Draft HCV Spezifikation

Table of Contents

[Draft HCV Spezifikation 1](#_Toc386446449)

[1 Aufbau des HCV-Datencontainer 2](#_Toc386446450)

[2 KBL 3](#_Toc386446451)

[3 Index.xml 3](#_Toc386446452)

[4 SVG 7](#_Toc386446453)

[4.1 Unterstützte SVG Elemente 8](#_Toc386446454)

[4.1.1 Gruppe 8](#_Toc386446455)

[4.1.2 Linie 8](#_Toc386446456)

[4.1.3 Polylinie 9](#_Toc386446457)

[4.1.4 Polygon 9](#_Toc386446458)

[4.1.5 Path 9](#_Toc386446459)

[4.1.6 Circle 10](#_Toc386446460)

[4.1.7 Ellipse 10](#_Toc386446461)

[4.1.8 Rect 10](#_Toc386446462)

[4.1.9 Text 11](#_Toc386446463)

[4.1.10 Tspan 11](#_Toc386446464)

[4.2 Transformationen 11](#_Toc386446465)

[4.2.1 Matrix 11](#_Toc386446466)

[4.2.2 Rotate 12](#_Toc386446467)

[4.2.3 Scale 12](#_Toc386446468)

[4.2.4 Translate 12](#_Toc386446469)

[4.3 Wichtige Hinweise 12](#_Toc386446470)

[4.4 Verlinkung zwischen SVG und KBL 13](#_Toc386446471)

[4.5 Typ-Spezifizierer 14](#_Toc386446472)

[4.5.1 type:ref 14](#_Toc386446473)

[4.5.2 type:table 14](#_Toc386446474)

[4.5.3 type:row 14](#_Toc386446475)

[4.5.4 type:cell 15](#_Toc386446476)

[4.5.5 type:dimension 15](#_Toc386446477)

[4.5.6 type:DocumentFrame 16](#_Toc386446478)

[4.6 Objekte und ihre Darstellung 16](#_Toc386446479)

[4.6.1 Steckersymbol 17](#_Toc386446480)

[4.6.2 Steckertabelle 17](#_Toc386446481)

[4.6.3 Zubehörteile 19](#_Toc386446482)

[4.6.4 Bezugspfeile zur Tabelle: 21](#_Toc386446483)

[4.6.5 Hinweistexte 22](#_Toc386446484)

[4.6.6 Fixings 22](#_Toc386446485)

[4.6.7 Splices 24](#_Toc386446486)

[4.6.8 Vertex 26](#_Toc386446487)

[4.6.9 Segment / Bandierung / Rohr 26](#_Toc386446488)

[4.6.10 Kabelkanäle 27](#_Toc386446489)

[4.6.11 Uhrzeiten 28](#_Toc386446490)

[4.7 Bemaßungen 29](#_Toc386446491)

[4.8 Besonderheiten und implizite Logik bei der Darstellung 30](#_Toc386446492)

# Aufbau des HCV-Datencontainer

Der **H**arness **C**ontainer for **V**iewing dient zum Transport aller notwendigen Dateien, die einen gesamten Leitungssatz inklusive der verschiedenen Varianten bzw. Module beschreiben. Es können aber auch einzelne Leitungssatzmodule damit transportiert werden.

Er wird hauptsächlich zur Masterfreigabe verwendet, ist aber auch in vielen anderen Einsatzgebieten der Entwicklung oder Kalkulation, sowie des Musterbaus gut einsetzbar. Da sowohl die grafische Repräsentation der Leitungssatzzeichnung als auch alle technischen Informationen darin enthalten sind, können die meisten Fragestellungen bezüglich des Produktes selber als auch der Fertigung beantwortet werden.

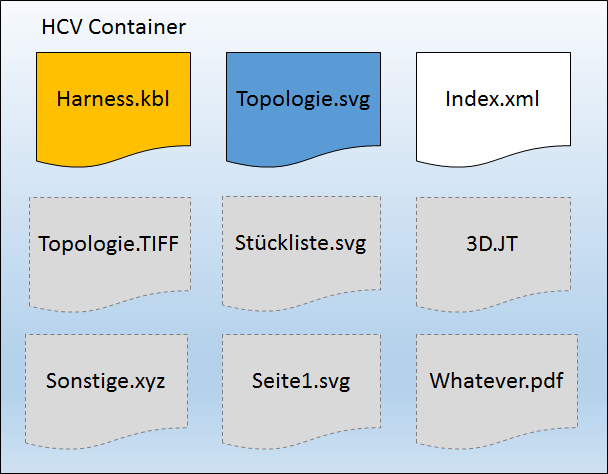
Das Softwareprodukt EE-Browser (Viewer) kann diese HCV Container lesen und deren Inhalt entsprechend visualisieren, vergleichen oder auch analysieren.

Ein HCV ist ein komprimierter (ZIP) Datencontainer, bestehend aus:

* genau einer KBL, die den gesamten Leitungssatz mit den Varianten und Modulen beinhaltet
* einer oder mehrerer SVG Dateien, die die grafische Repräsentation der Leitungssatzzeichnung auf gegebenenfalls mehreren Seiten darstellt. Die Namen der Dateien sollten den Inhalt kennzeichnen, damit man sie im Dokumentenbaum leichter wiederfinden kann, und müssen die Endung .svg haben. (Bsp.: Topologie.svg, Deckblatt.svg, Stückliste.svg…)
* genau einer Index-XML Datei, die eine Modulstückliste repräsentiert
* anderer optionaler Dateien, wie z.B. 3D Konturmodelle des Leitungssatzes
* optionaler „Redlining“ Informationen, die der EE-Browser dort einstellen kann

Die Komprimierung muss nach ZIP Standard erfolgen, es werden keine RAR/TAR oder GZIP Archive unterstützt.

Abhängig von Prozessanforderungen können in einem HCV Container auch weitere Dateien (z.B. HP-GL/2, TIFF, PDF) enthalten sein. Ein HCV Viewer muss die im Mindestumfang enthaltenen Dateien (KBL, SVG(s), Index XML) für das Viewing berücksichtigen. Die Zusätzlich enthaltenen Dateien können vom Viewing System berücksichtigt und entsprechend angeboten/ausgewertet werden.



# KBL

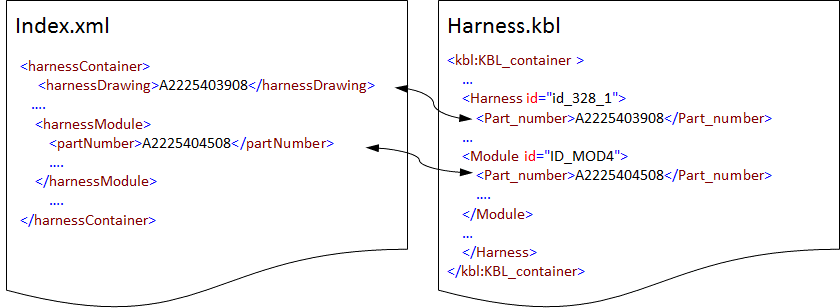
Ein HCV Container muss genau eine KBL Datei beinhalten. Die KBL Datei muss nach dem KBL Schema 2.3 ( siehe Kapitel [KBL](#_KBL) ) aufgebaut sein.

# Index.xml

Die Index XML ist eine Stückliste des Leitungssatzes mit entsprechender Zuordnung der Teile zu den jeweiligen Modulen. Sie wird in verschiedenen Prozessabläufen der Freigabe oder Auswertung benötigt.

Damit man diese einfache Basisinformation nicht jedes Mal aus der relativ komplexen Datenstruktur der KBL auslesen muss, wird die Datei aus dem Datenmodell des jeweiligen Autorensystems bei der Erstellung des HCV Containers erzeugt, oder aber von anderen Systemen generiert und in den Container eingestellt.

Der Datenstand der Index.xml **muss** mit dem Datenstand der KBL in Version und Inhalt übereinstimmen:

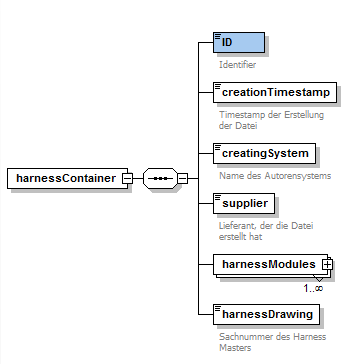


Der EE- Browser kann diese Datei, sofern vorhanden, lesen und deren Inhalt visualisieren. Die Funktion ist unter dem Menüpunkt BOM (Active Modules / All Modules) zu finden.

Alle Attribute bzw. Attributknoten müssen geschrieben werden (die min und max Occurence sind in der entsprechenden Schemadatei auf 1 gesetzt)

Das XML Schema der Index XML gliedert sich in drei Teilbereiche:

Die Metadaten des Harness-Containers:



ID <string>:

Identifier des Containers oder der Datei. Dieser wird zu Zeit nicht weiter ausgewertet und ist nur vorgehalten.

„id\_08\_15“

creationTimestamp<string>:

Zeitstempel der Erstellung der Index XML. Hier wird aus Gründen der Kompatibilität zurzeit ein String Datentyp verwendet, wobei das Format YYYY-MM-DD HH:MM:SS vorzusehen ist:

„2014-06-01 20:07:34“

creatingSystem<string>:

Name des Autorensystems von dem die Index XML generiert worden ist.

„HarnessDesigner 1.5“

supplier<string>:

Lieferantenname oder Kennzeichen desjenigen, der die Index XML generiert hat.

„Automotive Systems“

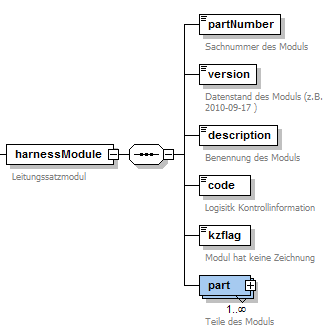
harnessDrawing<string>:

Sachnummer des Harness Masters. Die Sachnummer ist ohne Leerzeichen oder sonstige Sonderzeichen zu schreiben.

„A2225403908“

Die Harness-Module:

In diesem Abschnitt werden alle Module des Harness Masters aufgeführt.



partNumber<string>:

Sachnummer des Moduls. Die Sachnummer ist ohne Leerzeichen oder sonstige Sonderzeichen zu schreiben.

„A2225404108“

version<string>:

Datenstand des Moduls. Hier kann eine Versionsinformation eingetragen werden. Im Daimler Umfeld ist die Datumsinformation des Tagesstandes im Format YYYY-MM-DD zu verwenden.

„2011-11-07“

description<string>:

Beschreibung bzw. Benennung des Moduls.

„ZB EL.LTG.SATZ LL COCKPIT RDU“

code<string>:

Logistik-Kontrollinformation in Form eines Optionscodes.

„+LL+TY0+889“

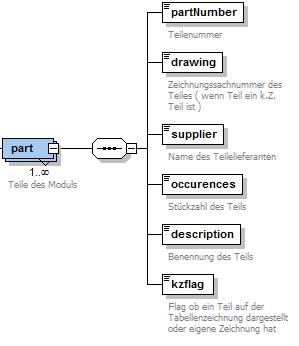
kzflag<boolean> („keine Zeichnung Flag“):

Wenn das Modul keine eigene Zeichnung hat, wird das Flag auf *true* gesetzt. Bei Masterumfängen ist das Flag automatisch auf *true* zu setzen, da alle Module als („kz-siehe“) abgebildet sind.

Mit folgender Vorschrift lässt sich das kzflag ermitteln:

Wenn der Eintrag der harnessDrawing in der Sektion harnessContainer leer ist, wird das kzflag auf *false* gesetzt. Anderenfalls ist dieser Eintrag mit dem Eintrag der partNumber unter der Sektion harnessModule zu vergleichen (Case Insensitive). Bei Ungleichheit ist das Flag auf *true* zu setzten, sonst verbleibt es *false*.

Die zu einem Modul gehörigen Teile:



partNumber<string>:

Sachnummer des Teils. Die Sachnummer ist ohne Leerzeichen oder sonstige Sonderzeichen zu schreiben.

„A2129820126“

drawing<string>:

Zeichnungsnummer, auf der das Teil dargestellt ist.

„A2129820126“

supplier<string>:

Lieferantenname des Teils oder entsprechendes Kurzzeichen.

„Coroplast“

occurences<integer>:

Anzahl der Teile. Im Falle von Bandierung wird hier die Anzahl der bandierten Segmente angegeben. „5“

description<string>:

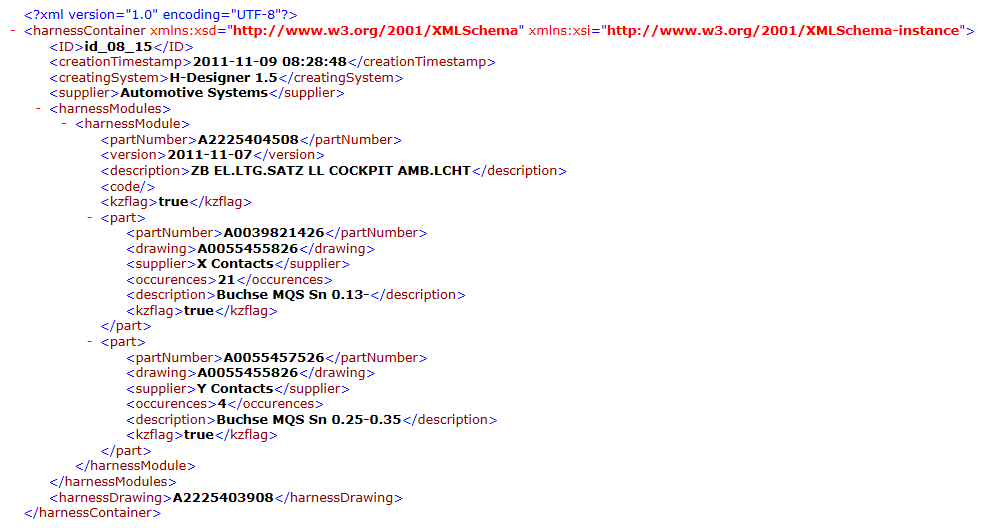
Beschreibung des Teils.

„Buchsenkontakt (gerade) Au 0.14-0.14“

kzflag<boolean>(„keine Zeichnung Flag“):

Einzelteile habe in der Regel keine eigene Zeichnung, daher ist das Flag hier immer auf *true* zu setzen (-> siehe Tabellenzeichnung).

Ausschnitt aus einer Index XML:



# SVG

Dieser Stand der Spezifikation beschreibt die Strukturen, die der EE-Browser in der Version 5.99 unterstützt. Als Referenz für Fragen der Darstellung kann der Microsoft Internet Explorer in der Version 9 verwendet werden.

Es sind in zukünftigen Versionen einige Änderungen nötig und sinnvoll, um besonders im Bereich der SVG Modellierung mehr Flexibilität und Dateneffizienz zu ermöglichen. In weiteren Entwicklungsschritten sollen die zusammengehörigen Leitungssätze eines gesamten Fahrzeuges mit in die Visualisierung einbezogen werden, um auch die Übergänge und Trennstellen zwischen den Leitungssätzen darstellen zu können. Damit steigen die Anforderungen an die zu verarbeitenden Datenmengen und Dateigrößen nochmals deutlich an. Daher ist bei der Generierung des HCV Containers und besonders der SVG Daten auf **schlanke** Strukturen und **effiziente** Darstellung zu achten.

Die SVG orientiert sich an dem W3C Standard (<http://www.w3.org/TR/SVG11>), aber es werden in der aktuellen Version nur bestimmte Elemente unterstützt, und innerhalb des jeweiligen Elementes sind nur bestimmte Spezifizierer bzw. Attribute zugelassen. Diese werden in den nachfolgenden Abschnitten behandelt.

Auch in zukünftigen Versionen werden bestimmte Elemente **nicht** unterstützt werden. Hierzu gehört in jedem Falle das „ClipPath“ Element. „ClipPathes“ werden bei der Verarbeitung der SVG und der Darstellung ignoriert.

SVG Befehle und Attribute, die nicht unterstützt werden, dürfen trotzdem in der SVG vorkommen, sie werden vom Browser einfach ignoriert.

In einer Svg werden zu Beginn die Breite und Höhe sowie die Größe der Viewbox angegeben. Damit die Svg z.B. auch im Internet Explorer oder anderen Programmen für sich alleine darstellbar ist, müssen diese Werte entsprechend bei der Generierung beschrieben werden. Der EE Browser verwendet diese Information **nicht**, er ermittelt die Koordinaten der umgebenden Begrenzung selber und alle Anzeigen basieren auf dem World Koordinatensystem (WCS), die Verwendung von User-Koordinatensystemen und verschiedenen Einheiten wird nicht unterstützt.

Die Hintergrundfarbe der Dokumente ist fest auf **weiß** eingestellt und kann nicht vorgegeben werden.

## Unterstützte SVG Elemente

Die grafischen Elemente und Gruppen können innerhalb des svg Dokumentknotens in beliebiger Reihenfolge angeordnet werden, wobei eine strukturierte Auflistung die Lesbarkeit deutlich vereinfacht.

### Gruppe

<g transform(*translate,rotate,scale*)/>

Das Gruppenelement wird zur Gruppierung von zusammengehörigen Basiselementen benötigt. Gruppen können in beliebiger Tiefe geschachtelt werden. Dieses Element wird für alle Zusammenfassungen von Business Objekten innerhalb des Browsers gebraucht. Z.B. Steckertabellen, Steckeransichten usw.

Beispiel:

*<g>*

*<g transform="translate(8976.7,46.8) rotate(-0.97)">*

*<line x1="4" y1="0" x2="117" y2="0" style="stroke:black; stroke-width:0.5"/>*

*<polygon points="0 0 ,4 -1 ,4 1" style="fill:black" />*

*</g>*

*</g>*

### Linie

<line x1,y1,x2,y2,style(stroke;stroke-width)/>

x1,y1: Startkoordinaten im WCS

x2,y2: Endkoordinaten im WCS

style: Es werden die stroke Attribute für Farbe und Linienbreite unterstützt. Bei der Farbangabe ist sowohl die Verwendung des Farbnamens (siehe W3C Spezifikation) sowie ein RGB Wert zulässig. Die RBG Werte sind Dezimal anzugeben.

Beispiel:

*<line x1="25" y1="10" x2="25" y2="15" style="stroke:black; stroke-width:0.5 " />*

*Oder*

*<line x1="25" y1="10" x2="25" y2="15" style="stroke: rgb(100,113,207); stroke-width:0.5 " />*

### Polylinie

<polyline points(list of points) style(fill; opacity; stroke; stroke-width) />

points: eine Liste von XY Paaren im WCS

style: Es werden die stroke Attribute für Farbe und Linienbreite unterstützt. Bei der Farbangabe ist sowohl die Verwendung des Farbnamens (siehe W3C Spezifikation) sowie ein RGB Wert zulässig. Die RBG Werte sind Dezimal anzugeben. Das fill Attribut legt die Füllfarbe fest oder steht auf none. Ein opacity Wert von 0 setzt die Füllung auf „ungefüllt“.

Beispiel:

*<polyline points="8.453 -20.554,11.034 -8.973 ,8.41 -8.322 ,8.215 -9.189 " style="stroke:black; stroke-width:0.1; fill:none" />*

### Polygon

< polygon points(list of points) style(fill; opacity; stroke; stroke-width) />

points: eine Liste von XY Paaren

style: Es werden die stroke Attribute für Farbe und Linienbreite unterstützt. Bei der Farbangabe ist sowohl die Verwendung des Farbnamens (siehe W3C Spezifikation) sowie ein RGB Wert zulässig. Die RBG Werte sind Dezimal anzugeben. Das fill Attribut legt die Füllfarbe fest oder steht auf none. Ein opacity Wert von 0 setzt die Füllung auf „ungefüllt“.

Beispiel:

*<polygon points="-0.462 3.5 ,1.813 1.225 ,13.187 1.225"*

*style="stroke:rgb(78,113,207); stroke-width:0.5; fill:none" />*

### Path

<path d(A, a, C, c, Q, q, S, s, T, t, H, h, L, l, M, m V, v) style(fill, opacity, stroke, stroke-width) />

Angaben in Großbuchstaben kennzeichnen absolute Koordinaten, mit Kleinbuchstaben sind relative Koordinaten bezeichnet.

* + - A, a: Arc-Definition
    - C, c, Q, q, S, s, T, t: Curve-Definition
    - H, h: Horizontal-Line-Definition
    - L, l: Line-Definition
    - M, m: MoveTo-Definition
    - V, v: Vertical-Line-Definition

style: Es werden die stroke Attribute für Farbe und Linienbreite unterstützt. Bei der Farbangabe ist sowohl die Verwendung des Farbnamens (siehe W3C Spezifikation) sowie ein RGB Wert zulässig. Die RBG Werte sind Dezimal anzugeben. Das fill Attribut legt die Füllfarbe fest oder steht auf none. Ein opacity Wert von 0 setzt die Füllung auf „ungefüllt“.

Beispiel:

*<path d="M-8 -4 A17 17,0,0,1,8 -4" style="stroke:black; stroke-width:0.25; fill:none" />*

### Circle

<circle cx,cy,r,style(fill, opacity, stroke, stroke-width)/>

cx,cy: Ursprungskoordinaten in WCS

r: Radius

style: Es werden die stroke Attribute für Farbe und Linienbreite unterstützt. Bei der Farbangabe ist sowohl die Verwendung des Farbnamens (siehe W3C Spezifikation) sowie ein RGB Wert zulässig. Die RBG Werte sind Dezimal anzugeben. Das fill Attribut legt die Füllfarbe fest oder steht auf none. Ein opacity Wert von 0 setzt die Füllung auf „ungefüllt“.

Beispiel:

*<circle cx="6545" cy="393" r="0.5" style="fill:rgb(78,113,207)" />*

### Ellipse

<ellipse cx,cy,rx,ry,style(fill, opacity, stroke, stroke-width)/>

cx,cy: Ursprungskoordinaten in WCS

rx,ry: X- und Y-Achsen Radius

style: Es werden die stroke Attribute für Farbe und Linienbreite unterstützt. Bei der Farbangabe ist sowohl die Verwendung des Farbnamens (siehe W3C Spezifikation) sowie ein RGB Wert zulässig. Die RBG Werte sind Dezimal anzugeben. Das fill Attribut legt die Füllfarbe fest oder steht auf none. Ein opacity Wert von 0 setzt die Füllung auf „ungefüllt“.

Beispiel:

*< ellipse cx="6545" cy="393" rx="0.5" ry="1" style=" fill:none stroke:blue stroke-width:20/>*

### Rect

<rect x,y,width,height,style(fill, opacity, stroke, stroke-width) />

x,y: Ursprungskoordinaten in WCS

width: Breite

height: Höhe

style: Es werden die stroke Attribute für Farbe und Linienbreite unterstützt. Bei der Farbangabe ist sowohl die Verwendung des Farbnamens (siehe W3C Spezifikation) sowie ein RGB Wert zulässig. Die RBG Werte sind Dezimal anzugeben. Das fill Attribut legt die Füllfarbe fest oder steht auf none. Ein opacity Wert von 0 setzt die Füllung auf „ungefüllt“.

Die Werte Rx und Ry für gerundete Ecken werden in der aktuellen Version nicht unterstützt.

Beispiel:

*<rect x="0" y="0" width="15" height="7" style="fill:rgb(255,255,255); opacity:0" />*

### Text

<text x,y, font-weight, style (fill, font-family, font-size, text-anchor, text-decoration)>Value</>

x,y: Ursprungskoordinaten in WCS, eine Liste von Punkten wird hier nicht unterstützt

font-weight: Bei diesem Attribut wird nur der Wert “bold” verwendet.

style: Mit dem Attribut text-anchor kann die horizontale Ausrichtung angegeben werden. Die font-family erlaubt die Angabe des Fonts. Font-Verweise sind nicht möglich. Dieser muss auf dem Zielsystem verfügbar sein, ansonsten wird auf einen Default zurückgegriffen. Über das Attribut fill kann die Farbe des Textes gesetzt werden. Bei der Farbangabe ist sowohl die Verwendung des Farbnamens (siehe W3C Spezifikation) sowie ein RGB Wert zulässig. Die RBG Werte sind Dezimal anzugeben. Mit der text-decoration können Unter-, Über-, und Durchstreichungen erstellt werden. Mit der font-size kann die Größe eingestellt werden. Diese ist in Pixel anzugeben.

Beispiel:

*<text x="375" y="13.8" font-weight="bold" style="text-anchor:middle; font-family:Arial; font-size:3px; fill:black; text-decoration:underline">Dies ist ein Text</text>*

### Tspan

<tspan x,y,font-weight,style(text-anchor, text-decoration)>Value</ tspan >

x,y: Ursprungskoordinaten in WCS, eine Liste von Punkten wird hier nicht unterstützt

font-weight: Bei diesem Attribut wird nur der Wert “bold” verwendet.

style: Mit dem Attribut text-anchor kann die horizontale Ausrichtung angegeben werden. Mit der text-decoration können Unter-, Über-, und Durchstreichungen erstellt werden.

Der tspan wird nur in Verbindung mit einem Text Element verwendet werden. Es werden die Angaben für die Größe, Farbe, Font vom Text Element übernommen.

Beispiel:

*<text x="0" y="10" style="text-anchor:middle; font-family:Arial; font-size:12px; fill:black">X30/20<tspan x="0" y="3.214" style="text-anchor:middle">nach elektrischer Prüfung</tspan><tspan x="0" y="17.013" style="text-anchor:middle">alle Abgriffe in CAN-Verteiler stecken</tspan></text>*

### Image

<image x,y,height,width,xlink:href />

x,y: Ursprungskoordinaten im WCS, die relativ zu den (möglichen) darübergelagerten Gruppenelement-Koordinaten interpretiert warden.

height,width: Höhe und Breite des Bildes in px.

xlink:href: Link zum im Filesystem abgelegten Bild, welches angezeigt werden soll. Dabei kann ein relativer Pfad angegeben werden.

Für die Verbesserung der internen Dateistruktur im HCV sollten Bilder, die für SVGs verwendet werden, in einem Unterordner „Pictures“ gehalten werden.

Beispiel:

*<image width="100px" height="100px" xlink:href="myImage.png" />*

## Transformationen

### Matrix

<matrix(a,b,c,d,e,f)>

Es werden die Matrix Transformationen unterstützt. Dabei können die Matrixelemente

A00:a

A01:c

A10:b

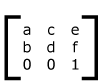
A11:d

A03:e

A13:f

verwendet werden (siehe auch W3C).

Anordnung der Matrix Elemente in der SVG Transformationsmatrix:



Beispiel:

*<g transform="matrix(-1,0,0,-1,-4.4,39.6)">*

### Rotate

<rotate(angle)>

Der Rotationswinkel ist in Grad anzugeben!

Beispiel:

*<g transform="rotate(-90)" />*

### Scale

<scale(x,y)>

Beispiel:

*<g transform="scale(1,2)">*

### Translate

<translate(x,y)>

Beispiel:

*<g transform="translate(0,1990) rotate(-90)" />*

Hinweis: die Skew Kommandos werden nicht unterstützt.

## Wichtige Hinweise

Grundsätzlich gibt es zwei Möglichkeiten Texte in einer SVG zu transportieren. Zum einen können SVG Text Elemente verwendet werden (siehe oben), zum anderen können Texte als gefüllte Polygone gewandelt werden. Viele SVG Konverter nutzen die zweitgenannte Möglichkeit, da man hier auf dem Zielrechner keine Information über den verwendeten Font benötigt. In typische CAD Zeichnungen lässt sich dieser Ansatz prinzipiell auch verwenden, da man hier nur wenige Texte hat.

Für den Einsatz im HCV Container ist **nur** der erstgenannte Weg über die Verwendung von Text Elementen sinnvoll, da die SVG Dateigröße sonst viel zu groß wird- speziell bei entsprechenden Innenraumleitungssätzen!

Weiterhin sollte darauf geachtet werden, nicht unnötig viele Koordinatensätze zu produzieren. Dies passiert häufig bei der Darstellung von z.B. Steckerbildern, die aus 3D Modellen mittels bestimmter Konverter in 2D überführt werden. Hier werden dann oft unzählige Polylinien mit vielen Vertexes generiert, die für die grafische Darstellung eigentlich völlig überflüssig sind, aber in ihrer Anzahl nachher auch signifikant zum Speicherverbrauch beitragen.

Die Verwendung von Pattern-Definitionen und Dash-Arrays wird in der aktuellen Version noch nicht unterstützt, da die rückwärtige Umsetzung von dieser SVG Informationen innerhalb des Browser nicht zu der gleichen Darstellung führt.

Textpathes (Texte, die beliebigen Verläufen folgen) werden nicht unterstützt.

## Verlinkung zwischen SVG und KBL

Um die Navigation zwischen Grafik und physikalischen Elementen zu ermöglichen, wird von der SVG aus innerhalb der Gruppenelemente auf Elemente der KBL über den internen KBL-Identifier (id) mittels definierter Kommentare verwiesen.

Die in der KBL vergebenen Identifier für die verschieden Objekte sind nach folgendem Muster aufgebaut.

|  |  |
| --- | --- |
| ID Präfix | Objekt Typ |
| ID\_BNJ | BundleJunction (Vertex) |
| ID\_BNS | BundleSegment (Segment) |
| ID\_CON | Connector (Stecker) |
| ID\_ACC | Accessory (Zubehörteil) |
| ID\_PLU | Plug (Blindstopfen) |
| ID\_SEA | Seal (Dichtung) |
| ID\_TER | Termination (Terminal) |
| ID\_TAP  ID\_WIR | Tape (Bandierung und Tube)  Wire (Leitung) |
| ID\_FIX | Fixing (Clip / Befestigung) |
| ID\_TXT | Texte, wie z.B. Fertigungshinweise |
| ID\_TCN | Zusatztabellen am Stecker |
| ID\_IMG | Bilder, können Fertigungshinweise sein |

Diese Vorgehensweise ist bei Daimler zu verwenden, da eine SVG Schemaerweiterung einen entsprechenden Aufwand auch in der Verwaltung und Bereitstellung des Schemas bedeutet hätte. Zudem ist diese Erweiterung einfacher in vorhandene Autorensysteme einzubringen.

Ist das erste Element innerhalb einer Gruppe ein solcher Kommentar <!--kbl-id:xyz ……-->, so wird er entsprechend ausgewertet; anderenfalls dient das Gruppenelement rein der Strukturierung.

Dieser Kommentar enthält ein oder mehrere Identifier von KBL Elementen, die zu dieser Gruppe gehören. Mehrere Identifier sind durch Leerzeichen zu separieren.

Die damit beschriebene Verlinkung kann entweder eine „ist“ (z.B. Steckersymbol) oder eine „gehört zu“ (z.B. Steckertabelle) Beziehung darstellen. Im zweiten Fall kann der Kommentar um einen zusätzlichen Typ-Spezifizierer (siehe folgender Abschnitt) erweitert werden, der es dem Browser ermöglicht, die verschiedenen Repräsentationen eines physikalischen Objektes gegebenenfalls auch unterscheiden zu können.

Ein Steckersymbol (face view) wird z.B. wie folgt gekennzeichnet:

*<g>*

*<!--kbl-id:ID\_CON167-->*

*…Grafikelemente des Steckersymbols…*

*</g>*

## Typ-Spezifizierer

Folgende Typ-Spezifizierer sind bisher vereinbart:

### type:ref

Der type:ref bezeichnet ein allgemein referenzierendes Objekt ohne weitere Präzisierung. Es wird verwendet, wenn z.B. spezielle Zusatzgrafik wie Bezugspfeile oder Textfahnen mit Objekten verbunden werden soll. Elemente vom Typ ref werden nicht in die Selektion genommen und auch bei bestimmten Ansichten, wie z.B. der Start-End Connector Ansicht nicht dargestellt.

*<g>*

*<!--kbl-id:ID\_CON1;type:ref-->*

*…Irgendwelche grafischen Elemente zum Stecker …*

*</g>*

### type:table

Der type:table bezeichnet die Repräsentation von Tabellen als Ganzes (z.B. Steckertabelle). Innerhalb dieser Struktur können Zeilen und Zellen angegeben werden.

*<g>*

*<!--kbl-id:ID\_CON1;type:table-->*

*…Inhalte und grafische Elemente der Tabelle …*

*</g>*

### type:row

Der Type:row bezeichnet die Repräsentation einer inhaltlichen Zeile (nicht Überschrift) innerhalb einer Tabelle.

*<g>*

*<!--kbl-id:ID\_CON1;type:table-->*

*…Inhalte und grafische Elemente der Tabelle …*

*<g>*

*<!--kbl-id:ID\_W1;type:row-->*

*…Inhalt der Tabellenzeile mit einem Bezug zur Leitung ID\_W1…*

*</g>*

*</g>*

### type:cell

Der type:cell bezeichnet die Repräsentation einer Zelle innerhalb einer Zeile in der Tabelle.

*<g>*

*<!--kbl-id:ID\_CON1;type:table-->*

*…Inhalte und grafische Elemente der Tabelle …*

*<g>*

*<!--kbl-id:ID\_W1;type:row-->*

*…Inhalt der Tabellenzeile mit einem Bezug zur Leitung ID\_W1…*

*<g>*

*<!--kbl-id:ID\_SEA1;type:cell-->*

*…Inhalt der Tabellenzelle mit einem Bezug zur Einzeldichtung SEA1…*

*</g>*

*</g>*

*</g>*

### type:dimension

Der type:dimension bezeichnet die Repräsentation eines Bemaßungsobjektes. Bemaßungsobjekte verweisen in der Regel auf die beiden Vertex-Identifier des entsprechenden Segments, für das die Bemaßung gilt (BNJ-BundleJunction). Im nachfolgenden Beispiel sind der Bemaßungstext und der Maßpfeil voneinander getrennt geschrieben worden. Das ist nicht zwingend nötig, findet sich aber in einigen Beispieldateien; im dritten Gruppenabschnitt ist das zusammengefasst worden.

Beispiel:

Bemaßungstext

*<g>*

*<!--kbl-id:ID\_BNJ1 ID\_BNJ109;type:dimension-->*

*<g transform="translate(6739.5,3) rotate(0) translate(0,-10)">*

*<text x="0" y="1.3" style="text-anchor:middle; font-family:CorpoS; font-size:5px; text-decoration:underline; fill:black">50</text>*

*</g>*

*</g>*

Maßpfeil

*<g>*

*<!--kbl-id:ID\_BNJ1 ID\_BNJ109;type:dimension-->*

*<g transform="translate(6739.5,37) rotate(0)">*

*<line x1="-68.5" y1="-1" x2="-68" y2="-11" style="stroke:black; stroke-width:0.25" />*

*<polygon points="-68.5 -10 ,-64 -11 ,-64.5 -9" style="fill:black" />*

*<line x1="-3.5" y1="-10" x2="-64" y2="-10" style="stroke:black; stroke-width:0.25" />*

*<line x1="68.5" y1="-1" x2="68" y2="-11" style="stroke:black; stroke-width:0.25 " />*

*<polygon points="68.5 -10 ,64 -11 ,64.5 -9" style="fill:black" />*

*<line x1="3.5" y1="-10" x2="64" y2="-10" style="stroke:black; stroke-width:0.25" />*

*</g>*

*</g>*

Zusammengefasst:

*<g>*

*<!--kbl-id:ID\_BNJ1 ID\_BNJ109;type:dimension-->*

*<g transform="translate(6739.5,3) rotate(0) translate(0,-10)">*

*<text x="0" y="1.3" style="text-anchor:middle; font-family:CorpoS; font-size:5px; text-decoration:underline; fill:black">50</text>*

*</g>*

*<g transform="translate(6739.5,377) rotate(0)">*

*<line x1="-68.5" y1="-1" x2="-68" y2="-11" style="stroke:black; stroke-width:0.25" />*

*<polygon points="-68.5 -10 ,-64 -11 ,-64.5 -9" style="fill:black" />*

*<line x1="-3.5" y1="-10" x2="-64" y2="-10" style="stroke:black; stroke-width:0.25" />*

*<line x1="68.5" y1="-1" x2="68" y2="-11" style="stroke:black; stroke-width:0.25 " />*

*<polygon points="68.5 -10 ,64.5 -11 ,64.5 -9" style="fill:black" />*

*<line x1="3.5" y1="-10" x2="64" y2="-10" style="stroke:black; stroke-width:0.25" />*

*</g>*

*</g>*

### type:DocumentFrame

Der type:DocumentFrame bezeichnet die Gruppe des Blattrahmens. In dieser Gruppe sind alle grafischen Objekte unterzubringen, die für den Blattrahmen benötigt werden (Planquadratangaben usw.)

*<g>*

*<!--kbl-id:DocumentFrame-->*

*<rect x="0" y="0" width="10289" height="841" style="fill:rgb(255,255,255)" />*

*<rect x="0" y="0" width="10289" height="841" style="stroke:black; stroke-width:0.01; fill:none" />*

*<rect x="20" y="10" width="10259" height="821" style="stroke:black; stroke-width:0.5; fill:none" />*

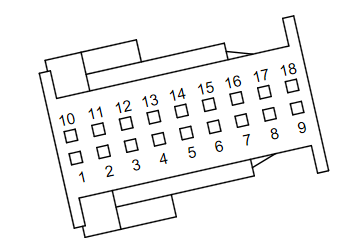
*….*

*</g>*

## Objekte und ihre Darstellung

### Steckersymbol

Das Steckersymbol verweist auf das entsprechende Connector Objekt in der KBL. Hier ist es in der Gruppe der zugehörigen Vertex angelegt, sodass bei Selektion der Vertex das Symbol automatisch mit in die Selektion genommen wird. Die Vertex selber hat keine Grafik hier - sie ist durch das Steckersymbol repräsentiert - nur der Verweis auf den zugehörigen node in der KBL ist eingetragen. (siehe auch Abschnitt Vertex weiter unten).



1

Svg Teilauschnitt:

<g>

Vertex des Symbols

<!--kbl-id:ID\_BNJ79-->

<g>

<!--kbl-id:ID\_CON54-->

<g transform="translate(3757.08,212.09) rotate(257.26) translate(-18.54,-2.1)">

<g transform="translate(0,0) scale(1,1) translate(18,25)">

<g transform="matrix(0,1,-1,0,-17.599,-25.599)">

<text x="4" y="-21.599" style="text-anchor:start; font-family:Arial; font-size:2.74px; fill:black">10</text>

1

…und weitere Kammertexte…

</g>

<line x1="-3" y1="21" x2="-3" y2="19" style="stroke:black; stroke-width:0.25 " />

…und weitere Grafik…

</g>

</g>

</g>

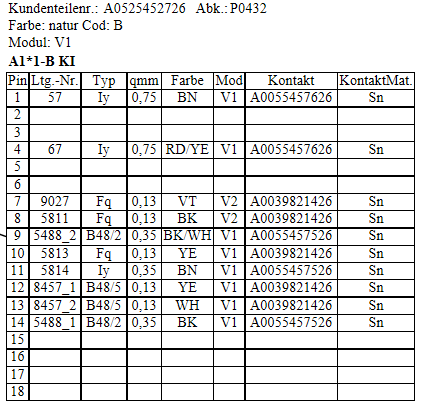
</g>

### Steckertabelle

Die Steckertabelle verweist auf das entsprechende Connector Objekt in der KBL. Sie wird über die Typ-Angabe spezifiziert. Zu Beginn werden die grafischen Elemente des Tabellenkopfes beschrieben. Nachfolgend kommt die Angabe der Spaltenüberschriften.

Im Anschluss daran werden die einzelnen Zeilen jeweils als eine Gruppe geschrieben und auf das zugehörige Wire Objekt der KBL verwiesen, wobei diese Verweise vom Typ row zu setzen sind. Innerhalb einer Zeile werden Untergruppen verwendet, um die Zellinhalte für z.B. Kontaktsachnummer zu spezifizieren. Diese Untergruppen verweisen auf das Termination Objekt in der KBL und müssen auf den Typ cell gesetzt werden. Im nachfolgenden Beispiel sind die Kontaktsachnummer und das Kontaktmaterial in einer solchen Untergruppe und können daher entsprechend ein und ausgeblendet werden.

Die Zellränder und Füllungen sind hier als gefüllte Rechtecke ausgeführt. Dabei ist zuerst der Zellhintergrund getrennt von der Umrandung angegeben. Idealerweise setzt man besser die Zellfüllung auf ungefüllt und beschreibt den Hintergrund der Zelle gar nicht weiter, denn der Rendering Prozess macht in bestimmten Ansichten aus explizit beschriebenen Farben einen grauen Hintergrund.



1

6

5

4

3

2

Svg Teilauschnitt:

<g>

<!--kbl-id:ID\_CON54;type:table-->

<g transform="translate(3810.0,149.09) rotate(0)">

<g>

Hintergrund und Zelle getrennt

<rect x="0" y="0" width="33" height="6" style="fill:rgb(255,255,255); opacity:0" />

<rect x="0" y="0" width="33" height="6" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="0.833" y="4.332" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">Kundenteilenr.:</text>

….

1

</g>

<g>

Zelle mit „transparentem“ Hintergrund

<rect x="0" y="25" width="8" height="6" style="stroke:black; stroke-width:0.25; fill:none" />

<text x="4" y="29.332" style="text-anchor:middle; font-family:CorpoS; font-size:5px; fill:black">Pin</text>

…

2

<g>

<!--kbl-id:ID\_WIR1239;type:row-->

<rect x="0" y="31" width="8" height="6" style="fill:rgb(255,255,255)" />

<rect x="0" y="31" width="8" height="6" style="stroke:black; stroke-width:0.25; fill:none" />

<text x="4" y="35.332" style="text-anchor:middle; font-family:CorpoS; font-size:5px; fill:black">1</text>

3

<rect x="8" y="31" width="18" height="6" style="fill:rgb(255,255,255)" />

<rect x="8" y="31" width="18" height="6" style="stroke:black; stroke-width:0.25; fill:none"/>

<text x="17" y="35.332" style="text-anchor:middle; font-family:CorpoS; font-size:5px; fill:black">57</text>

…..

4

<g>

<!--kbl-id:ID\_TER1837;type:cell-->

<rect x="83" y="31" width="32" height="6" style="fill:rgb(255,255,255)" />

<rect x="83" y="31" width="32" height="6" style="stroke:black; stroke-width:0.25; fill:none" />

<text x="99" y="35.332" style="text-anchor:middle; font-family:CorpoS; font-size:5px; fill:black">A0055457626</text>

</g>

5

<g>

<!--kbl-id:ID\_TER1837;type:cell-->

<rect x="115" y="31" width="27" height="6" style="fill:rgb(255,255,255)" />

<rect x="115" y="31" width="27" height="6" style="stroke:black; stroke-width:0.25; fill:none" />

<text x="128.5" y="35.332" style="text-anchor:middle; font-family:CorpoS; font-size:5px; fill:black">Sn</text>

</g>

6

</g>

</g>

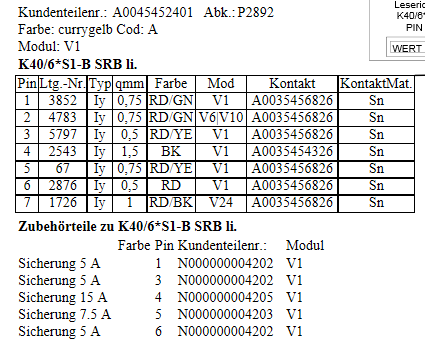
</g>

</g>

### Zubehörteile

Zubehörteile zum Stecker werden meist unterhalb der Steckertabellen dargestellt. Sie sind innerhalb der Gruppe der Steckertabelle angeordnet. Diese Teile verweisen in der Regel auf Accessory Objekte der KBL. Manchmal wird auch auf ein Component Objekt verwiesen (meist bei Sicherungen, wie hier im Beispiel).

Im Grunde handelt es sich bei dieser Darstellung wieder um eine Tabelle, und die einzelnen Zeilen werden wieder entsprechend über den Typ als row gekennzeichnet.



1

2

Svg Teilauschnitt:

<g>

<!--kbl-id:ID\_CON132;type:table-->

…..Hier ist der gesamte Block der eigentlichen Steckertabelle zu finden (siehe auch Abschnitt Steckertabelle)…

<g>

<rect x="0" y="75" width="35" height="7" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="0.833" y="79.756" font-weight="bold" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">Zubehörteile zu K40/6\*S1-B SRB li.</text>

1

<rect x="0" y="82" width="35" height="6" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<rect x="35" y="82" width="13" height="6" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="35.833" y="86.332" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">Farbe</text>

…..

<g>

<!--kbl-id:ID\_ACC54;type:row-->

<rect x="0" y="88" width="35" height="6" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="0.833" y="92.5" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">Sicherung 5 A</text>

2

……

</g>

</g>

</g>

Zugehörige Beschreibung in der KBL bei Verweisen auf eine Komponente:

<Component\_occurrence id="ID\_ACC54">

<Id>ID\_ACC54</Id>

<Mounting>ID\_CON132 id\_370\_774</Mounting>

<Part>id\_99902\_1</Part>

</Component\_occurrence>

<Component id="id\_99902\_1">

<Part\_number>N000000004202</Part\_number>

….

<Version>20110329-ST30611</Version>

<Abbreviation>P0026</Abbreviation>

<Description>Sicherung 5 A</Description>

<Processing\_information id="id\_336\_1">

<Instruction\_type>Type</Instruction\_type>

<Instruction\_value>Fuse</Instruction\_value>

</Processing\_information>

</Component>

Zugehörige Beschreibung in der KBL bei Verweisen auf ein Accessory Objekt:

<Accessory\_occurrence id="ID\_ACC65">

<Id>ID\_ACC65</Id>

<Placement id="id\_345\_197">

…

</Placement>

<Part>id\_300\_5</Part>

<Reference\_element>ID\_CON18</Reference\_element>

<Installation\_information id="id\_329\_1027">

<Instruction\_type>Colour</Instruction\_type>

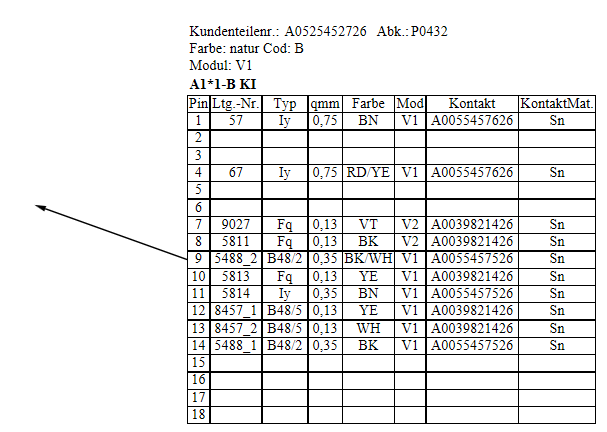
<Instruction\_value>violett</Instruction\_value>

</Installation\_information>

</Accessory\_occurrence>

### Bezugspfeile zur Tabelle:

Der Bezugspfeil wird hier quasi wie ein Stück der Tabelle gesehen (type:table) und verweist dementsprechend auch wieder auf das Connector Objekt in der KBL. Dieser Bezugspfeil könnte prinzipiell auch direkt mit in die Definition der Steckertabelle aufgenommen werden.



Svg Teilauschnitt:

<g>

<!--kbl-id:ID\_CON54;type:table-->

<g transform="translate(3757.0,212.0) rotate(19.7)">

<line x1="4" y1="0" x2="56.303" y2="0" style="stroke:black; stroke-width:0.5" />

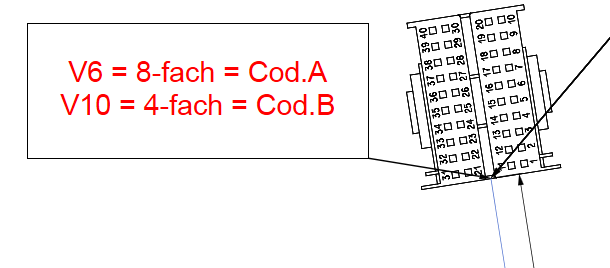
<polygon points="0 0 ,4 -1 ,4 1" style="fill:black" />

</g>

</g>

### Hinweistexte

Hinweise zu z.B. Steckern oder auch andere Objekten verweisen auf das zugehörige KBL Objekt und werden mit dem Typ Spezifizierer type:ref gekennzeichnet.



Svg Teilauschnitt:

<g>

<!--kbl-id:ID\_CON84;type:ref-->

<g transform="translate(5491.5,82.5) rotate(0)">

<rect x="-59.5" y="-23.5" width="119" height="47" style="stroke:black; stroke-width:0.5; fill:none" />

<text x="0" y="-14.57" style="text-anchor:middle; font-family:Arial; font-size:10px; fill:red">

<tspan x="0" y="-3.071" style="text-anchor:middle">V6 = 8-fach = Cod.A</tspan>

<tspan x="0" y="8.428" style="text-anchor:middle">V10 = 4-fach = Cod.B</tspan>

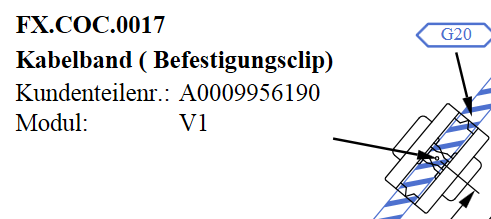
</text>

</g>

</g>

### Fixings

Befestigungselemente und Clips verweisen auf das entsprechende Fixing Objekt in der KBL.



1

2

Die Tabellendarstellung in der SVG:

<g>

<!--kbl-id:ID\_FIX18;type:table-->

<g transform="translate(3415.6,342.39) rotate(0) translate(-3415.6,-342.3)">

<g>

<rect x="3415.622" y="342.399" width="33" height="7" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="3416.455" y="347.155" xml:space="preserve" font-weight="bold" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">FX.COC.0017 </text>

...

1

</g>

</g>

</g>

Die grafische Darstellung des Fixings in der SVG:

<g>

<!--kbl-id:ID\_FIX18-->

<g transform="translate(3501.6,372.3) rotate(129.7) translate(-12,-12)">

<g transform="translate(0,0) scale(1,1) translate(12,12)">

<line x1="-10" y1="-2.5" x2="10" y2="-2.5" style="stroke:black; stroke-width:0.25" />

<path d="M5 -7 A1 1,0,0,0,4 -8" style="stroke:black; stroke-width:0.25; fill:none" />

...

2

</g>

</g>

</g>

Das Fixingelement in der KBL:

<Fixing\_occurrence id="ID\_FIX18">

<Id>ID\_FIX18</Id>

<Alias\_id id="id\_302\_328">

<Alias\_id>FX.COC.0017</Alias\_id>

</Alias\_id>

…

<Part>id\_323\_9</Part>

<Installation\_information id="id\_329\_1145">

<Instruction\_type>ADDITIONAL\_ATTRIBUTE</Instruction\_type>

<Instruction\_value>Additional Attribute</Instruction\_value>

</Installation\_information>

<Installation\_information id="id\_329\_1146">

<Instruction\_type>CustomerDrawNo</Instruction\_type>

<Instruction\_value>A0009956290</Instruction\_value>

</Installation\_information>

<Installation\_information id="id\_329\_1147">

<Instruction\_type>COMMENT\_1</Instruction\_type>

<Instruction\_value>Kabelband ( Befestigungsclip)</Instruction\_value>

</Installation\_information>

</Fixing\_occurrence>

<Fixing id="id\_323\_9">

<Part\_number>A0009956190</Part\_number>

<Version>20110811-NI</Version>

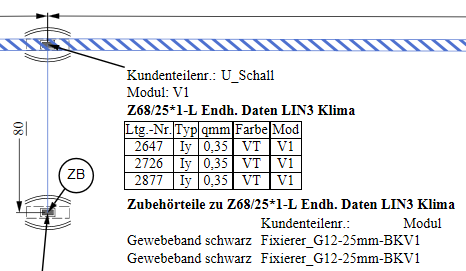
<Abbreviation>P3187</Abbreviation>

<Description>Kabelband ( Befestigungsclip)</Description>

</Fixing>

### Splices

Splices verweisen auf das zugehörige Connector Objekt in der KBL und werden wie sonstige Stecker auch behandelt.



2

3

1

4

Svg Teilauschnitt Symbol:

<g>

Vertex

<!--kbl-id:ID\_BNJ3-->

<g>

<!--kbl-id:ID\_CON128-->

<g transform="translate(2315,5) rotate(179.9) translate(-8.9,-8.0)">

<g transform="translate(0,0) scale(1,1) translate(9,8)">

<line x1="-2.5" y1="1.5" x2="-2.5" y2="-1.5" style="stroke:black; stroke-width:0.18" />

….

1

</g>

</g>

</g>

</g>

Svg Teilauschnitt Tabelle:

<g>

<!--kbl-id:ID\_CON128;type:table-->

<g transform="translate(2342,518) rotate(0)">

<g>

...

<rect x="0" y="12" width="53" height="7" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="0.833" y="16.756" font-weight="bold" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">Z68/25\*1-L Endh. Daten LIN3 Klima</text>

</g>

2

<g>

<rect x="0" y="19" width="17" height="6" style="stroke:black; stroke-width:0.25; fill:none" />

<text x="8.5" y="23.332" style="text-anchor:middle; font-family:CorpoS; font-size:5px; fill:black">Ltg.-Nr.</text>

...

<g>

<!--kbl-id:ID\_WIR1081;type:row-->

...

<text x="8.5" y="29.332" style="text-anchor:middle; font-family:CorpoS; font-size:5px; fill:black">2647</text>

...

</g>

</g>

Svg Teilauschnitt Zubehör:

<g>

<rect x="0" y="45" width="47" height="7" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="0.833" y="49.756" font-weight="bold" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">Zubehörteile zu Z68/25\*1-L Endh. Daten LIN3 Klima</text>

3

....

<g>

<!--kbl-id:ID\_ACC35;type:row-->

<rect x="0" y="58" width="47" height="6" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="0.833" y="62.5" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">Gewebeband schwarz</text>

4

<rect x="47" y="58" width="50" height="6" style="fill:rgb(255,255,255); opacity:0" />

....

</g>

</g>

</g>

</g>

Zugehöriges KBL Element:

<Connector\_occurrence id="ID\_CON128">

<Id>Z68/25\*1-L</Id>

<Alias\_id id="id\_302\_298">

<Alias\_id>Z68/25\*1-L</Alias\_id>

</Alias\_id>

<Description>Endh. Daten LIN3 Klima</Description>

<Usage>splice</Usage>

<Part>id\_315\_38</Part>

<Contact\_points id="id\_372\_821">

<Id>2877\_0</Id>

<Contacted\_cavity>id\_370\_954</Contacted\_cavity>

</Contact\_points>

<Contact\_points id="id\_372\_822">

<Id>2726\_0</Id>

<Contacted\_cavity>id\_370\_954</Contacted\_cavity>

</Contact\_points>

<Contact\_points id="id\_372\_823">

<Id>2647\_0</Id>

<Contacted\_cavity>id\_370\_954</Contacted\_cavity>

</Contact\_points>

….

<Slots id="id\_386\_128">

<Part>id\_341\_1</Part>

<Cavities id="id\_370\_954">

<Part>id\_308\_1</Part>

</Cavities>

</Slots>

</Connector\_occurrence>

### Vertex

Eine einzelne Vertex verweist auf das entsprechende Node Objekt in der KBL.



*<g>*

*<!--kbl-id:ID\_BNJ130-->*

*<circle cx="7250.14" cy="528.65" r="0.5" style="fill:rgb(78,113,207)" />*

*</g>*

Zugehöriges KBL Element:

<Node id="ID\_BNJ130">

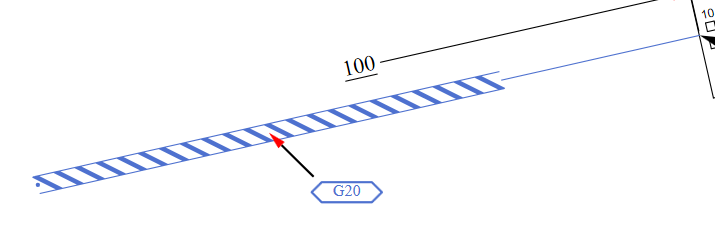
<Id>ID\_BNJ130</Id>

<Cartesian\_point>id\_307\_117</Cartesian\_point>

</Node>

### Segment / Bandierung / Rohr

Segmente werden eigentlich über den darauf liegenden Leitungsschutz dargestellt. Sie verweisen auf das Segment Objekt der KBL (Bundlesegment). Innerhalb der Segmentgruppe verweist der Leitungsschutz auf das entsprechende Wire-Protection Objekt in der KBL. Hier ist ein teilbandierter Abschnitt gezeigt.



2

1

Der SVG Ausschnitt der Segmentdarstellung:

<g>

<!--kbl-id:ID\_BNS6-->

<path d="M5026.643 392.017, l4.837 0, ……" style="stroke:rgb(78,113,207); stroke-width:0.25; fill:none" />

1

<g>

<!--kbl-id:ID\_TAP63-->

<path d="M5026.643 392.016, l33.861 0, .... "style="stroke:rgb(78,113,207); stroke- width:0.25; fill:none" />

2

</g>

</g>

Der zugehörige KBL Abschnitt:

<Segment id="ID\_BNS6">

<Id>ID\_BNS6</Id>

<Virtual\_length id="id\_334\_768">

<Unit\_component>id\_346\_1</Unit\_component>

<Value\_component>150</Value\_component>

</Virtual\_length>

<Physical\_length id="id\_334\_767">

<Unit\_component>id\_346\_1</Unit\_component>

<Value\_component>150</Value\_component>

</Physical\_length>

<End\_node>ID\_BNJ113</End\_node>

<Start\_node>ID\_BNJ35</Start\_node>

<Center\_curve id="id\_306\_6">

<Degree>1</Degree>

<Control\_points>id\_307\_31 id\_307\_103</Control\_points>

</Center\_curve>

….

<Protection\_area id="id\_337\_9">

<Start\_location>0.2</Start\_location>

<End\_location>1</End\_location>

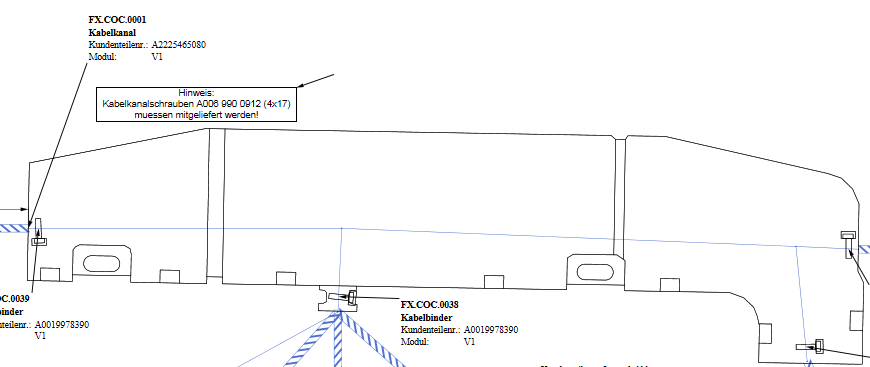
<Associated\_protection>ID\_TAP63</Associated\_protection>

</Protection\_area>

</Segment>

### Kabelkanäle

Kabelkanäle und ähnlich Teile werden wie Fixings behandelt und verweisen daher auf das Fixing Objekt in der KBL.



2

1

Der SVG Ausschnitt der Tabelle:

<g>

<!--kbl-id:ID\_FIX2;type:table-->

<g transform="translate(6839,264) rotate(0) translate(-6839,-264)">

<rect x="6839" y="264" width="33" height="7" style="stroke:rgb(255,255,255); opacity:0; stroke-width:0.25; fill:none" />

<text x="6839.833" y="268.756" xml:space="preserve" font-weight="bold" style="text-anchor:start; font-family:CorpoS; font-size:5px; fill:black">FX.COC.0001 </text>

……

1

</g>

</g>

Der SVG Ausschnitt der Kabelkanal Symbolgrafik:

<g>

<!--kbl-id:ID\_FIX2-->

<g transform="translate(6808,377) rotate(-359.0122) translate(-1,-55)">

<g transform="translate(0,0) scale(1,1) translate(221,60)">

<polyline points="-117.305 22.474 ,63.927 22.474 ……./>

……

2

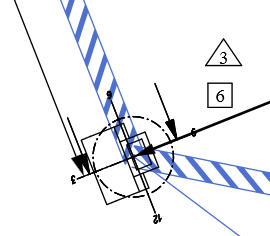
</g>

</g>

</g>

### Uhrzeiten

Die Symbole der Uhrzeiten (Ausbindungsrichtungen) können prinzipiell auf verschiedene Objekte verweisen. Zum Beispiel kann das Uhrzeitensymbol auf den dazugehörigen Clip verweisen oder mit einer Vertex bzw. einem Segment verbunden werden. Sie selber haben kein physikalisches Objekt in der KBL.



1

SVG Teilausschnitt:

<g>

<!--kbl-id:ID\_FIX1-->

<g transform="translate(315.9,589.2) rotate(-89.3) translate(-14,-13)">

<line x1="11.238" y1="-8.838" x2="-0.013" y2="-8.838" style="stroke:black; stroke-width:0.25" />

1

…. Uhrzeitengrafik…

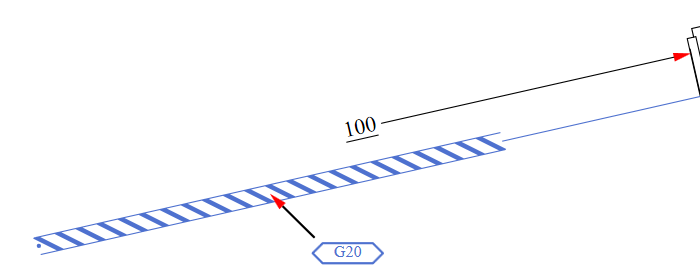
</g>

</g>

## Bemaßungen

Bemaßungen verweisen auf die beiden zugehörigen Vertexes (Start- und Endpunkt) und werden mit dem Typ dimension gekennzeichnet.

Im nachfolgenden Beispiel sind Text und Maßpfeil voneinander getrennt gruppiert worden, da hier unterschiedliche Transformationen angewendet werden. Die Bemaßung kann natürlich auch in einer einzigen Gruppe untergebracht werden.



1

2

SVG Abschnitt einer Bemaßung:

<g>

<!--kbl-id:ID\_BNJ46 ID\_BNJ125;type:dimension-->

<g transform="translate(7769.0,377.61) rotate(329.21) translate(0,-10)">

<text x="0" y="1.332" style="text-anchor:middle; font-family:CorpoS; font-size:5px; text-decoration:underline; fill:black">100</text>

</g>

1

</g>

<g>

<!--kbl-id:ID\_BNJ46 ID\_BNJ125;type:dimension-->

<g transform="translate(7769.01,377.61) rotate(329.21)">

<line x1="104.748" y1="-1" x2="104.748" y2="-11" style="stroke:black; stroke-width:0.25" />

<polygon points="104.748 -10 ,100.748 -11 ,100.748 -9" style="fill:red" />

<line x1="4.952" y1="-10" x2="100.748" y2="-10" style="stroke:black; stroke-width:0.25" />

</g>

2

</g>

## Besonderheiten und implizite Logik bei der Darstellung

Um eine sinnvolle Darstellung des Routings machen zu können, muss der Browser bezüglich der Segmentdarstellung eine besondere Logik anwenden. Segmente, die eigentlich über ihre Modulzugehörigkeit bzw. der Modulzugehörigkeit des entsprechend dargestellten Leitungsschutzes in einer Konfiguration ausgeblendet würden, müssen trotzdem dargestellt werden, wenn es Leitungen gibt, die in diesem Segment laufen und entsprechend ihrer Modulzugehörigkeit aktiv sind.