

translate.py - Text-Extraktion und -Übersetzung für DXF-Dateien

Übersicht

translate.py ist ein Python-Tool zur Extraktion, Filterung und Übersetzung von Texten aus DXF/DWG-Dateien. Es unterstützt mehrere Arbeitsweisen und ermöglicht die Erstellung mehrsprachiger technischer Zeichnungen für den internationalen Einsatz.

Quickstart - Haupt-Workflow für DXF-Übersetzung

Dieser Workflow beschreibt die **empfohlene Vorgehensweise** für die Übersetzung technischer Zeichnungen.

Schritt 1: Text-Extraktion aus DXF

Ziehen Sie die DXF-Datei per **Drag & Drop** auf die Desktop-Verknüpfung **tr_dxf2txt**:



Was passiert:

- Alle TEXT, MTEXT und Block-Attribute (TEXT-D) werden aus der DXF extrahiert
- Zwei Dateien werden im **work/** Ordner erstellt:
 - **<dateiname>_texts.json** - JSON-Datei mit allen extrahierten Texten
 - **<dateiname>_texts.txt** - Textdatei mit allen extrahierten Texten

Beispiel:

```
Eingabe:  Projekt_Anlage_Rev3.dxf
Ausgabe:  work/Projekt_Anlage_Rev3_texts.json
          work/Projekt_Anlage_Rev3_texts.txt
```

Schritt 2: Automatische Übersetzung mit Sprach-Config

Ziehen Sie die erzeugte **JSON-Datei** (**<dateiname>_texts.json**) per **Drag & Drop** auf die gewünschte Sprach-Verknüpfung:

- **tr2dxf_CS** (Tschechisches Icon) → Tschechische Übersetzung
- **tr2dxf_EN** (Englisch-Icon) → Englische Übersetzung
- **tr2dxf_FR** (Französisch-Icon) → Französische Übersetzung
- **tr2dxf_ES** (Spanisch-Icon) → Spanische Übersetzung

- **tr2dxf_IT** (Italienisch-Icon) → Italienische Übersetzung



Was passiert:

- Die JSON-Datei wird mit der aktuellen Sprachkonfiguration (z.B. **CS.cfg**) re-übersetzt
- Alle konfigurierten Begriffe werden automatisch übersetzt
- Der **[translate]** Block in der JSON wird mit Übersetzungen gefüllt
- Nicht-übersetzte Begriffe bleiben im **untranslated** Block

JSON-Struktur nach diesem Schritt:

```
{
  "[CS]": {
    "translations": [
      {"source": "Emergency stop", "target": "Nouzové zastavení"},
      {"source": "Power supply", "target": "Napájení"}
    ],
    "untranslated": [
      {"source": "Custom text XY", "target": ""}
    ]
  },
  "ignored": ["X=123.45", "LAYER=0"]
}
```

Schritt 3: Sichtkontrolle und manuelle Ergänzung

Öffnen Sie die **_texts.json** Datei mit einem **Texteditor** oder **JSON-Editor** und prüfen Sie:

1. **Translations Block:** Sind alle Übersetzungen korrekt?
2. **Untranslated Block:** Welche Begriffe fehlen noch?

Bei Bedarf:

- Fügen Sie fehlende Übersetzungen **manuell** in die JSON ein:

```
{"source": "Custom text XY", "target": "Vlastní text XY"}
```

- **Oder** ergänzen Sie die Sprach-Config (**translation/CS.cfg**) und wiederholen Sie Schritt 2

Hinweis zur .txt Datei: Die parallel erzeugte **.txt** Datei hat folgendes Format:

```
--- translations:

Emergency stop -> Nouzové zastavení
Power supply -> Napájení

--- untranslated:

Custom text XY

--- ignored:

X=123.45
LAYER=0
```

Die **.txt** Datei eignet sich gut für einen **schnellen Überblick**, während die **.json** Datei für die tatsächliche DXF-Übersetzung verwendet wird.

Schritt 4: DXF-Datei übersetzen

Ziehen Sie die **Original-DXF-Datei** per **Drag & Drop** auf die gewünschte **Sprach-Verknüpfung** (z.B. **tr2dxf_CS**).

Voraussetzung:

- Die **<dateiname>_texts.json** muss im **work/** Ordner vorhanden sein
- Die JSON sollte möglichst vollständige Übersetzungen enthalten

Was passiert:

- Eine neue DXF-Datei wird erstellt: **<dateiname>_CS.dxf** (oder EN, FR, etc.)
- Alle TEXT, MTEXT und Block-Attribute werden übersetzt
- Die Übersetzungen werden in die entsprechenden Ziel-Attribute geschrieben:
 - TEXT-D → TEXT-CS (bei Tschechisch)
 - TEXT-D → TEXT-E (bei Englisch)
 - TEXT-D → TEXT-ES (bei Spanisch)
 - etc.
- Das Original-Attribut TEXT-D bleibt **unverändert erhalten**

Beispiel:

```
Eingabe:  Projekt_Anlage_Rev3.dxf (+ work/Projekt_Anlage_Rev3_texts.json)
Ausgabe:  work/Projekt_Anlage_Rev3_CS.dxf
```

Zusammenfassung des Workflows

```
1. DXF → [tr_dxf2txt] → _texts.json + _texts.txt
      ↓
```

- ```
2. _texts.json → [tr2dxf_CS] → _texts.json (mit Übersetzungen)
 ↓
3. Manuelle Prüfung/Ergänzung der JSON
 ↓
4. DXF → [tr2dxf_CS] → DXF_CS.dxf (übersetzte Zeichnung)
```

---

## Konfiguration für neue Sprachen

Wenn Sie eine neue Sprache konfigurieren oder bestehende Übersetzungen erweitern möchten, folgen Sie dieser Anleitung.

### Aufbau der Sprach-Konfigurationsdateien

Sprach-Configs befinden sich im Verzeichnis `translation/` (z.B. `CS.cfg`, `EN.cfg`, `FR.cfg`).

Jede Config-Datei hat **vier Sektionen** in festgelegter Prioritätsreihenfolge:

1. **[multi]** - Phrasen mit 4 oder mehr Wörtern
2. **[trigramme]** - Phrasen mit genau 3 Wörtern
3. **[bigramme]** - Phrasen mit genau 2 Wörtern
4. **[single]** - Einzelwörter

**Wichtig:** Die Übersetzung erfolgt in dieser Reihenfolge. Längere Phrasen werden zuerst ersetzt, danach kürzere.

### Empfohlene Vorgehensweise beim Konfigurieren

#### 1. Erste Extraktion durchführen

Führen Sie zunächst eine Text-Extraktion **ohne Übersetzung** durch:

```
bin\translate.bat --filename projekt.dxf --extract --outname projekt_texts.json
```

Dies erzeugt `projekt_texts.json` und `projekt_texts.txt` mit allen extrahierten Texten.

#### 2. Zusammenhängende Phrasen identifizieren

Öffnen Sie die `.txt` Datei und suchen Sie nach zusammenhängenden Phrasen:

##### Beispiele für zusammenhängende Phrasen:

- "Main distribution panel" (3 Wörter)
- "Emergency stop button" (3 Wörter)
- "Power supply unit" (3 Wörter)
- "Safety circuit monitoring relay" (4 Wörter)

#### 3. Konfiguration ausfüllen - Sektion [multi]

Beginnen Sie mit den **längsten Phrasen** (4+ Wörter) in der **[multi]** Sektion:

**[multi]**

Safety circuit monitoring relay = Relé sledování bezpečnostního obvodu

Main circuit breaker control unit = Hlavní jistič řízení jednotky

Emergency stop button reset function = Nouzové tlačítko funkce reset

**Format:** Original = Übersetzung

#### 4. Konfiguration ausfüllen - Sektion **[trigramme]**

Fügen Sie **3-Wort-Phrasen** in die **[trigramme]** Sektion ein:

**[trigramme]**

Main distribution panel = Hlavní rozváděč

Power supply unit = Napájecí zdroj

Emergency stop button = Nouzové tlačítko zastavení

Safety light curtain = Bezpečnostní světelná závěsa

Motor protection switch = Ochrana motoru spínač

#### 5. Konfiguration ausfüllen - Sektion **[bigramme]**

Fügen Sie **2-Wort-Phrasen** in die **[bigramme]** Sektion ein:

**[bigramme]**

Emergency stop = Nouzové zastavení

Power supply = Napájení

Control panel = Ovládací panel

Safety relay = Bezpečnostní relé

Main switch = Hlavní spínač

Motor protection = Ochrana motoru

#### 6. Konfiguration ausfüllen - Sektion **[single]**

Fügen Sie **Einzelwörter** in die **[single]** Sektion ein:

**[single]**

Weight = Hmotnost

Cable = Kabel

Emergency = Nouzový

Stop = Zastavení

Power = Napájení

Control = Ovládání

Safety = Bezpečnost

Motor = Motor

```
Switch = Spínač
Panel = Panel
```

**Hinweis:** Einzelwörter werden auch innerhalb von Wörtern mit Sonderzeichen erkannt:

- "Weight:" → "Hmotnost:"
- "(Motor)" → "(Motor)"
- "\\fArial|b0;Switch" → "\\fArial|b0;Spínač"

Warum diese Reihenfolge?

Die Reihenfolge **Multi** → **Trigramme** → **Bigramme** → **Single** ist entscheidend für korrekte Übersetzungen:

**Beispiel:**

- Text: "Emergency stop button"
- Falsche Reihenfolge (single zuerst):
  - "Emergency" → "Nouzový"
  - "stop" → "zastavení"
  - Ergebnis: "Nouzový zastavení button" ✗
- Richtige Reihenfolge (trigramme zuerst):
  - "Emergency stop button" → "Nouzové tlačítko zastavení" ✓

Testen der Konfiguration

Nach dem Ausfüllen der Config testen Sie die Übersetzung:

```
Extraktion mit automatischer Übersetzung
bin\translate.bat --filename projekt.dxf --extract --translate CS --outname
test_cs.json

Prüfen Sie die JSON-Datei:
- "translations" Block: erfolgreich übersetzte Begriffe
- "untranslated" Block: fehlende Übersetzungen
```

Ergänzen Sie die Config basierend auf dem **untranslated** Block und wiederholen Sie den Test.

Re-Übersetzung einer bestehenden JSON

Wenn Sie die Sprach-Config erweitert haben, können Sie eine bestehende JSON-Datei **re-übersetzen**:

```
Manuelle Kommandozeile
bin\translate.bat --retranslate-json projekt_texts.json --translate CS

ODER: Drag & Drop der JSON auf die Sprach-Verknüpfung (tr2dxf_CS)
```

Dies aktualisiert alle Übersetzungen basierend auf der aktuellen Config.

## Hauptfunktionen

- **Text-Extraktion:** Extrahiert TEXT, MTEXT und Block-Attribute aus DXF-Dateien
- **Text-Filterung:** Filtert Texte basierend auf konfigurierbaren Mustern (Wildcards und Regex)
- **Automatische Übersetzung:** Übersetzt Texte mittels konfigurierbarer Wörterbücher (Multi-Phrasen, Trigramme, Bigramme, Einzelwörter)
- **DXF-zu-DXF Übersetzung:** Erstellt übersetzte DXF-Dateien mit mehrsprachigen Attributen
- **Mehrsprachige Übersetzung:** Unterstützt gleichzeitige Übersetzung in mehrere Sprachen
- **Export:** Exportiert Texte in verschiedene Formate (Excel, JSON, Text)

## Konfigurationsdateien

translator.cfg

Hauptkonfigurationsdatei im `cfg/` Verzeichnis:

```
[symbol_attribute]
Zuordnung von Sprachcodes zu Block-Attributnamen
DE = TEXT-D
EN = TEXT-E
ES = TEXT-ES
CS = TEXT-CS
FR = TEXT-F

[symbol_layer]
Zuordnung von Sprachcodes zu Layer-Namen
DE = 0-0_TEXT-D
EN = 0-0_TEXT-E
ES = 0-0_TEXT-ES
CS = 0-0_TEXT-CS
FR = 0-0_TEXT-F

[ignore_pattern_wildcard]
Wildcard-Muster für zu ignorierende Texte
coord_x = X=*
coord_y = Y=*
coord_z = Z=*
... weitere Muster

[ignore_pattern_regex]
Regex-Muster für zu ignorierende Texte
coord_arrows = ^\+\d{1,3}(?:[.,]\d{3})\s*(?:<<<|>>>)\s*\+\d{1,3}(?:[.,]\d{3})$
ip_address = ^.*;(\d{1,3}\.){3}\d{1,3}$
... weitere Muster
```

## Zweck der Sektionen:

- **[symbol\_attribute]**: Mapping von Sprach-Codes zu DXF-Block-Attributnamen
- **[symbol\_layer]**: Mapping von Sprach-Codes zu DXF-Layer-Namen
- **[ignore\_pattern\_wildcard]**: Einfache Muster mit `*` und `?` für zu ignorierende Texte
- **[ignore\_pattern\_regex]**: Reguläre Ausdrücke für komplexere Filter-Muster

## Übersetzungs-Configs (Beispiel: CS.cfg)

Sprachspezifische Konfigurationsdateien im [translation/](#) Verzeichnis:

```
[multi]
Phrasen mit 4+ Wörtern
Safety circuit monitoring relay = Relé sledování bezpečnostního obvodu
Main circuit breaker control unit = Hlavní jistič řízení jednotky

[trigramme]
3-Wort-Phrasen
Main distribution panel = Hlavní rozváděč
Power supply unit = Napájecí zdroj
Emergency stop button = Nouzové tlačítko zastavení

[bigramme]
2-Wort-Phrasen
Emergency stop = Nouzové zastavení
Power supply = Napájení
Control panel = Ovládací panel

[single]
Einzelwörter
Weight = Hmotnost
Cable = Kabel
Emergency = Nouzový
Stop = Zastavení
```

---

## Verwendungsbeispiele (Kommandozeile)

### 1. Text-Extraktion aus DXF

Extrahiert Texte aus einer DXF-Datei und speichert sie in verschiedenen Formaten.

```
Extraktion in JSON (Standard)
bin\translate.bat --filename file.dxf --extract --outname output.json

Extraktion in Excel
bin\translate.bat --filename file.dxf --extract --export-type excel --outname
output.xlsx

Extraktion in Textdatei
```



```
bin\translate.bat --filename file.dxf --extract --export-type text --outname
output.txt

Mehrere Formate gleichzeitig (komma-separiert)
bin\translate.bat --filename file.dxf --extract --export-type excel,json,text --
outname output
Erzeugt: output.xlsx, output.json, output.txt

Ohne Angabe von --outname: Automatische Namensvergabe
bin\translate.bat --filename easy.dxf --extract --export-type excel,json
Erzeugt: easy_texts.xlsx, easy_texts.json

Mit Quellsprache (Standard: DE -> TEXT-D)
bin\translate.bat --filename file.dxf --extract --lang_source EN
```

### Extrahierte Entitäten:

- TEXT-Objekte im Modellbereich
- MTEXT-Objekte im Modellbereich
- Block-Attribute (nur Blöcke mit IO, ID, VERW, BEZEICHNUNG, KENNZEICHNUNG)

## 2. Text-Extraktion mit Übersetzung

Extrahiert und übersetzt Texte automatisch mittels Übersetzungs-Config.

```
Extraktion mit automatischer Übersetzung nach CS (Tschechisch)
bin\translate.bat --filename file.dxf --extract --translate CS --outname
output.json

Extraktion mit Übersetzung nach EN (Englisch)
bin\translate.bat --filename file.dxf --extract --translate EN --export-type excel
```

### Output-Format für eine Sprache (JSON):

```
{
 "translations": [
 {"source": "Weight", "target": "Hmotnost"}
],
 "untranslated": [
 {"source": "Custom text", "target": ""}
],
 "ignored": [
 "X=123.45",
 "LAYER=0"
]
}
```

## 3. Mehrsprachige Text-Extraktion

Extrahiert und übersetzt Texte gleichzeitig in mehrere Sprachen mit hierarchischer JSON-Struktur.

```
Extraktion mit mehreren Sprachen (komma-separiert)
bin\translate.bat --filename file.dxf --extract --translate CS,EN --outname
output.json

Drei Sprachen gleichzeitig
bin\translate.bat --filename file.dxf --extract --translate CS,EN,FR --export-type
json

Mit Excel-Export (nur erste Sprache wird in Excel verwendet)
bin\translate.bat --filename file.dxf --extract --translate CS,EN --export-type
excel,json
```

### Output-Format für mehrere Sprachen (hierarchische JSON-Struktur):

```
{
 "[CS]": {
 "translations": [
 {"source": "Emergency stop", "target": "Nouzové zastavení"},
 {"source": "Weight", "target": "Hmotnost"}
],
 "untranslated": [
 {"source": "Custom text", "target": ""}
]
 },
 "[EN]": {
 "translations": [
 {"source": "Emergency stop", "target": "Emergency stop"},
 {"source": "Weight", "target": "Weight"}
],
 "untranslated": [
 {"source": "Custom text", "target": ""}
]
 },
 "ignored": [
 "X=123.45",
 "LAYER=0"
]
}
```

## 4. DXF-zu-DXF Übersetzung

Erstellt eine neue DXF-Datei mit übersetzten Texten in mehrsprachigen Attributen.

```
Übersetzung TEXT-D -> TEXT-CS (verwendet automatisch file_texts.json)
bin\translate.bat --filename file.dxf --todxf file_cs.dxf --translate CS
```

```
Mit expliziter JSON-Datei
bin\translate.bat --filename file.dxf --todxf output.dxf --translate CS --
translation-json custom.json

Mit alternativer Quellsprache
bin\translate.bat --filename file.dxf --todxf output.dxf --lang_source EN --
translate CS
```

### Verhalten:

- **TEXT/MTEXT:** Direkt im Entity übersetzt
- **Block-Attribute:**
  - Liest aus Quell-Attribut (z.B. TEXT-D)
  - Schreibt in Ziel-Attribut (z.B. TEXT-CS)
  - Quell-Attribut bleibt unverändert
  - Erstellt fehlende ATTDEF im Block
  - Erstellt fehlende Layer (aus `[symbol_layer]`)

## 5. JSON-Reprocessing

Übersetzt den `untranslated` Block einer JSON-Datei erneut.

```
bin\translate.bat --fromjson output.json --translate CS
```

## 6. JSON Re-Translation

Re-übersetzt den `translations` Block mit aktualisierter Sprach-Config.

```
bin\translate.bat --retranslate-json output.json --translate CS
```

---

## Umgebungsvariablen

Das Programm benötigt folgende Umgebungsvariablen (werden in `bin/setenv.bat` gesetzt):

- `PROJECT_WORK`: Arbeitsverzeichnis für Ein-/Ausgabe-Dateien
- `PROJECT_CFG`: Konfigurationsverzeichnis (`cfg/`)
- `PROJECT_TRANSLATION`: Übersetzungs-Konfigurationsverzeichnis (`translation/`)

---

## Fehlerbehandlung und Ausgabe

Das Programm gibt detaillierte Statusmeldungen aus:

```
Anzahl gefundener TEXT Objekte: 150
Anzahl gefundener MTEXT Objekte: 25
```

```
Anzahl gefundener TEXT-D in Blöcken: 340
Duplikate entfernt: 75
Eindeutige Texte extrahiert: 440

Filterung abgeschlossen:
 Akzeptierte Texte: 380
 Ignorierte Texte: 60

DXF-Übersetzung abgeschlossen:
 TEXT übersetzt: 140, übersprungen: 10, ignoriert: 0
 MTEXT übersetzt: 20, übersprungen: 5, ignoriert: 0
 TEXT-CS (Blöcke) erstellt/aktualisiert: 310, übersprungen: 30, ignoriert: 0
 Gesamt: 470 übersetzt, 45 übersprungen, 0 ignoriert
```

---

## Tipps und Hinweise

- Texte werden als Set verarbeitet → Duplikate werden automatisch entfernt
- Leere Texte werden ignoriert
- Übersetzungen erfolgen in der Reihenfolge: Multi → Trigramme → Bigramme → Einzelwörter
- Bei DXF-zu-DXF Übersetzung bleibt das Original-Attribut immer unverändert
- Layer und ATTDEF werden nur bei Bedarf erstellt (lazy creation)
- Alle Ausgabedateien landen im **PROJECT\_WORK** Verzeichnis
- Die **.txt** Ausgabe eignet sich für schnelle Übersicht und Textverarbeitung
- Die **.json** Ausgabe wird für DXF-zu-DXF Übersetzung verwendet
- Die **.xlsx** Ausgabe eignet sich für externe Übersetzungsdienstleister

---

## Anhang A: Python-Modul translate.py

### Kommandozeilen-Parameter

#### Eingabe-Modi (mutual exclusive)

- **--extract** / **-e**: Extrahiere Texte aus DXF/DWG (benötigt **--filename**)
- **--fromtxt** **FILE.txt**: Filtere Texte aus Textdatei
- **--fromjson** **FILE.json**: Re-prozessiere JSON (benötigt **--translate**)
- **--retranslate-json** **FILE.json**: Re-übersetzt translations Block mit aktueller Config (benötigt **--translate**)
- **--todxf** **OUTPUT.dxf**: DXF-zu-DXF Übersetzung (benötigt **--filename**)

#### Hauptparameter

- **--filename** **FILE** / **-f** **FILE**: Eingabe-DXF/DWG-Datei
- **--outname** **NAME** / **-o** **NAME**: Name der Ausgabedatei (ohne Endung bei mehreren Formaten)
- **--translate** **LANG**: Übersetzungssprache (CS, EN, FR, etc.) oder mehrere kommasepariert (CS,EN,FR)
- **--lang\_source** **LANG**: Quellsprache für Block-Attribute (Standard: DE)
- **--export-type** **TYPE**: Exportformat(e): **json** (Standard), **excel**, **text** oder komma-separiert (z.B. **excel,json,text**)

- `--translation-json FILE`: JSON-Datei mit Übersetzungen für `--todxf`
- `--debug / -d`: Debug-Ausgabe für Pattern-Matching

## Wichtige Funktionen

### **extract\_text\_from\_dxf(filename, text\_attr\_name='TEXT-D')**

Extrahiert alle TEXT, MTEXT und Block-Attribute aus einer DXF-Datei.

#### **Parameter:**

- `filename`: Pfad zur DXF-Datei
- `text_attr_name`: Name des Text-Attributs in Blöcken (Standard: "TEXT-D")

#### **Rückgabe:**

- Alphabetisch sortierte Liste aller gefundenen Texte (ohne Duplikate)

### **filter\_texts(texts, wildcard\_patterns, regex\_patterns, debug=False)**

Filtert Texte basierend auf Wildcard- und Regex-Mustern.

#### **Parameter:**

- `texts`: Liste der zu filternden Texte
- `wildcard_patterns`: Liste der Wildcard-Muster
- `regex_patterns`: Liste der Regex-Muster
- `debug`: Debug-Ausgabe aktivieren

#### **Rückgabe:**

- Tuple mit (akzeptierte Texte, ignorierte Texte)

### **translate\_text(text, multi, trigramme, bigramme, single)**

Übersetzt einen Text durch Ersetzen von Multi-Phrasen, Trigrammen, Bigrammen und Einzelwörtern.

#### **Parameter:**

- `text`: Zu übersetzender Text
- `multi`: Dictionary mit Multi-Wort-Phrasen
- `trigramme`: Dictionary mit Drei-Wort-Phrasen
- `bigramme`: Dictionary mit Zwei-Wort-Phrasen
- `single`: Dictionary mit Einzelwörtern

#### **Rückgabe:**

- Übersetzter Text (leer wenn keine Übersetzung gefunden)

### **translate\_dxf\_texts(input\_dxf, output\_dxf, translations, ...)**

Übersetzt alle TEXT, MTEXT Objekte und Text-Attribute in Blöcken einer DXF-Datei.

**Parameter:**

- **input\_dxf**: Pfad zur Quell-DXF
- **output\_dxf**: Pfad zur Ziel-DXF
- **translations**: Dictionary mit source -> target Mappings
- **wildcard\_patterns**: Wildcard-Muster zum Filtern
- **regex\_patterns**: Regex-Muster zum Filtern
- **text\_source\_attr**: Quell-Attributname (Standard: "TEXT-D")
- **text\_target\_attr**: Ziel-Attributname (Standard: "TEXT-CS")
- **target\_layer**: Ziel-Layername für neue Attribute

**retranslate\_json\_file(json\_path, translation\_dicts\_all, translation\_languages)**

Re-übersetzt eine JSON-Datei mit neuen Übersetzungs-Configs.

**Parameter:**

- **json\_path**: Pfad zur JSON-Datei
- **translation\_dicts\_all**: Dict mit {LANG: (multi, trigramme, bigramme, single)}
- **translation\_languages**: Liste der Sprachen

## Interne Arbeitsweise

1. **Extraktion**: Texte werden aus DXF gelesen und in einem Set gespeichert (Duplikate werden entfernt)
2. **Filterung**: Ignore-Patterns werden angewendet (erst Wildcard, dann Regex)
3. **Übersetzung**: Texte werden in der Reihenfolge Multi → Trigramme → Bigramme → Single übersetzt
4. **Export**: Ergebnisse werden in gewünschte Formate geschrieben
5. **DXF-Übersetzung**: Neue DXF wird mit übersetzten Texten erstellt (Quell-Attribute bleiben erhalten)

## Block-Attribute und ATTDEF-Erstellung

Wenn ein Block noch kein ATTDEF für das Ziel-Attribut hat:

1. **ATTDEF wird erstellt** in der Block-Definition
2. **Layer wird erstellt** (falls nicht vorhanden, aus [**symbol\_layer**])
3. **Eigenschaften werden übernommen** vom Quell-ATTDEF:
  - Position (**insert**)
  - Höhe (**height**)
  - Stil (**style**)
  - Rotation (**rotation**)
  - Layer (aus Config oder vom Quell-Attribut)

## Erforderliche Block-Attribute

Blöcke müssen folgende Attribute haben, um verarbeitet zu werden:

- **IO**
- **ID**
- **VERW**
- **BEZEICHNUNG**

- **KENNZEICHNUNG**
  - Quell-Textattribut (z.B. **TEXT-D**)
- 

## Siehe auch

- [CLAUDE.md](#) - Allgemeine Projektdokumentation
- [cfg/translator.cfg](#) - Hauptkonfiguration
- [translation/\\*.cfg](#) - Sprachspezifische Übersetzungs-Configs
- [bin/tr2dxf.bat](#) - Zentrale Batch-Datei für Übersetzungen
- [bin/tr2dxf\\_\\*.bat](#) - Sprachspezifische Wrapper-Skripte