehende Stützpunkt ein durch *Survey2GIS* eingefügter Kreuzungspunkt (s. 9.1.7) ist. Wenn diese Kriterien keine befriedigenden Resultate bei der automatischen Korrektur bringen, dann kann das Heraufsetzen von „--threshold“ u. U. dafür sorgen, dass sehr kurze Ketten aus dicht gesetzten Linienstützpunkten zu einem einzigen Stützpunkt verschmolzen werden (s. 9.1.1). Letzteres kann

jedoch dazu führen, dass das Einfangen die Geometrie der Linie leicht verschiebt, da die ursprüngliche Linieneinrichtung evtl. ebenfalls verschoben wird. Andernfalls können die von *Survey2GIS* automatisch eingefügten Kreuzungspunkte immer noch zur manuellen Korrektur genutzt werden.

Die Korrektur von Über- und Unterschwingern verändert in beiden Fällen die ursprünglichen Linienlängen und sie kann zu unerwarteten Ergebnissen führen, wenn die ersten oder letzten  $n$  Stützpunkte allesamt Kreuzungspunkte sind. Letzteres führt dazu, dass es zu einem Fangen auf die *nächste* Linie oder Polygongrenze kommt, welches nicht immer der Intention der Benutzerin entsprechen wird. Die Abbildung unten zeigt eine Linie, welche sowohl auf eine andere Linie als auch auf eine Polygongrenze gestutzt wurde.



In anderen Fällen kann das, was wie ein Überschwinger aussieht, ein Unterschwinger sein, wenn man den Stützpunkt auf eine andere Geometrie bezieht. Dann wählt *Survey2GIS* diejenige Korrektur, welche der kürzesten Distanz entspricht (sollte der extrem unwahrscheinliche Fall eintreten, dass Über- und Unterschwinger von derselben Länge sind, so wird eine Korrektur für Überschwinger angewendet).

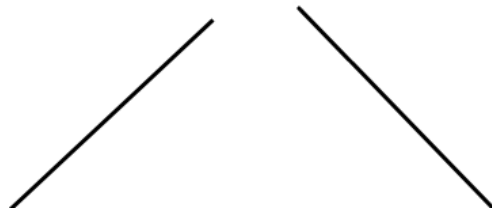
Diese automatische Korrektur hat einige Leistungsgrenzen:

- Unterschwinger innerhalb einer Geometrie können nicht gefunden und korrigiert werden.
- Unterschwinger werden per Extrusion über eine in 2D berechnete Strecke korrigiert, was für 2D-GIS-Layer akkurate Ergebnisse liefert. Bei sehr langen Unterschwingern, die evtl. noch dazu an Liniensegmenten mit starkem Z-Gefälle hängen, kann dies zu einer signifikanten Verfälschung des 3D-Verlaufs des betroffenen Liniensegments führen.
- Die Korrektur von Überschwingern an kurzen Linien, die aus nur zwei Segmenten bestehen, kann zum Extremfall führen, bei dem die gesamte Linie gelöscht werden müsste. In diesem Fall wird die Linie nicht verändert, sondern nur eine Warnung ausgegeben und ein Topologiefehler vermerkt.

Aus diesen Gründen sollte das Ergebnisse der automatischen Korrektur stets sorgfältig geprüft und ggfs. manuell verbessert werden (zur Erleichterung fügt *Survey2GIS* an wichtigen Stellen Stützpunkte ein: [9.1.7](#)).

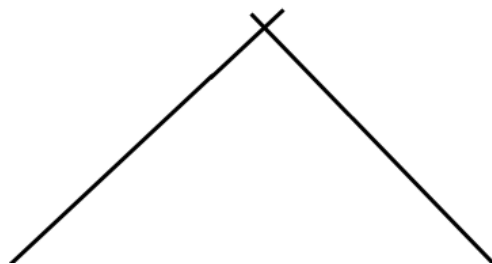
Um eine übermäßig radikale Stützung bzw. Extrusion von Linien zu vermeiden, sollten kleine Schwellenwerte für die Korrektur gewählt werden.

Ein besonders schwieriges Problem ergibt sich aus „doppelten Unterschwingern“, die häufig an den Ecken von Linienzügen auftreten und nicht automatisch erkannt werden können. Da hier auch keine Kreuzungspunkte eingefügt werden können, sind sie auch manuell nur schwer korrigierbar:



Um „doppelte Unterschwinger“ zu vermeiden, gibt es während der Vermessung zwei Möglichkeiten:

- Die Linienzüge werden stattdessen als Polygone kodiert, so dass sie automatisch geschlossen werden.
- Das erste wird mit dem letzten Segment absichtlich gekreuzt, so dass ein Kreuzungspunkt entsteht, an dem die manuelle oder automatische Entfernung von Überschwingern ansetzen kann:





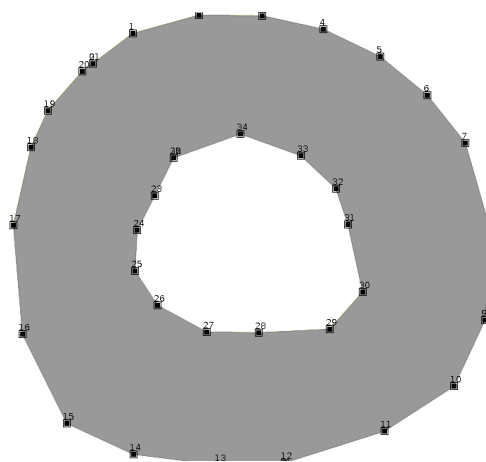
### 9.1.9 Polygonrichtung

Die Digitalisierungsrichtung, d. h. die Reihenfolge der Stützpunkte eines Polygons, ist im GIS von Bedeutung. Die meisten GIS versuchen, eine korrekte Stützpunktreihenfolge automatisch herzustellen. Ist dies jedoch nicht erfolgreich, so kann es zu Fehlern bei der Darstellung und räumlichen Abfragen von Polygondaten kommen, die für den Benutzer schwer erklärlich sind.

Insbesondere die Darstellung von 3D Daten mit heterogenen Polygonrichtungen führt oft zu visuellen Defekten.

Folgende Regeln sind daher zu beachten:

1. Die Digitalisierung der Außengrenzen von Polygonen (auch wenn es sich um mehrere, separate Teile handelt) sollte stets im Uhrzeigersinn erfolgen.
2. Die Digitalisierung von Innengrenzen („Löchern“) in Polygonen sollte stets gegen den Uhrzeigersinn erfolgen.



Beim Konstruieren von Polygonen aus den Messdaten versucht *Survey2GIS* automatisch, diese beiden Regeln einzuhalten. Um Probleme zu vermeiden, ist es jedoch sinnvoll, sie bereits bei der Feldarbeit zu beachten. Da jedes Vermessungssystem eine Nord-Süd-Achse besitzt, ist es möglich, die Konzepte „im Uhrzeigersinn“ und „gegen den Uhrzeigersinn“ eindeutig umzusetzen.

## 9.2 Praktische Hinweise und Einschränkungen

Praktisch alle geläufigen GIS kennen nur ein zweidimensionales Topologiemodell aus der Draufsicht. Dadurch kann es zu topologischen Problemen kommen, obwohl die zu verarbeitenden Daten im 3D-Raum eigentlich topologisch fehlerfrei sind. Besonders das Problem der überlappenden Polygone ist in der GIS-Datenverarbeitung heikel, da aus der 2D-Perspektive auch Polygone als überlappend erscheinen können, die es im 3D-Raum nicht sind.

Wenn dreidimensional korrekte Daten ein größeres Problem für die 2D-GIS-Analyse darstellen sollten, so besteht die Möglichkeit, zweidimensionale Daten von *Survey2GIS* erzeugen zu lassen, indem nur die *X*- und *Y*-Koordinatenfelder als solche definiert werden (s. 4.4 auf Seite 36). In diesem Fall arbeitet dann auch die topologische Bereinigung von *Survey2GIS* quasi in 2D. Die produzierten Daten sind genau genommen weiterhin dreidimensional, ihre *Z*-Koordinaten werden jedoch einheitlich auf „0“ gesetzt, was sie vollkommen planar macht.

Gerade bei sehr komplexen Polygonen ist auf jeden Fall eine weitere Kontrolle der topologischen Qualität und anschließende manuelle Bereinigung unter Zuhilfenahme der im GIS vorhandenen Werkzeuge ratsam.

Die Topologie-Funktionen in *Survey2GIS* sind nicht dazu geeignet, grobe oder sehr komplexe Topologiefehler (etwa mehrfache Überlappungen oder „Löcher in Löchern“) zu bereinigen. Das beste Mittel zu deren Vermeidung ist eine „saubere“ Arbeitsweise bei der Vermessung.

Alle Funktionen zur topologischen Bereinigung laufen *nach* einer evtl. Reprojektion (s. 8) ab, um zu vermeiden, dass letztere in ihrer Genauigkeit leidet. Sie laufen aber vor einer evtl. Transformation der Achsen (s. 7) ab, so dass die Ergebnisse von der Orientierung des Koordinatensystems unabhängig sind.

Daten, die in Längen- und Breitengraden vorliegen, können nur mit begrenzter Qualität bereinigt werden. *Nur* die folgenden Funktionen werden für derartige Daten ausgeführt:

- Entfernen redundanter Stützpunkte (s. 9.1.1) und „Splitter“ (s. 9.1.2).
- Prüfung auf Selbstüberschneidungen (s. 9.1.4).
- Stützpunktfang („Snapping“) für benachbarte Polygongrenzen: 9.1.5.
- Korrektur der Stützpunktreihenfolge von Polygonen (s. 9.1.9).

Das Entfernen redundanter Stützpunkte und der Stützpunktfang arbeiten mittels eines planaren Distanzmaßes (in diesem Fall in Dezimalgrad anzugeben!), was im Fall von Längen- und Breitengraden eine reduzierte Genauigkeit nach sich zieht. Das Programm wird eine entsprechende Warnung ausgeben.

Wichtig: Wenn topologische Bereinigung mit einer Datenauswahl (s. 6) kombiniert wird, können sich die resultierenden Geometrien von einem Ergebnis ohne Auswahl unterscheiden! Der Grund dafür ist, dass nicht in die Auswahl eingeschlossene Geometrien keinen Effekt mehr auf die topologische Bereinigung der verbleibenden Geometrien haben. Bspw. werden die Grenzen von Polygonen nicht beschnitten, wenn sie mit Grenzen von Polygonen überlappen, die nicht Teil der Auswahl sind. Falls dieses Verhalten unerwünscht ist, ist es besser, die Daten vollständig auszugeben und im Anschluss manuell, bswp. mit den Auswahlfunktionen eines GIS, zu selektieren.

## 10 Ausgabeformate

Um die weitere Verarbeitung mit unterschiedlichen Anwendungen zu ermöglichen, unterstützt *Survey2GIS* verschiedene Ausgabeformate.

Für die Archivierung sollten neben den von *Survey2GIS* produzierten Dateien sollten stets auch Kopien der originalen Vermessungsdaten aufbewahrt werden.

### 10.1 ESRI Shapefile

Option:

```
-f shp
```

Das Format „ESRI Shapefile“ ist das Standardformat für die Datenausgabe durch *Survey2GIS*. Es ist zwar kein von einem unabhängigen Gremium zertifizierter Standard, hat aber die Bedeutung eines Quasi-Standards in der GIS-Welt. Dies liegt vor allem daran, dass der Hersteller ESRI eine vollständige und frei verfügbare Beschreibung des Formats publiziert hat (<https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>).

Ein „Shapefile“ ist genau genommen keine einzelne Datei, sondern ein Satz von mindestens drei Dateien mit den Endungen „.shp“, „.shx“ und „.dbf“. Letztere kennzeichnet eine Datei im DBase-Format, welche die Attributdaten zu jeder gespeicherten Geometrie enthält.

Das Format unterliegt einigen Beschränkungen, die zum Teil wiederum aus der Verwendung des DBase-Formats für die Speicherung der Attributdaten hervorgehen:

1. Die maximale Dateigröße wird durch die Größe des Index in der „.shx“-Datei beschränkt. Typische Grenzen liegen, je nach GIS, bei 2 oder 4 GB.
2. Feldnamen dürfen maximal 10 Zeichen lang sein (DBase-Beschränkung).
3. Ein in der Attributtabelle zu speichernder Zahlenwert kann maximal 18 Stellen beinhalten. Dies schließt Vor- und Nachkommastellen ein. Bei sehr großen Zahlenbereichen muss die Genauigkeit (Zahl der Nachkommastellen) daher u. U. begrenzt werden (DBase-Beschränkung).
4. Ein zu speichernder Textwert darf maximal 254 Zeichen lang sein (DBase-Beschränkung).
5. Die Attributtabelle im DBase-Format speichert keine Informationen zur Kodierung von Textdaten. Werden nationale Sonderzeichen verwendet, die nicht Bestandteile der ASCII-Tabelle sind, so muss der Anwender selbst sicher stellen, dass alle verwendeten Programme diese korrekt unterstützen (DBase-Beschränkung).
6. Ein Shape-Dateisatz kann nur entweder Punkte, Linien oder Polygone enthalten. Aus diesem Grund wird *Survey2GIS* automatisch zusätzliche Ausgabedateien erstellen, wenn die Eingabedaten mehr als einen Geometrietyp enthalten.

### 10.1.1 ESRI Shapefiles und Etiketten

Bei der Erstellung von Etiketten (s. 5) unter Verwendung von „Shapefile“ als Ausgabeformat wird ein zusätzliches Shapefile mit Punkten erzeugt, das die folgenden Attributfelder besitzt:

**labeltext** Dieses Feld enthält den Text, der für die Etikettierung am jeweiligen Punkt verwendet wird.

**fonttype** Ein Feld vom Typ „Text“, das den Namen des Schriftsatzes (Font) enthält, welcher für die Etikettierung verwendet werden soll. Per Voreinstellung verwendet *Survey2GIS* den Eintrag „Arial“.

**fontstyle** Dieses Feld enthält die Textauszeichnung als Integerwert. Per Voreinstellung wird es auf „0“ (normal) gesetzt. Verwendete Werte sind: „0“ (normal), „1“ (fett) und „2“ (kursiv).

**fontcolor** In diesem Feld wird die Zeichenfarbe als Integerwert gespeichert. Die Kodierung folgt derjenigen der Java-Klasse *java.awt.Color* (s. [https://docs.oracle.com/javase/7/docs/api/java/awt/Color.html#getRGB\(\)](https://docs.oracle.com/javase/7/docs/api/java/awt/Color.html#getRGB())). Der von *Survey2GIS* gesetzte Wert ist „-16777216“ (schwarz).

**fontsize** Dieses Feld vom Typ „Double“ enthält die Zeichengröße (wird von *Survey2GIS* auf „10.0“ gesetzt).

**fontrotate** Dieses Feld vom Typ „Double“ enthält die Textrotation (im Uhrzeigersinn) und wird von *Survey2GIS* auf „0.0“ gesetzt.

**geomtype** Dieses Feld vom Typ „Integer“ kodiert den Geometrietyp, auf den sich das jeweilige Etikett bezieht. Die möglichen Werte sind: „0“ (Punkt), „1“ (Linie) und „2“ (Polygon).

Das obige Schema wurde aus Gründen der Kompatibilität mit den „Annotations“-Funktionen von *gvSIG CE* (<http://gvsigce.org>) gewählt. Aus demselben Grund wird eine Datei mit der Endung „.gva“ erzeugt, welche die Etikettierungseinstellungen enthält (wird von anderen GIS ignoriert). Etiketten-Shapefiles, die von *Survey2GIS* erzeugt wurden, könnten direkt als Annotationsebenen in *gvSIG CE* geladen und mit den dortigen Werkzeugen weiter bearbeitet werden.

### 10.1.2 „Null“-Werte in DBF Attributtabellen

Sollte der Benutzer es nicht anders festgelegt haben (s. Option „no\_data“ in 4.3), dann gilt das Folgende: Attributfelder, deren Inhalt „Null“ ist, d. h. die keine Daten enthalten, werden in der DBF-Datei dargestellt, indem das gesamte Feld mit Leerzeichen gefüllt wird. Wie dies interpretiert wird, hängt vom GIS ab. Üblicherweise werden Textfelder als leere Felder, numerische Felder als „0“ wiedergegeben.

## 10.2 Drawing Exchange Format (DXF)

Warnung 1: Das Format „DXF“ eignet sich nicht für die langfristige Archivierung und Verfügbarmhaltung von Vermessungsdaten (Details s. u.)!

Warnung 2: Daten im Format „DXF“ eignen sich i. d. R. nicht für räumliche Analysen, die topologisch korrekte Eingabedaten benötigen (Details s. u.)!

Option:

-f dxf

Beim DXF (Drawing Exchange Format) handelt es sich um ein Dateiformat, das von den meisten CAD-Programmen verarbeitet werden kann. Im Vergleich zu anderen CAD-Formaten, wie etwa DWG, besitzt es eine *relativ* einfache Struktur und offen einsehbare Dokumentation durch den Hersteller AutoDesk (<https://www.autodesk.com/techpubs/autocad/acad2000/dxf/>).

DXF erlaubt die Speicherung aller Geometrietypen in einer einzigen Datei. Für die Speicherung von Attributdaten ist es allerdings kaum geeignet, da es keine relationalen Datenstrukturen unterstützt. In der DXF-Datei kann lediglich der Inhalt eines Objekt-Handles gespeichert werden. Letzteres kann genutzt werden, um einen Primärschlüssel zu speichern, über den mittels GIS weitere Daten aus einer separaten Attributtabelle angehängt werden können.

Die von *Survey2GIS* produzierte DXF-Datei hat die folgenden Eigenschaften:

- Die DXF-Version entspricht „AC1015“ (AutoCAD Release 15/AutoCAD 2000).
- Punkte, „rohe“ Punktmessungen, Linien und Polygone werden in separaten Ebenen abgelegt. Zu „rohen“ Punktmessungen existiert eine Etikettierungsebene mit den originalen Koordinaten.
- Jedes CAD-Objekt der Typen „Punkt“, „Linie“ und „Polygon“ hat ein Handle vom Typ Integer, welches der „Geometrie-ID“ entspricht, die *Survey2GIS* jedem Objekt bei der Prozessierung zuweist. Die Handles werden in je einer Etikettierungsebene abgelegt.
- Zusätzlich existiert eine weitere Etikettierungsebene für jedes Attributfeld.
- Die Handles und Feldwerte für Punkte werden an den Punktkoordinaten etikettiert, diejenigen für Linien am mittleren Knotenpunkt und diejenigen für Polygone am Flächenschwerpunkt.
- Die meisten Ebenen sind im DXF zunächst als „unsichtbar“ gesetzt. Viele CAD-Programme ignorieren dies jedoch und schalten einfach jede Ebene auf „sichtbar“.
- Polygone werden zu Polylinien vereinfacht, es sei denn, dass 2D-Ausgabe (Option „-z“) gewählt wurde (s. [10.2.3 auf der nächsten Seite](#)).
- Topologische Qualitätsmerkmale wie „korrekte Polygonrichtung“ (s. [9.1.9 auf Seite 73](#)) und „Innengrenzen von Polygonen“ (s. [9.1.6 auf Seite 68](#)) werden für den DXF-Export nicht unterstützt.
- Attributdaten werden in eine separate, einfache Textdatei geschrieben (s. [10.2.4 auf Seite 79](#)). Die Werte im ersten Feld („geom\_id“) korrespondieren mit den Handles der CAD-Objekte.

Die Option zur DXF-Ausgabe existiert in *Survey2GIS* ausschließlich für den Zweck, CAD-basierte Arbeitsprozesse zu unterstützen. Bei der Verwendung von DXF für die Speicherung und/oder Prozessierung von Vermessungsdaten existieren signifikante Einschränkungen.

Innerhalb der von *Survey2GIS* erzeugten DXF-Datei wird *immer* der einfache Punkt („.“) als Dezimaltrennzeichen verwendet, egal wie die Option “--decimal-point=” gesetzt wurde oder was die Einstellungen des Betriebssystems sind.

### 10.2.1 DXF und Datenarchivierung

Zur langfristigen Archivierung und Nutzung von Daten im DXF muss gesagt werden, dass dieses Format (genau wie alle proprietären CAD-Datenformate) nicht unabhängig standardisiert ist. DXF ist eine eingetragene Marke der Firma AutoDesk, welche auch AutoCAD produziert. Nach wie vor entstehen neue Versionen von DXF (dies muss notwendigerweise so sein, da DXF eine sequentielle Abbildung der AutoCAD-internen, objekt-orientierten Datenbank darstellt), und die Spezifikation des Formats sieht sogar explizit die Möglichkeit von herstellerabhängigen Erweiterungen vor, die zwischen unterschiedlichen CAD-Programmen inkompatibel sein können. DXF eignet sich somit prinzipiell *nicht* als Format zur langfristigen Archivierung von Vermessungsdaten. Allerdings produziert *survey2gis* eine sehr einfache ASCII-Version von DXF, aus welcher sich die Daten mittels eines Texteditors herstellen lassen, sollte dies erforderlich sein.

Für die langfristige Archivierung und Vorhaltung sollten die Daten dennoch besser (neben der Speicherung als Rohdaten) in einem einfachen, gut dokumentierten GIS-Format wie Shapefile (s. 10.1) abgelegt werden.

### 10.2.2 DXF und topologische Daten

Der Import von Daten im Format DXF ins GIS für die weitere Bearbeitung und Analyse ist eine weitverbreitete Praxis, die aber nicht zu empfehlen ist. DXF kennt keine topologischen Strukturen (s. 9 auf Seite 62) und kennt keine einfache, für die GIS-Praxis geeignete Repräsentation von flächigen Objekten (Polygonen). Es gibt keinen „vernünftigen“ Weg, Eigenschaften wie „Löcher“ und Mehrteiligkeit in der DXF-Repräsentation von Polygonen unterzubringen. Dies wird unvermeidbar Probleme nach sich ziehen, wenn Geometrien mit Attributdaten verknüpft werden sollen. Für die GIS-basierte Datenverarbeitung sollten die Daten daher besser im Format Shapefile (s. 10.1) ausgegeben werden.

### 10.2.3 DXF und planare Polygone bzw. Polylinien

Anm.: Methoden zur Planarisierung von Polygonen sind derzeit in *Survey2GIS* noch nicht implementiert!

Wie oben erwähnt, kennt AutoCAD (und somit auch DXF) keine einfache Repräsentation für Flächenobjekte, d. h. Polygone. Stattdessen werden derartige Objekte als in Grenzlinien (Polylinien) eingesetzte Schraffuren modelliert. *Allerdings* sind Schraffuren in AutoCAD/DXF strikt als *planare* Objekte definiert. Dies bedeutet, dass alle Stützpunkte auf der Umgrenzung einer Schraffur (Polylinie) innerhalb derselben X/Y-Ebene im Raum liegen müssen. Diese Voraussetzung kann von realen Vermessungsdaten aufgrund der Unebenheit

des natürlichen Geländes aber kaum erfüllt werden. Stattdessen haben echte Vermessungsdaten i. d. R. eine signifikante Varianz in den Z-Koordinaten. Im Resultat ergeben sich die folgenden Möglichkeiten für die DXF-Ausgabe:

1. Alle Polygone werden als Polylinien ausgegeben (Vorgabeeinstellung).
2. Es kann 2D-Ausgabe gewählt werden, so dass alle Z-Koordinaten auf „0“ verflacht werden.
3. Die Stützpunkte der Polygone werden planarisiert, so dass sie alle in derselben X/Y-Ebene liegen *und* die neue Ebene parallel zur X/Y-Ebene des Weltkoordinatensystems liegt. Letzteres ist die Voraussetzung dafür, dass AutoCAD Schraffuren zeichnet, wenn eine lotrechte Draufsicht auf die Objekte gewählt wird.

Streng betrachtet, sollten auch im GIS Polygone planarisiert werden. In der Praxis ignorieren die meisten GIS jedoch Variationen in den Z-Koordinaten von Polygonstützpunkten, da die Datensicht ohnehin auf eine lotrechte Draufsicht beschränkt ist. Mehr Informationen zu diesem Thema finden sich unter ?? auf Seite ??.

#### 10.2.4 DXF und Attributdaten

Es gibt keine perfekte Methode, um Attributdatensätze 1:1 mit den Zeichnungs-Objekten eines CAD-Projekts zu verbinden. Im Falle von *Survey2GIS* besteht das einfache Vorgehen darin, die Attributdaten in eine zusätzliche Textdatei (Endung „.txt“) zu schreiben, welche einen Datensatz pro Zeile enthält.

Ein Auszug aus einer solchen Datei könnte etwa wie folgt aussehen:

```
geom_id;konst1;idx;planum;typ;zusatz;nummer;koorx;koory;koorz
0;123.450000;67;1;"GR";"W";0;3513037.664000;5279881.392000;399.563000
1;123.450000;1;1;"L0";"W";0;3513041.874000;5279875.482000;399.025000 2;123.450000;15;1;"L0";
...
```

Zur Formatierung der Datei gibt es einige wenige Dinge zu beachten:

- Die Attributfelder sind durch “;” (Semikolon) getrennt.
- Die erste Zeile enthält die Namen der Felder.
- Die Inhalte von Textfeldern sind in Anführungszeichen (,,) eingeschlossen.

Beim ersten Feld handelt es sich stets um das Integer-Feld *geom\_id*. Es enthält einen Primärschlüssel, der mit dem jeweiligen Handle des CAD-Objekts übereinstimmt, auf den sich der Datensatz bezieht.

Für „rohe“ Punktmessungen werden keine Attributdaten geschrieben.

Innerhalb der von *Survey2GIS* erzeugten TXT-Datei mit Attributen wird *immer* der einfache Punkt („.“) als Dezimaltrennzeichen in numerischen Attributen verwendet, egal wie die Option “--decimal-point=” gesetzt wurde oder was die Einstellungen des Betriebssystems sind.

### 10.2.5 „Null“-Werte in DXF-Attributen

Sofern der Benutzer in der Parserdefinition die Option „no\_data“ (s. 4.3) nicht gesetzt hat, gilt das Folgende: Die Abbildung von Attributwerten, die „Null“ (keine Daten) sind, unterscheidet nach dem Datentyp: Textfelder werden als leere Felder gespeichert, numerische Felder als „0“.

### 10.2.6 DXF und Etiketten

Etiketten (s. 5) werden als separate Ebene innerhalb der DXF-Datei gespeichert.

## 10.3 GeoJSON

Das Ausgabeformat „GeoJSON“ besteht aus einer einzigen Textdatei, welche ein *JavaScript*-Objekt enthält, das wiederum sämtliche von *Survey2GIS* produzierten Geometrien und Attribute enthält. GeoJSON ist kein sehr effizientes Format und nicht für den GIS-Endanwender gedacht, sondern für den verlustfreien Datenaustausch mit Datenbanken und WebGIS-Anwendungen.

Die von *Survey2GIS* erzeugte GeoJSON-Datei setzt die offiziellen Vorgaben von <http://geojson.org> recht genau um (beachten Sie aber die Hinweise zu KBS, unten!) und verwendet zudem eine leserliche Formattierung.

Achtung: Ähnlich wie KML (s. 10.4), wurde auch GeoJSON für Daten mit großräumiger geographischer Abdeckung entwickelt, die üblicherweise in Längen- und Breitengraden vorliegen. Tatsächlich wird seit der Version RFC 7946 der GeoJSON-Spezifikation *nur* noch ein einziges KBS mit Längen-/Breitengraden und WGS-84-Datum (entsprechend EPSG 4326) unterstützt. Da GeoJSON, wie erwähnt, für die Integration von *Survey2GIS* in eine automatisierte Verarbeitungskette nützlich ist, erlaubt *Survey2GIS* es dennoch, Daten mit anderen KBS als GeoJSON auszugeben, es sei denn, es läuft im Modus „strikt“ (s. Option „--strict“ in 2). Beachten Sie jedoch, dass es nicht möglich ist, KBS-Informationen standardkonform in ein GeoJSON-Objekt zu schreiben.

Geometrien werden nach Typ sortiert ausgegeben (um das Potenzial von Überdeckungen bei der Darstellung im GIS zu minimieren: zuerst „rohe“ Stützpunkte, dann Punkte, Linien und schließlich Polygone) und in eine durch Kommata getrennte Liste geschrieben, die einfache, leserlich formatierte Einträge enthält.

Geometrien werden jeweils im Abschnitt „geometry“ gespeichert, Attribute in „properties“:

```
{ "type": "FeatureCollection",
  "features": [
    { "type": "Feature", "id": 0,
      "geometry": {
        "type": "Point",
        "coordinates": [3513040.585000, 5279881.854000, 399.102000]
      },
```



```

    "properties": {
      "geom_id": 0,
      "const1": 123.450000,
      "idx": 256,
      "level": 1,
      "type": "Gold",
      "aux": "FZ",
      "_id": 0,
      "coorx": 3513040.585000,
      "coory": 5279881.854000,
      "coorz": 399.102000
    }
  },
  ...

```

Das Feld „geom\_id” wird stets automatisch erzeugt und enthält einen innerhalb des GeoJSON-Objekts eindeutigen Primärschlüssel.

Zu beachten ist, dass es keine expliziten Feldtypen für die Attribute („properties”) gibt. Ob ein Feld vom Typ „text” (z. B. „type” im Beispiel oben), „double” (z. B. „coorx”) oder „integer” (z. B. „level”) ist, entscheidet sich lediglich anhand der Notation des Feldwerts.

Innerhalb des von *Survey2GIS* erzeugten GeoJSON-Objekts wird *immer* der einfache Punkt („.”) als Dezimaltrennzeichen verwendet, egal wie die Option “--decimal-point” gesetzt wurde oder was die Einstellungen des Betriebssystems sind.

### 10.3.1 Geometrietypen in GeoJSON

Das von *Survey2GIS* erzeugte GeoJSON verwendet stark typisierte Geometrien. Dies bedeutet, dass jeweils der einfachste Typ verwendet wird, der ausreicht um eine Geometrie abzubilden:

- Punkte werden als GeoJSON-Typ “Point” gespeichert.
- Polygone werden als GeoJSON-Typ “Polygon” gespeichert, wenn sie einteilig sind, ansonsten als “MultiPolygon”.
- Linien werden als GeoJSON-Typ “LineString” gespeichert, wenn die einteilig sind, ansonsten als Typ “MultiLineString”.

Hinweis: Da die *Survey2GIS*-Typen „Punkt” und „Rohdaten” auf denselben GeoJSON-Typ „Point” abgebildet werden, kann man im GeoJSON-Objekt nicht länger zwischen ihnen unterscheiden! Sollte dies ein Problem im avisierten Arbeitsfluss darstellen, dann lässt sich eine Geometrieauswahl (s. 6) verwenden, um separate Ausgabedateien zu erzeugen.

### 10.3.2 Primärschlüssel in GeoJSON (Feld „id“)

Der GeoJSON-Standard verwendet ein Feld namens „id“ als Primärschlüssel. Sollten die Eingabedaten bereits ein Feld gleichen Namens enthalten, so kann es nicht in die „properties“ des GeoJSON-Objekts geschrieben werden, ohne mit dem Primärschlüssel zu kollidieren.

In diesem Fall wird *Survey2GIS* versuchen das Problem zu umgehen, indem es das benutzerdefinierte Feld „id“ in „\_id“ (mit Unterstrich) umbenennt. Sollte ein Feld namens „\_id“ ebenfalls bereits existieren, dann bricht das Programm mit einer Fehlermeldung ab.

Genau genommen sollte sich diese Problem überhaupt nicht stellen, da die GeoJSON-Spezifikation festlegt, dass der Primärschlüssel „id“ im „type“-Abschnitt des Objekts deklariert wird und nicht in „properties“. Allerdings scheinen nicht alle GIS (insbesondere diejenigen, welche für den GeoJSON-Import auf <http://www.gdal.org> zurückgreifen) diese Unterscheidung zu respektieren (*Survey2GIS* tut dies allerdings und platziert „id“ im Abschnitt „type“).

### 10.3.3 „Null“-Werte in GeoJSON-Objekten

Sofern der Benutzer in der Parserdefinition die Option „no\_data“ (s. 4.3) nicht gesetzt hat, gilt das Folgende: Die Abbildung von Attributwerten, die „Null“ (keine Daten) sind, unterscheidet nach dem Datentyp: Textfelder werden als leere Felder gespeichert, numerische Felder als „0“ (Ganzzahlen) oder „0.0“ (Realzahlen).

### 10.3.4 Etiketten in GeoJSON

Etiketten (Label) werden nur eingeschränkt im GeoJSON-Export berücksichtigt.

Die Eigenschaften der Etiketten (Platzierung, Zeichensatz, etc.: s. 5 werden für alle etikettierten Geometrien als Teil der „properties“ gespeichert (s. oben). Etiketten werden jedoch *nicht* als separate Punkte gespeichert (im Gegensatz z. B. zum Shapefile-Export: s. 10.1).

Da GeoJSON nur einen „properties“-Abschnitt je Geometrie erlaubt, kann auch nur ein X-Y-Koordinatenpaar für die Platzierung exportiert werden. Im Falle von mehrteiligen Geometrien repräsentiert dieses das Etikett für den *ersten* Teil.

## 10.4 Keyhole Markup Language (Google KML)

Die „Keyhole Markup Language“ (Google KML) ist ein XML-basiertes Format, das für die Verwendung mit Google Earth (<https://www.google.com/earth/>) konzipiert wurde. Dementsprechend besteht seine einzige Funktion darin, Daten für die flexible 3D-Visualisierung in Google Earth vorzuhalten. KML opfert Genauigkeit für Geschwindigkeit und Bequemlichkeit und eignet sich somit nicht für die weitere Datenverarbeitung oder gar -archivierung.

Hinweis: KML sieht *ausschließlich* Koordinaten in Längen- und Breitengrade (in Dezimalschreibweise) vor. Die Verwendung von projizierten Koordinaten (etwa UTM) ist nicht vorgesehen. Außerdem wird *davon ausgegangen*, dass es sich um ein WGS-84-Datum handelt (was aber eigentlich nicht der Fall ist: s. [8.1.4!](#)). Der Datenexport nach KML steht demnach nur zur Verfügung, sofern die Eingabedaten entweder in Längen- und Breitengraden vorliegen oder als solche reprojiziert werden können (s. [8](#)).

Die von *Survey2GIS* erzeugte KML-Datei ist auf Übersichtlichkeit und Handhabbarkeit größerer *Vermessungsdatensätze* in Google Earth optimiert.

KML verwendet „Placemarks“ um geographische Objekte (Punkte, Linien und Polygone mit zugehörigen Attributdaten) abzubilden. Diese können wiederum in Ordnern (KML-Element „Folder“) gruppiert werden, um größere Mengen von Placemarks gleichzeitig ein- und ausblenden zu können. Die von *Survey2GIS* erzeugte KML-Datei enthält folgende Ordner:

1. „Points“ (Punkte)
2. „Lines“ (Linien)
3. „Polygons“ (Polygone)
4. „Vertices“ (Stützpunkte/Rohdaten)
5. „Labels“ (Etiketten)

Die Inhalte obiger Ordner sollten größtenteils selbsterklärend sein. Der Ordner „Vertices“ enthält die Rohdaten der Messungen (d. h. jeden einzelnen Stützpunkt von Linien und Polygonen) und wird nur erzeugt, wenn die Option „--raw-data“ (s. [2](#)) verwendet wird; die in diesem Ordner enthaltenen Stützpunkte werden per Voreinstellung ausgeblendet. Der Ordner „Labels“ enthält (sofern vom Benutzer gewünscht: s. [5](#)) zusätzliche Etiketten. Die Reihenfolge der Anzeige von Punkten, Linien und Polygonen wird von Google Earth entschieden.

Die Speicherung von Attributdaten nutzt das KML-Element „Extended Data“. Dementsprechend legt *Survey2GIS* ein Attributschema in der KML-Datei an und speichert eine genaue Abbildung von Typen und Inhalten aller Attributfelder. Alle Attributfelder und ihre Typen werden angezeigt, wenn der Benutzer ein „Placemark“ in Google Earth anklickt.

Über Google Earth wird berichtet, dass es hin und wieder Probleme bei der Darstellungen komplexerer Polygone hat. Der Grund hierfür ist vermutlich in der mangelnden Genauigkeit der Triangulation von Polygonen für die 3D-Darstellung zu suchen, welche stark im Hinblick auf Geschwindigkeit optimiert ist.

Innerhalb der von *Survey2GIS* erzeugten KML-Datei wird *immer* der einfache Punkt („.“) als Dezimaltrennzeichen in Koordinaten und Attributwerten verwendet, egal wie die Option “--decimal-point=” gesetzt wurde oder was die Einstellungen des Betriebssystems sind.

### 10.4.1 Geometrietypen in KML

Die Koordinaten von Punkten und Stützpunkten werden in der Reihenfolge Längengrad (als Dezimalgrad), Breitengrad (als Dezimalgrad) und Höhe (immer in Metern) in die KML-Datei geschrieben. Die durch KML-Placemarks abgebildeten Geometrietypen sind:

- Polygon (Polygone)
- LineString (Linien)
- Point (Punkte)

Mehrteilige Geometrien werden abgebildet, indem mehr als ein „Polygon“- oder „LineString“-Element pro „Placemark“-Element gespeichert wird.

Google Earth geht davon aus, dass die Stützpunkte von Polygonen (sowohl Außengrenzen als auch Löcher) *gegen den Uhrzeigersinn* in die KML-Datei geschrieben werden (Google Earth benötigt diese Orientierung, um nach der 3D-Triangulation die korrekte Ausrichtung der Maschenelemente für die künstliche Beleuchtung zu bestimmen).

Höhenwerte (Z-Daten) können von Google Earth auf verschiedene Art interpretiert werden, aber *Survey2GIS* setzt das KML-Element `<altitudeMode>` stets auf „absolute“, so dass Z-Daten als „Höhe über dem Meeresspiegel“ gelesen werden (Hinweis: Durch lokale Genauigkeitsabweichungen zwischen dem digitalen Geländemodell von Google Earth und den Vermessungsdaten kann es zu Darstellungsfehlern kommen).

### 10.4.2 Label layers in KML

Die KML-Datei enthält bei Bedarf einen zusätzlichen Ordner „Labels“ mit allen vom Benutzer definierten Etiketten (Details s. 5).

## 11 Umgang mit Sonderzeichen

Bei der Benutzung von *Survey2GIS* existieren prinzipiell *nur die folgenden* Kontexte für die Verwendung von Sonderzeichen (d. h. Zeichen die nicht im ASCII-Code enthalten sind), wie z. B. deutschen Umlauten:

1. Programmmeldungen und -oberfläche,
2. Datei- und Ordernamen,
3. Auswahlkommandos, die sich auf Ersetzungsausdrücke mit Sonderzeichen beziehen (s. 6) ...
4. ... und einige wenige Inhalte von Parser-Schemata (s. 4):
  - (a) Der Inhalt von „info“ im Abschnitt „[Parser]“,
  - (b) der Inhalt von „name“ im Abschnitt „[Parser]“,
  - (c) der Inhalt von „info“ im Abschnitt „[Field]“ ...
  - (d) ... und auf „@“ folgende Ersetzungsausdrücke im Abschnitt „[Field]“.

In allen anderen Zusammenhängen (Optionsnamen für Kommandozeile und Parserdatei, Feldnamen für Attributdaten, etc.) ist die Verwendung von Sonderzeichen nicht möglich!

Intern verwendet *Survey2GIS* die Unicode-Version „UTF-8“ für die Kodierung aller internationalen Sonderzeichen. Alle ASCII-Zeichen verfügen in UTF-8 über exakt dieselben Codes, so dass eine Konvertierung von ASCII nach UTF-8 nicht notwendig ist. Reine ASCII-Daten können von *Survey2GIS* daher stets problemlos verarbeitet werden.

UTF-8 ist u. a. die im Internet und auf zahlreichen Betriebssystemen am häufigsten anzutreffende Zeichenkodierung (sollte ein Linux- oder macOS-System nicht auf UTF-8 eingestellt sein, so lässt sich dies im Nachhinein ändern).

Eine Ausnahme ist das Betriebssystem Windows, welches aus historischen Gründen mit einer Vielzahl von „Codepages“ ausgestattet ist (die Windows-Codepage, welche UTF-8 am nächsten kommt, hat die Nummer 65001). Seit Version 1.3.1 ist *Survey2GIS* in der Lage, auf Windows-Betriebssystemen mit der dort für Dateinamen verwendeten Zeichenkodierung korrekt umzugehen. Um auf der Windows-Kommandozeile (*cmd.exe*) mit UTF-8-kodierten Sonderzeichen arbeiten zu können, empfiehlt es sich, für *cmd.exe* eine Schriftart zu setzen, die Unicode-Zeichen enthält. Gute Resultate lassen sich z. B. mit „Lucida Console“ erzielen.

Sollen Sonderzeichen in einem Parser-Schema (wo möglich: s. o.) verwendet werden, so muss die Textdatei, welche das Schema enthält, mit einem Programm geschrieben werden, welches UTF-8 kodierte Textdateien erzeugen kann.

Werden nach der Prozessierung durch *Survey2GIS* Ersetzungsausdrücke mit Sonderzeichen in Attributfeldern gespeichert, so verwenden diese ebenfalls die Kodierung UTF-8. Somit werden bspw. Sonderzeichen in den

DBF-Tabellen von Shapefiles in einem GIS nur dann korrekt angezeigt, wenn letzteres ebenfalls UTF-8 für deren Kodierung verwendet.

Die Verwendung von Sonderzeichen (ebenso wie Leerzeichen) in Datei- und Ordernamen sollte stets vermieden werden, da sich hieraus zahlreiche Probleme (im Zusammenhang mit anderen Programmen oder beim Datenaustausch über unterschiedliche Betriebssysteme hinweg) ergeben können.

## 12 Hinweise zur Vermessungspraxis

Die Qualität der Ausgabedaten hängt direkt von der Stringenz der Vermessungspraxis ab. Die in *Survey2GIS* eingebauten Korrekturen können im Feld entstehende Probleme nur zum Teil bereinigen. Daher hier einige wichtige Hinweise:

Die Reihenfolge der Messungen, d. h. die Reihenfolge der Datensätze in der/den Eingabedatei(en), ist von großer Wichtigkeit:

1. Linien erhalten durch die Reihenfolge der Stützpunkte ihre Richtung.
2. Bei Polygonen sollten Außengrenzen im Uhrzeigersinn, Innengrenzen gegen den Uhrzeigersinn eingemessen werden. Zwar ist *Survey2GIS* in der Lage, falsch herum eingemessene Polygongrenzen in Shapedateien zu korrigieren, für andere Ausgabeformate funktioniert dies aber nicht (s. auch 9.1.9).
3. Eingabedateien sollten in der Reihenfolge angegeben werden, in der sie bei der Vermessung produziert wurden.
4. Topologische Funktionen wie das Snapping oder Ausstanzen von Löchern in überlappenden Polygonen (s. 9.1) gehen davon aus, dass größere „Basisobjekte“ vor kleineren Objekten eingemessen wurden.

Die wichtigste Regel ist das *hierarchische* Einmessen: erst die Basislinien und Grundobjekte, dann daran anstoßende kleinere Objekte, Einschlüsse, Löcher, etc.

Bei der Einmessung von Polygonen ist es nicht notwendig, den ersten Stützpunkt zum Abschluss noch einmal einzumessen. Insofern das Ausgabeformat dies bedingt, wird *Survey2GIS* automatisch einen Stützpunkt einfügen, welcher dem ersten exakt entspricht, um das Polygon zu schließen.

Eine wichtige Überlegung betrifft die Kodierung von Punktmessungen. In einigen Parsermodi („End“ und „Min“) ist es möglich, den Geometriemarker für Punkte nicht zu spezifizieren (d. h. keine Angabe für „geom\_tag\_point“ zu machen: s. 4.2). In diesem Fall versucht *Survey2GIS* einzelne Punktmessungen automatisch von Knotenpunkten für Linien und Polygone zu trennen. Dies kann jedoch nur dann zuverlässig funktionieren, wenn der Primärschlüssel-Wert (d. h. der Inhalt des Felds „key\_field“) *über alle Geometrien eindeutig* ist. Sollte dies nicht der Fall sein, dann kann es zu folgenden Problemen kommen:

- Parsermodus „End“: *Wenn* ein Punkt direkt *vor* einem Linien- oder Polygonobjekt steht *und wenn* dieser Punkt denselben Primärschlüssel-Wert hat wie die Stützpunkte des darauf folgende Linien- oder Polygonobjekts, dann wird er als falscher Knoten mit in das Objekt einbezogen.
- Parsermodus „Min“: *Wenn* ein Punkt direkt *nach* einem Linien- oder Polygonobjekt steht *und wenn* dieser Punkt denselben Primärschlüssel-Wert hat wie die Stützpunkte des vorhergehende Linien- oder Polygonobjekts, dann wird er als falscher Knoten mit in das Objekt einbezogen.

Um solche Probleme zu vermeiden, kommen zwei Möglichkeiten in Betracht:

1. Es wird eine Angabe für „geom\_tag\_point“ gemacht und alle Punktmessungen werden konsequent damit kodiert.
2. Man vergibt Nummernblöcke für die Primärschlüssel der unterschiedlichen Geometrietypen, die nicht überlappen, z. B. 0-9999 für Punkte, 10000-19999 für Linien, 20000 bis 29999 für Polygone.



## A Bekannte Probleme

Die folgenden Probleme sind in der aktuellen Version von *Survey2GIS* bekannt:

- Beim Start per *cmd.exe* Unter Windows kann *Survey2GIS* nicht gleichzeitig im GUI-Modus benutzt werden und Text auf der Konsole ausgeben. Stattdessen sind zwei unterschiedliche Versionen von „survey2gis.exe“ notwendig, eine zur Benutzung von der Kommandozeile (*cmd.exe*) und eine für die Benutzung im GUI-Modus.
- Der Austausch von Dateien mit gespeicherten Einstellungen (Endung „.s2g“) unterliegt einer Beschränkung hinsichtlich der Spracheinstellungen: Numerische Einstellungen werden im aktuell gesetzten Zahlenformat gespeichert. Werden diese anschließend in eine Instanz von *Survey2GIS* geladen, die andere Spracheinstellungen verwendet, kann es vorkommen, dass die Einstellungen nicht korrekt verarbeitet werden. Es ist in solchen Fällen ratsam, nach dem Laden der Einstellungen die Zahlenformate aller numerischen Optionen zu prüfen und ggfs. zu korrigieren.
- Die Unterstützung für Eingabedaten mit Längen- und Breitengraden ist begrenzt. Insbesondere werden einige Funktionen zur topologischen Bereinigung unzufriedenstellende Ergebnisse liefern oder gar nicht zur Verfügung stehen. Das Programm gibt ggfs. entsprechende Warnungen aus, wenn derartige Daten verarbeitet werden.
- Punkte, welche dieselben Werte in „key\_field“ (Schlüselfeld) besitzen wie darauf folgende Linien oder Polygone, werden mit den Folgegeometrien verbunden, so dass z. B. eine Auswahl nicht korrekt funktioniert. Dieses Problem tritt nur auf, wenn die Geometriemarker im Schlüsselfeld gespeichert werden (d. h. „tag\_field“ und „key\_field“ sind identisch: s. 4.2), und Schlüsselwerte verwendet werden, die nicht global eindeutig sind (d. h. eine Punktgeometrie kann denselben Schlüsselwert haben wie eine Linien- oder Polygoneometrie). Um dieses Problem zu vermeiden, sollten „tag\_field“ und „key\_field“ unterschiedliche Felder sein.

## B Lizenz

©2015-2019 *survey-tools.org*-Entwicklergruppe (<http://www.survey-tools.org>).

Diese Software (*Survey2GIS*) steht unter der *GNU General Public License* (Version 2).

Den vollständigen Originaltext (englisch) der Lizenz finden Sie in der Textdatei „LICENSE“, die mit dieser Software ausgeliefert wurde.

Eine deutsche Übersetzung finden Sie im Internet auf <https://www.gnu.org/licenses/old-licenses/gpl-2.0.de.html>.