



abs-Viewer V 1-9 – Dokumentation

BVBS-konformer Viewer und Parser für Bewehrungsdaten (3.1 – Dt 202100614)

© 2025 CHP Constructions)

Inhalt

- 1 Überblick
- 2 Aufbau und Funktionsweise
- 3 Block-Struktur und Farblogik
- 4 Funktionen und Verhalten
- 5 Technische Details (Implementierung)
- 6 Parser-Logik
- 7 Rendering-Pipeline
- 8 Muffen-Logik
- 9 Sprachverwaltung
- 10 Erweiterbarkeit und zukünftige Versionen

1 Überblick

Der **abs-Viewer V 1-9** ist eine vollständig lokal lauffähige Web-Applikation zur Analyse, Visualisierung und Interpretation von **BVBS-konformen Bewehrungsdaten** (Format 3.1). Er basiert auf der BVBS-Richtlinie „Austausch von Bewehrungsdaten 3.1 (Dt 202100614)“ und wurde so erweitert, dass die Blöcke **Header (H)**, **Geometrie (G)**, **Muffen (M)** und **Prüfsumme (C)** präzise erkannt, farblich hervorgehoben und interaktiv analysiert werden.

2 Aufbau und Funktionsweise

Die App besteht aus einer einzigen HTML-Datei mit eingebettetem CSS/JavaScript, läuft offline (Chrome/Edge/Firefox) und benötigt keine Server- oder Datenbankverbindung. Das Layout ist dreigeteilt: links Import/Sprachenwahl; mittig Rendering + ABS-Dock; rechts Eigenschaften + Instruktionen.

3 Block-Struktur und Farblogik

Blocktyp	Farbe	Beispiele
Header (H)	 #7dd3fc	r = Plannummer, p = Positionsnummer, d = Durchmesser
Geometrie (G)	 #a7f3d0	l = Schenkellänge, r = Radius, w = Winkel
Muffen (M)	 #fca5a5	a,b,c,n,o,p = Verbindungs-/Kennzeichnungsdaten
Prüfsumme (C)	 #fcd34d	value = Checksumme

4 Funktionen und Verhalten

- **Dateiimport:** Einzelsatz-Analyse je ABS-Zeile, Navigation auf den aktiven Datensatz.
- **Sprachumschaltung:** 8 Sprachen, letzte Auswahl wird in `localStorage` gespeichert.
- **Instruktionen:** Rechts nur Feldkennungen des aktiven Datensatzes; gruppiert nach Block; ohne Dopplungen, ohne „@“; ausgeschriebene Blocknamen.
- **ABS-Dock:** Farbige Hervorhebung der Tokens exakt nach Blocktypen.
- **Muffen/ Gewinde-Abschnitt:** Wird nur angezeigt, wenn Block M vorhanden ist; zeigt a,b,c,n,o,p inkl. Interpretation (0 = kein Teil, 1 = Muffe, 2 = Gewinde).
- **Rendering-Labels:** Wenn Block M vorhanden ist, erscheinen rote Text-Labels der Verbindungstypen (`b` / `o`) an Stab-Anfang und -Ende.

5 Technische Details (Implementierung)

Struktur: Single-File-App (HTML mit eingebettetem CSS/JS). **Parser:** Token-basiert, erkennt „@B@“-Blöcke und Feldkennungen; mappt Felder kontextsensitiv. **Visualisierung:** SVG, Pfadaufbau aus Schenkeln/Bögen, farbliche Token-Hervorhebung; Text-Labels an Enden. **Sprach-Storage:** `localStorage("absLang")`. **Quelle:** BVBS-Richtlinie 3.1 (Dt 202100614).

6 Parser-Logik

Pro Zeile wird ein aktueller Blockkontext geführt. Großbuchstabe startet/wechseln den Block, Kleinbuchstabe kennzeichnet Felder. Identische Feldkennungen können je Block unterschiedliche Bedeutungen haben (z.B. `r` = Plannummer in H, `r` = Biegeradius in G). Der Parser legt je Zeile eine Struktur `map[Block] = [{id, val}]` an und erzeugt daraus Eigenschaften, Instruktionen und die farbliche Dock-Ansicht.

7 Rendering-Pipeline

2D/3D-Darstellung mittels SVG: Punkte werden skaliert/gedreht, Pfade gezeichnet, Labelpositionen mit kleinem Offset nahe der Stabenden. Bei fehlenden Koordinaten werden Bounding-Box-Enden als Anker verwendet.

8 Muffen-Logik

Wenn Block M enthalten ist, wird neben der textlichen Darstellung im Eigenschaften-Panel eine visuelle Kennzeichnung (rot) am Anfang/Ende gerendert. Kennzeichen `c` / `p` werden zusätzlich interpretiert (0/1/2).

9 Sprachverwaltung

Die UI-Sprache wird über ein Dropdown gewählt, in `localStorage` persistiert und beim Start wiederhergestellt. Feldbeschreibungen basieren auf einem mehrsprachigen Dictionary.

10 Erweiterbarkeit und zukünftige Versionen

Die Architektur erlaubt einfache Erweiterungen: zusätzliche BVBS-Blöcke, mehr Felder/Sprachen, BIM/IFC-Schnittstellen, erweiterte 3D-Interaktion (Zoom/Orbit), Maßstabsanzeige und Exportformate.

Hinweis: Diese Dokumentation beschreibt die Version **abs-Viewer V 1-9**. Für Anpassungen (z. B. neue Blocktypen) genügt eine Erweiterung der Feld-Dictionarys und des Tokenizers; Rendering-Labels lassen sich unabhängig vom Datensatztyp zufügen.