



AUCOTEC
Create Synergy – Connect Processes

Engineering Base

Formelattribute

Dezember 2019

AUCOTEC AG

Hannoversche Str. 105
D- 30916 Isernhagen
Phone: +49 (0)511 61 03-0
Fax: +49 (0)511 61 40 74

www.aucotec.com

Urheberrecht: Alle Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung, bleiben vorbehalten. Kein Teil dieses Buches darf ohne vorherige schriftliche Zustimmung von **AUCOTEC AG** in irgendeiner Form durch Fotokopie, Mikrofilm oder andere Verfahren reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Haftungsausschluss: Texte und Software wurden mit größter Sorgfalt erarbeitet. Herausgeber und Autoren können für etwaige fehlerhafte Angaben und deren Folgen weder eine juristische noch irgendeine Haftung anders lautender Art übernehmen.

Warenzeichen: Engineering Base® ist ein eingetragenes Warenzeichen der AUCOTEC AG, Deutschland. Microsoft Office Visio®, Microsoft SQL Server und Windows® sind eingetragene Warenzeichen der Microsoft Corporation, USA.

Inhalt

1	Was sind Formelattribute?	1
1.1	Definition der Formeln	1
1.2	Grundlagen für die Erstellung von Formelattributen	4
1.3	Wo werden Formelattribute definiert?	7
2	Definition der Formelattribute.....	8
2.1	Einbindung von Systeminformationen	8
2.1.1	Versionsinformationen	8
2.1.2	Umgebungsinformationen	8
2.1.2.1	Datum und Uhrzeit.....	10
2.1.2.2	Environment Text	11
2.1.2.3	Environment Language	11
2.2	Einbindung von Texten	12
2.3	Einbindung von Attributen	13
2.4	Zugriff auf übergeordnete Objekte (Aggregationen)	16
2.4.1	Parent Objekte	16
2.4.2	Bottom Up	16
2.4.3	Mostup Parent	17
2.4.4	Top Down	18
2.5	Navigationshilfen.....	19
2.5.1	Objekt Navigation	19
2.5.2	Objektnavigation durch Klammerung.....	20
2.6	Zugriff auf assoziierte Objekte (Assoziation).....	22
2.6.1	Relation Rolle	23
2.6.2	Relation Multi	26
2.6.3	Relation Objektgruppen	27
2.7	Einbindung von mathematischen Operationen.....	29
2.8	Objektprüfung	32
2.9	Bedingungen.....	33
2.10	Anhang	37
2.10.1	Formatierung der Ausgabe	37

1 Was sind Formelattribute?

Formelattribute sind individuell definierbare Formeln, über die Sie auf Systeminformationen und Attribute von Objekten zugreifen können. Sie dienen zur kompakten Darstellung von Attributen ‚zugehöriger‘ Objekte. Dabei wird über die Relation des Ausgangsobjekts zu den anderen Objekten der Zugriff auf deren Attribute gesteuert.

Zur Erstellung von Formelattributen sind gute Kenntnisse der EB-Objektstruktur sowie Programmierkenntnisse empfehlenswert.

Formelattribute können in Dialogmasken, in der Shape-Bearbeitung, in Grafiken oder Arbeitsblättern verwendet werden.

Ein Formelattribut ermöglicht:

- Die Navigation innerhalb der Objektstruktur.
- Den Zugriff auf Attribute und das Auslesen der Werte.
- Das Extrahieren von Teilstrings aus einem Textwert.
- Die Erstellung von Bedingungsabfragen
- Mathematische Operationen.
- Den Zugriff auf Versions- und Umgebungsinformationen.

Die Aufgabenstellung:

„Darstellung der an einem Draht angeschlossenen Geräte mit deren Referenzbezeichner, Materialnummer und zugeordneter Funktion (in Klammern)“

kann z. B. mit Formelattributen realisiert werden.

1.1 Definition der Formeln

Der Gesamtstring eines Formelattributes setzt sich aus miteinander verbunden oder verschachtelten Zugriffsbeschreibungen zusammen.

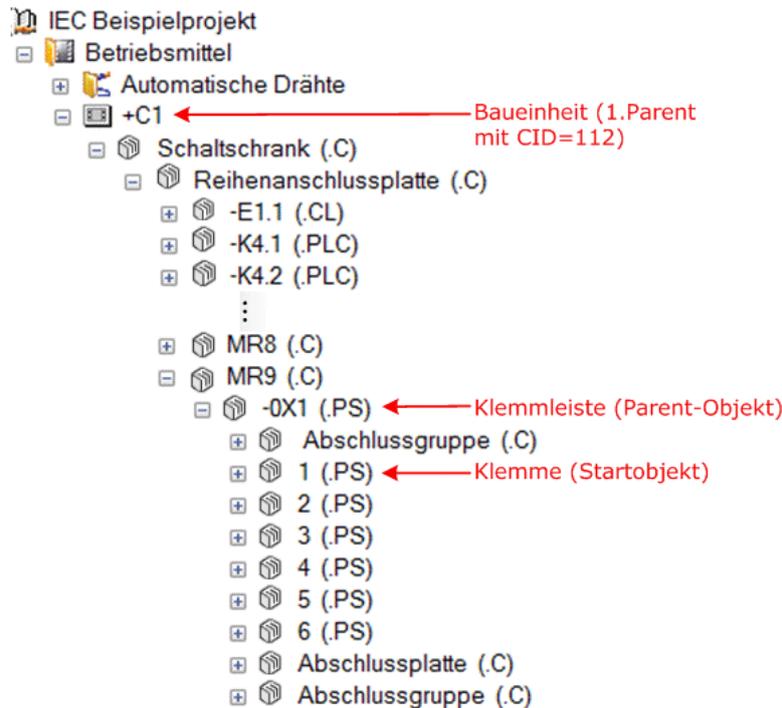
Eine Zugriffsbeschreibung ist vereinfacht folgendermaßen aufgebaut:

Schlüsselkennungsvariablen;

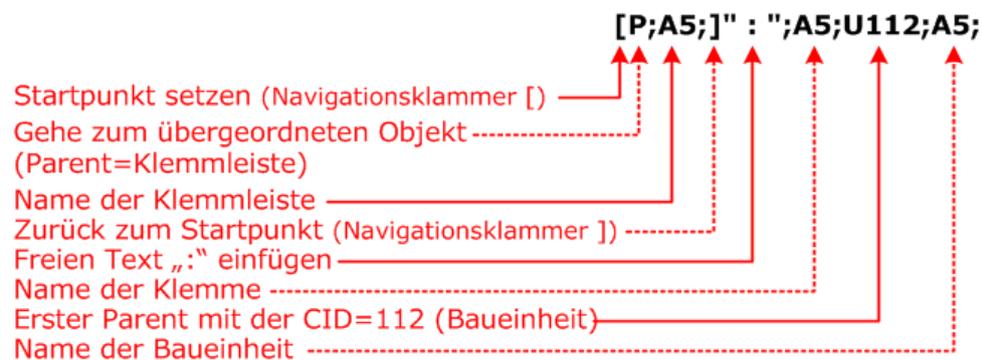
Über die Schlüsselkennung wird die Art des Zugriffs auf Attribute, Attributinformationen oder Systeminformationen gesteuert.

Beispiel:

Zu einer Klemme soll ein Formelattribut erstellt werden, das den Namen der Klemmleiste, der Klemme und der Baueinheit ausgibt.



Verwendete Formel:



Ergebnis der Formel: -0X1 : 1+C1

Folgendes ist bei der Erstellung einer Formel zu beachten:

- Eine Formel kann aus mehreren Anweisungen bestehen.
- Die einzelnen Teile der Formel müssen immer mit Semikolon (;) getrennt werden.
- Die Länge der Formel ist auf 1000 Zeichen begrenzt.
- Am Ende der Formel muss zwingend ein Semikolon (;) stehen.
- Eine Schlüsselkennung besteht in den meisten Fällen aus ein oder zwei Buchstaben.
- Formelattribute können nicht rekursiv genutzt werden.
- Die Auswertung des Formelattributes bricht ab, wenn auf ein gefordertes Objekt nicht navigiert werden kann.

- Bei den Parametern und Variablen ist die Groß- Kleinschreibung der Schlüsselkennungen zu beachten.
- Folder-Objekte bleiben bei der Suche von Parent-Objekten unberücksichtigt. Dies gilt speziell bei den gezählten Navigationsschritten.

Formelattribute müssen unbedingt in einer Testumgebung erstellt und auf ihre Verwendbarkeit geprüft werden! Von Erstellung und Test der Formelattribute in der produktiven Arbeitsumgebung muss dringend abgeraten werden!

1.2 Grundlagen für die Erstellung von Formelattributen

Pro Objekt werden alle Informationen in Attributen (z.B. Gerätebezeichnung, zugeordnete Funktion, etc.) gespeichert. Die Objekte werden über einzelne Attribute bestimmten globalen Typen (z.B. Projekttyp, Gerätetyp, Kabeltyp, etc.) und untergeordneten Typen (Anschlussgruppe, Generator, Motor, etc.) zugeordnet.

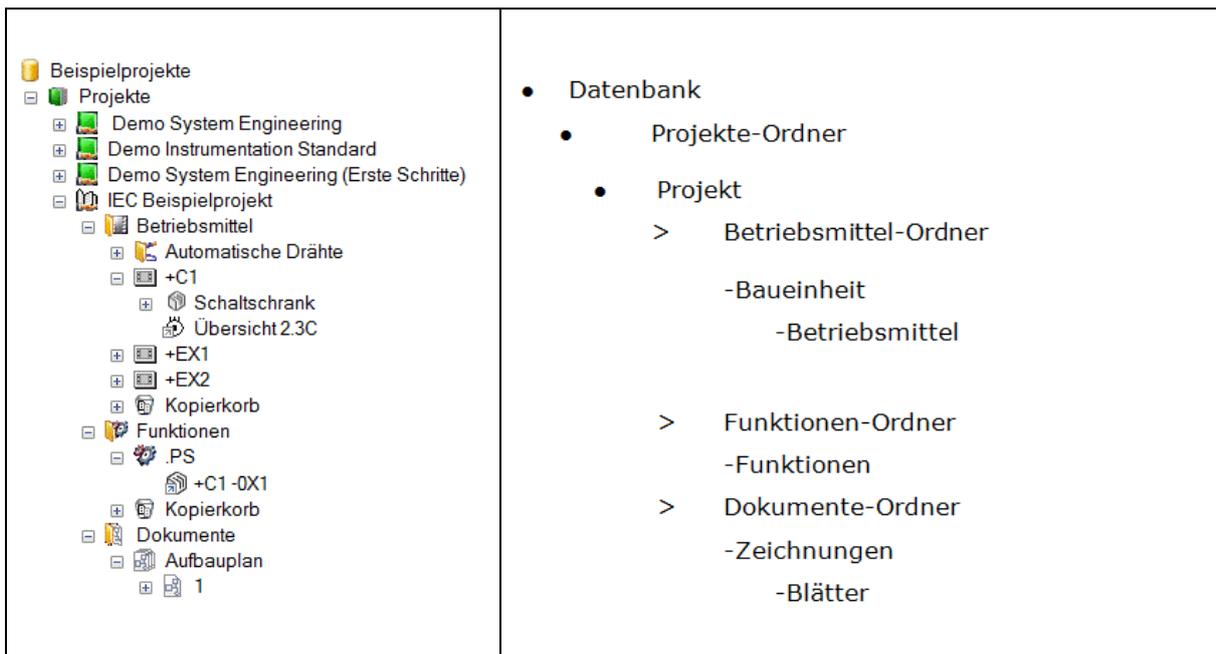
In der Engineering-Base Datenbank werden die Objekte in einer hierarchischen Struktur abgelegt.

Durch diese Struktur ist es möglich, innerhalb einer Datenbank auf die Attribute aller Objekte, die mit dem Ausgangsobjekt in Relation stehen, zu zugreifen. Systeminformationen (z.B. EB-Version, Datum etc.) können ebenfalls angezeigt werden.

Über die hierarchische Struktur kann gezielt auf die Attribute von übergeordneten (Eltern / Parent) oder untergeordneten Objekten (Kind / Child) zugegriffen werden.

Über die globale Typdefinition, die Cover ID (CID), und die Typdefinition, Type ID (TID), kann gezielt auf Attribute bestimmter Objekttypen zugegriffen werden.

Aufbau der Objekt-Hierarchie:



Beziehung zwischen den Objekten:

- **Kind** (Child): Objekt, das dem Ausgangsobjekt untergeordnet ist (bspw. sind Betriebsmittel Kinder der Baueinheit).
- **Eltern** (Parent): Objekt, das dem Ausgangsobjekt übergeordnet ist (bspw. ist die Baueinheit ein Eltern-Objekt eines Betriebsmittels).

Verknüpfung zwischen Objekten sind Relationen über Rollen (bspw. die Beziehung zwischen Anschluss und Draht). Relationen haben einen Start- und Endpunkt, also eine Richtung.

Beim Zugriff auf assoziierte oder übergeordnete Objekte ist eine Navigation „vor“ und „zurück“ möglich. Multi-Ergebnisse können erfasst werden.

Wird auf ein assoziiertes oder übergeordnetes Objekt zugegriffen, wird dieses zum „aktuellen“ Objekt und alle folgenden Attributfunktionen werden auf diesem Objekt ausgeführt.

Globale Typdefinition (CID):

Jedem Objekt sind bestimmte Attribute zugeordnet. Die Zuordnung der Attribute wird über die globale Typdefinition (CID) gesteuert.

- ☐  Typdefinitionen
 - +  Anschlussvorschriftentypen
 - +  Anmerkungstypen
 - +  Anschlusstypen
 - +  Aufgabentypen
 - +  Aufstellungsorttypen
 - +  Baueinheitstypen
 - +  Blatttypen
 - +  Chemische Komponententypen
 - +  Chemische Substanztypen
 - +  Dokumententypen
 - +  Drahttypen
 - +  Funktionstypen
 - +  Geometrie Spezifikationstypen
 - +  Gerätetypen
 - +  Kabeltypen
 - +  Massenstromtypen
 - +  Potenzial-/Stofftypen
 - +  Projekttypen
 - +  Prozesstypen
 - +  Punktypen
 - +  Revisionstypen
 - +  Rohrleitungssegmenttypen
 - +  Rohrleitungstypen
 - +  Segmenttypen
 - +  Typen von Modulen und Varianten
 - +  Zeichnungstypen
 - +  Zustand- / Kenngrößentypen
 - +  Änderungsmaßnahmentypen

Pro globaler Typdefinition wird eine Cover ID (CID) vergeben.

Beispiele für CIDs:

Betriebsmittel	7
Projekt	10
Baueinheit	112
Geräte	113

[Eine Liste der verwendbaren CIDs](#) finden Sie unter [References](#) in der EB-Web-Help.

In Engineering Base finden neben den o. g. Typdefinitionen weitere CIDs für interne Zwecke Verwendung.

Typdefinition (TID):

Die globalen Objekttypen (CID) werden weiter in Objekttypen (TID) untergliedert.

- ☐  Gerätetypen
 - +  Anschlagmaterial
 - +  Anschlussgruppe
 - +  Antriebssteuerung
 - +  Automatisierungsgeräte
 - +  Baugruppe, Schrank
 - +  Bauliche Einrichtungen
 - +  Beleuchtung, Heizung, Kühlung
 - +  Brücke
 - +  Clip
 - +  E-Komponente
 - +  Elektrisch angetriebenes mechanisches Gerät
 - +  Entstörgerät
 - +  Generator
 - +  Gerätekanal
 - +  Halbleiter
 - +  Hochspannung
 - +  Induktor, Reaktor
 - +  Instrument, Zähler, Messgerät
 - +  Kabelkanal
 - +  Kammer
 - +  Klemmen und Leisten
 - +  Kommunikationselement

Objekttypen des globalen Objekttyps Gerätetypen (CID 113)

Pro Objekttyp wird eine Type ID (TID) vergeben.

Beispiele für TIDs:

Motor	123
Klemme	136
PE	141

[Eine Liste der verwendbaren TIDs](#) finden Sie unter [References](#) in der EB-Web-Help.

1.3 Wo werden Formelattribute definiert?

1. Öffnen Sie mit Engineering Base eine **Datenbank zum Testen von Formelattributen** oder erstellen Sie dazu eine separate Datenbank.
2. Klicken Sie im Engineering Base-Explorer auf **Datenbank**.
3. Wählen Sie den Ordner **Attribute** aus.
4. Zeigen Sie im Kontextmenü auf **Neu** und klicken Sie dann auf **Attribut**.

Der Dialog **Neues Attribut** wird geöffnet.

5. Tragen Sie in das Feld **Bezeichnung** den Namen des neuen Attributes ein. Mit STRG+F3 kann ein Eintrag aus dem DB-Wörterbuch gewählt werden oder bei Bedarf ein neuer Eintrag erstellt werden.
6. Markieren Sie **Formel** unter **Attributtyp**.
7. Tragen Sie die gewünschte Formel in das nebenstehende Feld ein.
8. Klicken Sie **Ok**, um das neue Attribut zu erstellen.
9. Starten Sie Engineering Base neu.
10. Testen Sie das Formelattribut vor der Übernahme in die produktive Datenbank mit den von Ihnen gewünschten Anwendungsfällen.

Das neu definierte Formelattribut kann nun verwendet werden.

2 Definition der Formelattribute

Die nachfolgenden Detailbeschreibungen für Formelattribute basieren auf der Engineering Base-Version 6.4.1. In älteren Versionen können einige Sprachelemente noch nicht implementiert sein.

Legende der Syntaxbeschreibung:

Fettschrift	Literale (Schlüsselkennungen, Variablen und zwingend notwendige Zeichen einer Anweisung (;, {, }, ...).
<i>Kursiv</i>	Anwendereingaben werden kursiv geschrieben
< Begriff >	Ein Begriff in spitzen Klammern wird als symbolischer Platzhalter für eine Menge verschiedener Werte verwendet, deren Bedeutungen später im Detail erklärt werden.
[]	Umschließt die optionalen Teile einer Anweisung.
	Trennzeichen (Exklusiv-Oder) zwischen alternativen Eingabewerten.

2.1 Einbindung von Systeminformationen

2.1.1 Versionsinformationen

Schlüsselkennung V	Version
Syntax: V [v] [b];	
V;	Engineering Base V ersionsnummer. B uildnummer.
Vv;	Engineering Base V ersionsnummer.
Vb;	B uildnummer der Engineering Base V ersion.

2.1.2 Umgebungsinformationen

Schlüsselkennung Eu	Environment U ser - Benutzerinformationen
Syntax: Eu [f s n c <AccountSelData>];	
Euf;	Full Name - Benutzername (default).
Eus;	Short Name - Benutzerkurzname
Eun;	Windows Login- N ame
Euc;	Computer Name.

Über < AccountSelfData > können erweiterte Benutzerkontodaten angezeigt werden.	
Eu0;	NameFullyQualifiedDN Name, eindeutig vollständig qualifiziert Bspw. CN=Paula Mustermann, OU=User, OU=AUCOTEC, DC=aucotec,DC=com.
Eu1;	NameSamCompatible Account-Name, Bspw. AUCOTEC/PMu.
Eu2;	NameDisplay Name, Bspw. Paula Mustermann.
Eu5;	NameUniqueId Eindeutige Kennung des Namens Bspw. {edf3accf-47d2-4d5f-9d49-efea0f4ea8a18}.
Eu6;	NameCanonical Name kanonisch, Bspw. aucotec.com/OU-Users/AUCOTEC_DE/User/OU-GS/Paula Mustermann.
Eu7;	NameUserPrincipal Bspw. PMu@aucotec.com.
Eu8;	NameCanonicalEx Name kanonisch erweitert, Bspw. aucotec.com/OU-Users/AUCOTEC_DE/User/OU-GS/ Paula Mustermann.

2.1.2.1 Datum und Uhrzeit

Schlüsselkennung Ed	Environment date and time - Datum und Uhrzeit	
Syntax:		
Ed [d D t]	["<Zeitformatbestimmung>"];	
d	Datum Kurzformat auf Basis der aktuellen Einstellungen unter Systemsteuerung / Region und Sprache.	
D	Datum Langformat auf Basis der aktuellen Einstellungen.	
t	Zeit auf Basis der aktuellen Einstellungen.	
Zeitformatbestimmung	%x	Datum im Format der aktuellen Ländereinstellung, bspw. TT.MM.YYYY.
	%a	Wochentag in Kurzform bspw. Mi für Mittwoch.
	%A	Wochentag.
	%d	Tag als Dezimalzahl (TT, 01-31).
	%j	Tag als Dezimalzahl (001-366).
	%w	Wochentag als Dezimalzahl (0-6, Sonntag =0).
	%b	Monat in Kurzform, bspw. Aug für August.
	%B	Monat.
	%m	Monat als Dezimalzahl (01-12).
	%y	Jahr in Kurzform, bspw. 14 für 2014.
	%Y	Jahr.
	%U	Wochennummer (00-53, Sonntag als erster Wochentag).
	%W	Wochennummer (00-53, Montag als erster Wochentag).
	%c	Datum und Uhrzeit (TT.MM.YYY hh.mm.ss)
	%H	Stunden im 24-Stunden-Format (00-23).
	%h	Stunden im 12-Stunden-Format (01-12).
	%M	Minuten als Dezimalzahl (0-59).
%S	Sekunden als Dezimalzahl (0-59).	
%X	Uhrzeit im Format der aktuellen Ländereinstellung bspw. hh.mm.ss.	
%z, %Z	Zeitzone bspw. Mitteleuropäische Zeit.	
%%	Prozentzeichen.	

<u>Beispiel:</u>	
<u>Formel:</u>	<u>Wert des Formelattributes (Zeichenfolge):</u>
Ed " %x";	Datum im Format der aktuellen Ländereinstellung, bspw. TT.MM.YY.
Edt " %X";	Uhrzeit im Format der aktuellen Ländereinstellung, bspw. hh.mm.ss.

2.1.2.2 Environment Text

Schlüsselkennung Et	Environment Text. Steuert die Reihenfolge der Ausgabe des Formelattributs, wenn es aus mehr als einem Ergebniswert besteht. Somit lassen sich beispielsweise Werte vom Objekt hierarchisch aufwärts ermitteln und trotzdem in anderer Reihenfolge (top down) anzeigen, ohne die Hierarchie mehrmals zu durchlaufen.
Syntax:	
Et l r;	
Etl;	Ausgabe erfolgt von rechts nach links.
Etr;	Ausgabe erfolgt von links nach rechts (default).

2.1.2.3 Environment Language

Schlüsselkennung EI	Unterdrückung der Übersetzung (L anguage). Schaltet die Übersetzung enthaltener Translatelemente für die gesamte Formel aus. Enthaltene Referenzen werden in Form <l:ttt-dd> geliefert, z.B. <0:100-24> wobei l = Auswahl der Sprache via Translateeinstellung des Projektes ttt = Textnummer -dd = Wörterbuch ID; (kann bei globaler Referenz fehlen).
Syntax:	
EI;	

2.2 Einbindung von Texten

Schlüsselkennung " "	Eingabe von freiem Text (Zeichenfolgen) oder Translatereferenzen	
Syntax: "<freierText> [<Translate-Referenzen>] [\<Steuerzeichen>]";		
"freier Text";	Zwischen die Hochkommata kann eine beliebige Zeichenfolge eingegeben werden, die in das Ergebnis des Formelattributes eingefügt wird.	
	Translate-Referenzen können über das ‚Display-Format‘ <0:4711> eingefügt werden.	
	<table border="1"> <tr> <td>\Steuerzeichen</td> <td>Steuerzeichen sind vordefinierte Sonderzeichen oder direkte Charactercodes die die Ausgabe beeinflussen. Z.Zt. wird nur die Eingabe des Character-Codes in hexadezimaler Form unterstützt (max. 4 Stellen, führende Nullen sind erlaubt, entsprechen den Character-Codes von Windows).</td> </tr> </table>	\Steuerzeichen
\Steuerzeichen	Steuerzeichen sind vordefinierte Sonderzeichen oder direkte Charactercodes die die Ausgabe beeinflussen. Z.Zt. wird nur die Eingabe des Character-Codes in hexadezimaler Form unterstützt (max. 4 Stellen, führende Nullen sind erlaubt, entsprechen den Character-Codes von Windows).	

<u>Beispiele für Formeln:</u>	
<u>Formel:</u>	<u>Wert des Formelattributes (Zeichenfolge):</u>
"Test Text <0:100>";	„Test Text Geräteliste (strukturiert)“.
"Test Text\0A";	„Test Text“ mit nachfolgender Zeilenschaltung.

2.3 Einbindung von Attributen

Für die Einbindung von Attributen muss die Attribut-ID (*aid*) des gewünschten Attributes bekannt sein.

Die Attribut-ID kann in den Objekteigenschaften, den Attributeigenschaften angezeigt oder auch in der Listenansicht aller Attribute eingeblendet werden

Schlüsselkennung A	Einfacher Attributzugriff (skalare Attribute)	
Syntax:		
A [<modif>] ["<Präfix>"] < aid > ["<Postfix>"]		
[([<numchar> [:<poschar>]] [[Präfix F] "<Feldformat>"])];		
modif	Selektor für spezielle vordefinierte Attribute	
	F	AF ["<Präfix>"] aid ["<Postfix>"] (<Wert>); Test, ob an dem angegebenen Attribut Flags (manuell, automatisch, ...) gesetzt sind. Ergebnis: 1 oder Präfix wenn das Flag gesetzt ist. 0 oder Postfix wenn das Flag nicht gesetzt ist.
	Wert	Kennnummer des Flags 2 = Attribut ist read only 4 = Der Attributwert ist leer 256 = manuell eingetragener Wert 1024 = Attributwert aus Katalog 2048 = Translate-Text vorhanden 8192 = Attribut ist Schreibgeschützt
	j	Aj_{aid} ; Zugriff auf Attribute des Projektes; das Projektobjekt wird zum aktuellen Objekt.
	J	AJ_{aid} ; Zugriff auf Attribute des Projektes; das aktuelle Objekt bleibt unverändert.
	r	Ar ; klassischer Referenzbezeichner des Objektes. (mit Teil von..).
	R	AR ; Referenzbezeichner des Objektes mit Pfadausdruck (mit Ordner).
	o oder 1	Ao ; oder A1 ; Objekt-ID des Objektes.
	c oder 2	Ac ; oder A2 ; Cover-ID des Objektes.

	t	At; Typ-ID des Objektes.
	c	AC_{aid}; Kommentarattribut..
Präfix / Postfix		Zeichenfolge, die im Falle eines nicht-leeren Attributwertes vor bzw. nach dem Attributwert eingefügt wird.
aid		Attributreferenz (numerisch oder vordefinierte C, r, R).
numchar		Anzahl der Zeichen, die aus dem ermittelten Wert des Formelattributs angezeigt werden sollen (>0 vom Anfang; <0 vom Ende beginnend).
poschar		Zeichenposition, ab der zu beginnen ist.
Feldformat		spezifiziert die exakte Zeichenfolgenlänge, inklusive einzufügender Ersatzzeichen. Für Präfix F = „-“, „e“ und „f“ ergibt sich eine andere Bedeutung (siehe unten).
Präfix F		Präfix für Feldformat.
	-0	führende Nullen werden gestrichen.
	-	Die Zeichen in „Feldformat“ werden aus dem Wert des Formelattributes gestrichen.
	e	„Feldformat“ wird nur genutzt, wenn der Wert des Formelattributes nicht leer ist.
	f	„Feldformat“ wird als c-konformer Formatstring interpretiert. Die möglichen Formatierungen sind im Anhang aufgeführt.

<u>Beispiele für Formeln mit Attributen:</u>	
<u>Formel:</u>	<u>Wert des Formelattributes (Zeichenfolge):</u>
A"Präfix"aid"Postfix";	Präfix/Postfix = Zeichenfolge, die im Falle eines nicht-leeren Attributwertes vor bzw. nach dem Attributwert eingefügt wird.
Aaid(Anzahl);	Anzeige einer bestimmten Anzahl von Zeichen(>0 vom Anfang; <0 vom Ende beginnend).
Aaid(n:x);	Anzeige von n Zeichen ab Position x.
Aaid("0000000000")	Anzeige des Attributwertes, Auffüllen mit „0“ auf 10 Zeichen.
Aaid(-n- '0'-----);	Wert des Attributs, die letzten n Zeichen verwenden, führende Nullen streichen und mit „-“ auf 8 Zeichen auffüllen.
Aaid(n-"Wert");	Wert des Attributs, die ersten n Zeichen verwenden und die Zeichen streichen, die „Wert“ entsprechen.

Aaid(f"%8.2f");	Ausgabe des Attributwerts nach C-Syntax formatiert Hier Fließkommazahl mit 8 Vorkomma- und 2 Nachkommastellen.
Das Attribut mit der ID= 5 enthält den Objektnamen, er sei hier 00123456.	
A5("0000000000"); Wert des Attributs mit der ID = 5, Auffüllen mit „0“ auf 10 Zeichen.	Wert des Attributs mit der ID = 5, Auffüllen mit „0“ auf 10 Zeichen. „0000123456“
A10212;"-";A10386;"-";A10175;	Werte der aufgeführten Attribute mit „-“ getrennt anzeigen. Planart-Blattformat-Gezeichnet von
A5(4-'0);	Wert des Attributs mit der ID=5, die ersten 4 Zeichen verwenden und führende Nullen streichen „12“.
A5(4-'0'-----);	Wert des Attributs mit der ID=5, die ersten 4 Zeichen verwenden, führende Nullen streichen und mit „-“ auf 8 Zeichen auffüllen. „-----12“
A5(-4-'0'-----);	Wert des Attributs mit der ID=5, die letzten 4 Zeichen verwenden, führende Nullen streichen und mit „-“ auf 8 Zeichen auffüllen. „----3456“
A5(6-"0012");	Wert des Attributs mit der ID=5, die ersten 6 Zeichen verwenden und die Zeichenfolge „0012“ streichen. „34“
A5(3:5);	Wert des Attributs mit der ID=5, ab Position 5 werden 3 Zeichen angezeigt. „456“
AF"manuell"5(256);	„manuell“, wenn das Attribut mit der ID = 5 manuell eingetragen wurde.
A("102875")("f"%2f);	Attribut 102875 sei eine Fließkommazahl mit dem Wert -47,1256 „(-47,13)“ Der Wert wird auf 2 Nachkommastellen gerundet.
A102875(f"%0 10.3f");	„-000047,136“ Der Wert wird als 10-stellige Fließkommazahl dargestellt, es werden führende Nullen eingefügt.
A102675(f"%6d");	Attribut 102675 sei eine Ganzzahl (Integer) mit dem Wert 4711. „004711“ Das Ergebnis soll 6-stellig sein, es werden daher führende Nullen eingefügt.

2.4 Zugriff auf übergeordnete Objekte (Aggregationen)

Zu einem Objekt können Informationen aus einem übergeordneten Objekt (Parent) verarbeitet werden.

2.4.1 Parent Objekte

Schlüsselkennung P	Access to P arent Objects - Zugriff auf Parent-Objekte
Syntax: P [j] ;	
P;	Zugriff auf direktes Parent-Objekt.
Pj;	Zugriff auf das Projektobjekt.

<u>Beispiele für Formeln mit Zugriff auf Parent Objekte:</u>	
<u>Formel:</u>	<u>Wert des Formelattributes (Zeichenfolge):</u>
P;A5;	Zeigt den Namen des Parent-Objekts an.
Pj;A5(2);	Zeigt die ersten 2 Zeichen des Projektnamens an.
Pj;Ar;	Gibt den Referenzbezeichner des Projekts an.

2.4.2 Bottom Up

Schlüsselkennung U	Bottom U p - Zugriff auf erstes Parent-Objekt einer CoverID / Typ (bottom-up), also auf das erste übergeordnete Objekt mit der definierten ID (TID oder CID).
Syntax: U [t] [<id>] [(<aid> : ("<valuetext>" <aidvalue>))] ;	
t	Auswahl über Typ (und nicht CID).
id	Definition des relevanten Parentobjektes (CID oder TID).
aid	Zusätzliche Prüfung auf Attributwert.
valuetext	Testwert (Text) in Hochkommata "Testwert." .
aidvalue	Attribut-ID, deren Wert am Startobjekt den Testwert bestimmt.

<u>Beispiele für Formeln mit Zugriff auf erstes Parent Objekt:</u>	
<u>Formel:</u>	<u>Wert des Formelattributes (Zeichenfolge):</u>
U_{cid};	Definition des Parent-Objektes mit der entsprechenden CID.
Ut_{tid};	Definition des relevanten Parent-Objektes mit entsprechender Typ ID.
U₁₁₃;A5;	Zeigt den Namen des ersten Geräts oberhalb des aktuellen Objekts an (CID Gerät =113).
Ut₁₂₃;A5;	Zeigt den Namen des ersten Motors oberhalb des aktuellen Objekts an (Typ ID Motor =123).

2.4.3 Mostup Parent

Schlüsselkennung M	Zugriff auf das MostUp-Parent-Objekt einer CoverID / Typ (bottom-up), also auf das oberste übergeordnete Objekt mit der vorgegebenen CoverID. M gibt das aktuelle Objekt zurück, falls dieses die Bedingung erfüllt und oberhalb kein Objekt existiert, das die Bedingung erfüllt.
Syntax: M [t] <id>;	
t	Auswahl über Typ (und nicht CID).
id	Definition des relevanten Parentobjektes (CID oder TID).

Beispiele für Formeln mit Zugriff auf oberstes übergeordnetes Parent Objekt:	
Formel:	Wert des Formelattributes (Zeichenfolge):
M_{cid};	Zugriff auf das oberste Parent-Objekt mit der angegebenen CID. Es gibt keine Einschränkungen, welche Objektarten in der Parent-Kette vorhanden sein dürfen (Bsp. M223 greift auf das oberste Parent-Objekt aller Kabel zu).
Mt_{tid};	Zugriff auf das oberste Parent-Objekt mit der angegebenen Typ ID (Bsp. Mt1101 greift auf das oberste Parent-Objekt mit dem Typ Multi-Core Kabel zu).
M₁₁₃;A₅;	Zeigt den Namen des obersten Geräts oberhalb des aktuellen Objekts an (CID Gerät =113).
Mt₁₂₃;A₅;	Zeigt den Namen des obersten Motors oberhalb des aktuellen Objekts an (Typ ID Motor =123).

2.4.4 Top Down

Schlüsselkennung T	Top down Navigation - Navigiert vom Referenzobjekt aus um eine bestimmte Anzahl Parent-Objekte nach oben oder unten.
Syntax: T [s] <counter> [: <cid>];	
s	Innerhalb der Aggregationskette dürfen beliebige Objektarten vorkommen. Default: Die Hierarchiekette darf nur durch Ordner-Elemente strukturiert sein, ansonsten bricht die Navigation ab.
counter	Schrittzahl für die Navigationstiefe (>0 nach oben; <0 nach unten)
cid	Relevante Objektart

<u>Beispiele für Formeln mit Zugriff über Top Down Navigation:</u>	
<u>Formel:</u>	<u>Wert des Formelattributes (Zeichenfolge):</u>
T_n;	Navigiert zum Parent-Objekt der n-ten Stufe nach oben.
T_{-n};	Navigiert zum Parent-Objekt und von da aus in der n-ten Stufe nach unten.
T_n:cid;	Es wird in der Hierarchiekette der Parent-Objekte (mit der vorgegebenen CID) zum Parent-Objekt der n-ten Stufe nach oben navigiert.
Ts_n:cid;	Es wird in der Hierarchiekette der Parent-Objekte (mit der vorgegebenen CID) zum Parent-Objekte der n-ten Hierarchiestufe nach oben navigiert. Innerhalb der Hierarchiekette dürfen beliebige Objektarten vorkommen.
T1:111;A5;	Zeigt den Namen (A5) des Ordners (CID 111 = Ordner). der 1. Hierarchiestufe (1) an.

2.5 Navigationshilfen

2.5.1 Objekt Navigation

Schlüsselkennung O	Object Navigation - Navigationscode Hiermit kann ein Objekt (Objektreferenz) für die weitere Bearbeitung markiert werden. Je nach verwendeter Schlüsselkennung wird nach einer ungültigen Navigation bei der gespeicherten Objektreferenz wieder aufgesetzt. Es steht nur ein Navigationsspeicher zur Verfügung!
Syntax:	
O [c C m s S j J I L g GE];	
c	Markierung des Wiederaufsetzpunktes löschen (global).
C	Markierung des Wiederaufsetzpunktes innerhalb einer Multi-Behandlung löschen.
m	Markierung des Wiederaufsetzpunkts nach einer Multi-Behandlung.
s	Setzt einen Navigationspunkt, die Multibehandlung wird unterbrochen. Rückkehr zu diesem Punkt mit Oj.
S	Setzt einen Navigationspunkt. Eine Multi-Behandlung wird nicht unterbrochen.
j	Rücksprung (J ump) zu mit Os gesetzten Navigationspunkt. Die Multi-Behandlung wird unterbrochen.
J	Rücksprung (J ump) zu mit OS gesetzten Navigationspunkt. Eine Multi-Behandlung wird nicht unterbrochen.
I	Setzt eine Sprungmarke (für L oops).
L	Löschen der Sprungmarke.
g	Gehe zur Sprungmarke (OI) (entspricht „goto“). Die Verarbeitung wird an der Marke OI weiter geführt.
GE	Auswertung der gesamten Formel wird beendet („ g oto e nd“).

2.5.2 Objektnavigation durch Klammerung

Die eckigen Klammern müssen paarweise ‚auf gleicher Ebene‘ gesetzt werden. Sie umspannen einen ganzen Bedingungs Ausdruck oder liegen vollständig innerhalb einer Bedingung. Ein Öffnen **vor** der Bedingung und ein Schließen **in** der Bedingung sind nicht möglich

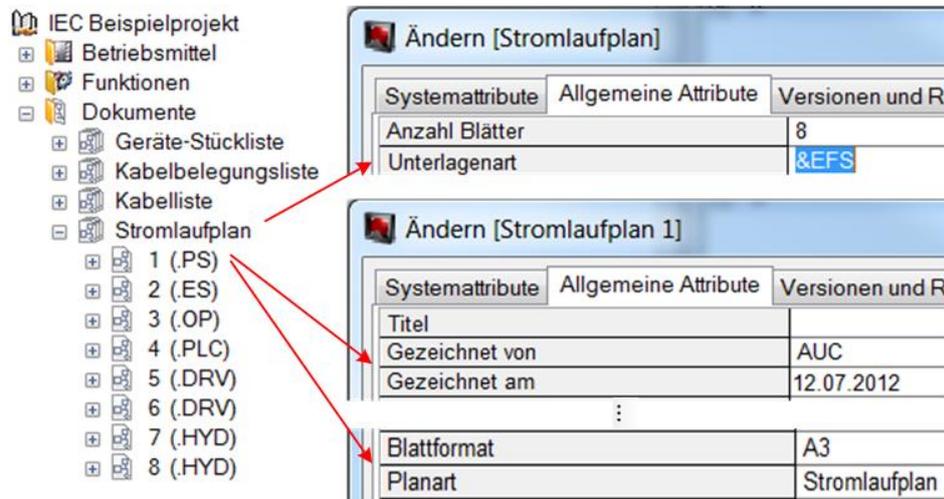
Die Klammerung funktioniert analog zu Os ... Oj (Navigationspunkt setzen und Rückspringen). Eine aktive Multi-Behandlung wird an dieser Stelle grundsätzlich **nicht** abgebrochen.

Schlüsselkennung []	Navigationsklammerung Behandlung des Navigationsobjektes durch ‚Klammerung‘; d.h. erreicht die Auswertung die schließende Klammer „]“, so wird das Navigationsobjekt auf den Anfangswert zurückgesetzt und mit der weiteren Auswertung der Statements fortgefahren.
Syntax:	
[[+] <statement sequence>];	
+	Ausführung am Ende der Klammerung fortführen, auch wenn innerhalb der Klammerung eine Objektnavigation fehlschlägt.

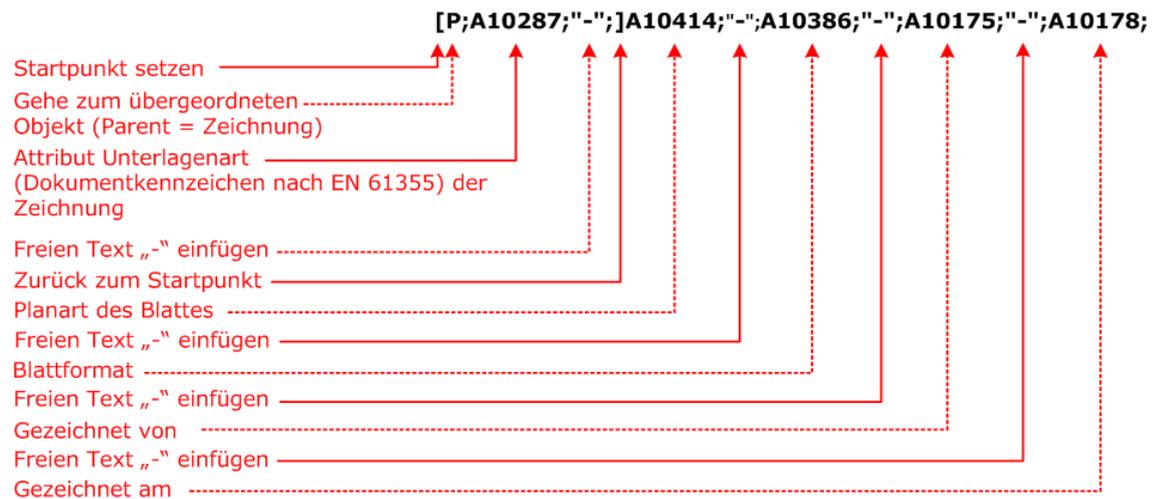
<u>Beispiele für Objektnavigation:</u>	
<u>Formel:</u>	<u>Wert des Formelattributes (Zeichenfolge):</u>
Rb ₁₀₀ ; [P ; A ₅ ;]A ₅ ;	Navigation zur Funktion eines Objektes. Innerhalb der Klammer wird zum Parent der Funktion navigiert und die entsprechende Designation (Name =Attribut-ID=5) ausgegeben. Danach wird der Name (Attribut-ID=5) der Funktion ausgegeben.

Aufgabe

Es soll eine zusammengesetzte Zeichnungsnummer im Blatt erstellt werden, die aus dem Dokumentkennzeichen nach der EN 61355 und den Attributen Planart, Blattformat, „Gezeichnet von“ und „Gezeichnet am“ bestehen soll. Die einzelnen Werte sollen mit „-“ getrennt sein. Das Dokumentkennzeichen ist in der übergeordneten Zeichnung im Attribut Unterlagenart enthalten.



Verwendete Formel:



Ergebnis der Formel: &EFS-Stromlaufplan-A3-AUC-12.07.2012

2.6 Zugriff auf assoziierte Objekte (Assoziation)

Zugriff auf Informationen eines Objekts , das mit dem aktuellen Objekt assoziiert ist.

Eine Assoziation zeigt von einem Objekt (Ursprung/Source) auf eventuell viele andere (Ziele/Targets). Beispielsweise kann ein Ort mehreren Blättern zugewiesen werden aber nicht anders herum. Der Ort ist in diesem Fall der Ursprung und die Blätter das Ziel der Assoziation. Assoziationsstypen unterscheiden sich durch ihre Rolle.

[Eine Liste der verwendbaren Rollen](#) finden Sie unter [References](#) in der EB-Web-Help.

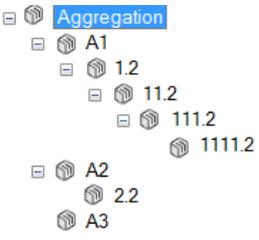
Beispiel für Assoziationsrollen:

Rolle	ID
Funktion zu Gerät	100
Funktion zu Blatt	106
Projekt zu Katalog	40
Objekt zu Kindobjekt	-3
Ersatzrollen für die Multi-Behandlung von Ader- und Kabelzielen. Diese Ersatzrollen sind nur in Formelattributen verfügbar.	
Sammelrolle für 108/109 Aderziele	1104
Sammelrolle für 110 Kabelziele	1105
Ersatzrolle für 110 Kabelziele links	1106
Ersatzrolle für 110 Kabelziele rechts	1107

2.6.1 Relation Rolle

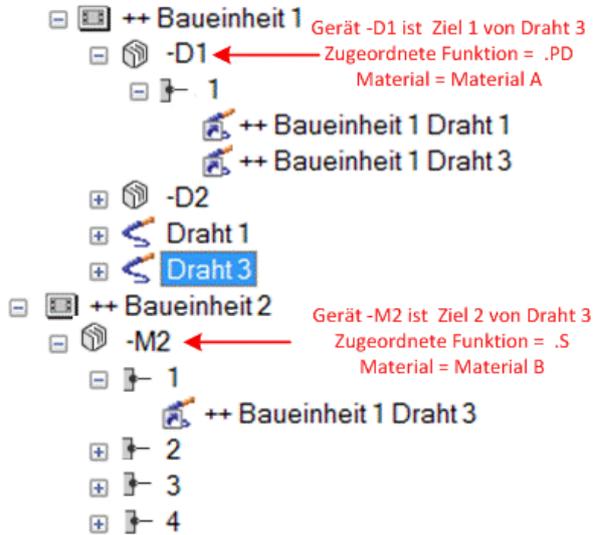
Schlüsselkennung R	Access via Role - Zugriff über Ziel-Assoziationen (R) oder Quell-Assoziation (Rb).
Syntax: R [b] [cs] [t] ([#] [s]) (<role> O) [(<depth>)] [: [f F] <assocflag>] [: <cid>] [: + d + f <calculation> "<formatstring>"] [" <text> "];	
b	Zugriff auf assoziiertes Objekt über Quell-Assoziation (base).
t	Zugriff auf assoziiertes Objekt über Ziel-Assoziationen mit Einschränkung auf den Typ, <cid> steht hier für den Objekttyp (tid).
cs	Zugriff auf die ‚Geschwister‘ (collect sibling).
#	Gibt die Anzahl der assoziierten Objekte an. Das Navigationsobjekt bleibt erhalten.
s	Sortierung (nach Standardsortierregel von EB).
role	Assoziationsrolle.
O	Auflistung der Elemente einer aktiven Objektgruppe (siehe Objektgruppen GO).
depth	In Verbindung mit cs: Navigation zum Geschwisterobjekt (<0 zum Vorgänger; >0 zum Nachfolger).
assocflag	Relationen, die die angegebenen Relationsflags haben (: f) oder nicht haben (: F). Definierte Relationsflags: 1 = die Assoziation weist einen Konflikt auf 2 = Manuell vergebene Assoziation 32 = die Assoziation ist Readonly 64 = die Assoziation ist negativ (besondere kontextabhängige Bedeutung)
cid	Einschränkung auf Objekte der spezifizierten cid.
+d	Berechnung in Ganzzahlarithmetik (long).
+f	Berechnung in Dezimalzahlarithmetik (double).
formatstring	Ausgabestring mit Formatierungsinformationen Die möglichen Formatierungen sind im Anhang aufgeführt.
text	Text, der zur Trennung von Mehrfachergebnissen verwendet werden soll.

Syntax von <calculation>: (<aid> [<charpos>] #) [[* + - /] [p] <aid> #]	
aid	Attributs-ID.
charpos	Startindex beim Einlesen von Zahlen aus Textattributen (0-basiert).
#	In Verbindung mit #-Operator die Anzahl der Objekte.
* + - /	Mathematische Operatoren.

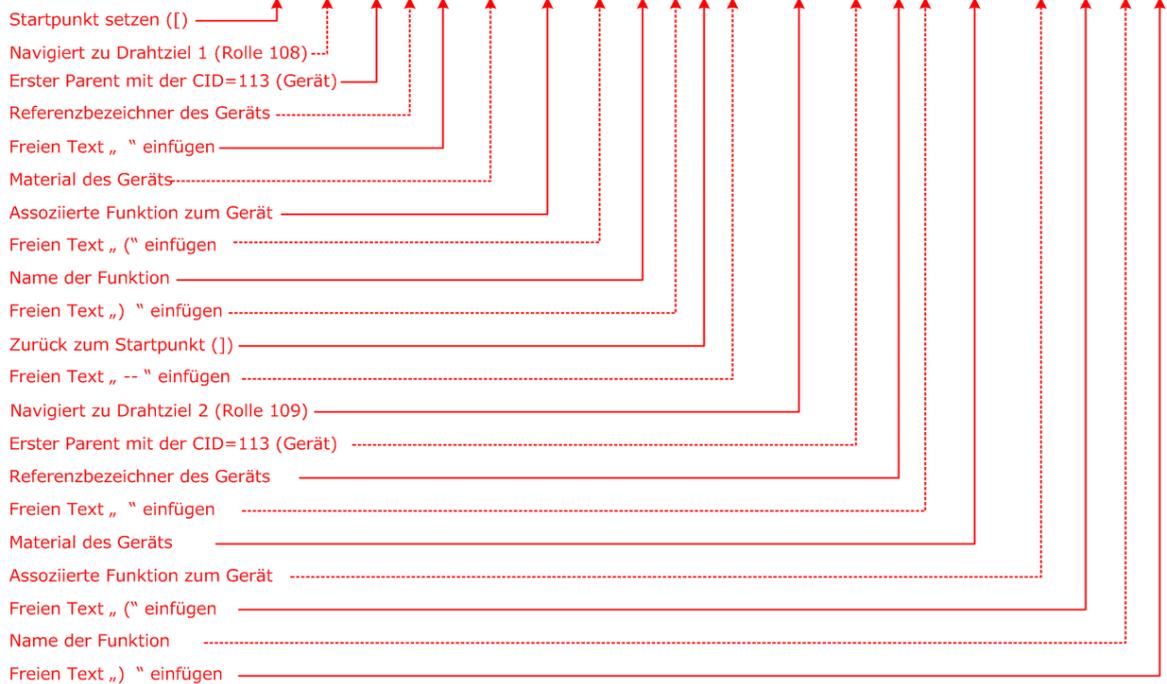
Beispiele für Formeln:	
Formel:	Wert des Formelattributes (Zeichenfolge):
R-3;	Navigation zu Kindobjekten (Rolle -3).
R108;	Navigiert vom Pin zum Draht (Rolle 108).
Rt108:904;	Navigiert vom Pin zum Draht (Rolle 108) beschränkt auf PE (TID = 904).
R100" / ";A5;	An einer Funktion: gibt die Namen aller mit der Funktion assoziierten Geräte (Rolle 100), getrennt durch „ / “ an.
R100:223" / ";Ar;	An einer Funktion: gibt die (langen/Full name) Namen aller mit der Funktion assoziierten Kabel (Rolle 100 = alle Geräte mit Einschränkung auf CID=223), getrennt durch „ / “ an.
R100:223" / ";RM1:2;A5;	An einer Funktion: gibt die Namen aller mit der Funktion assoziierten Kabel, getrennt durch „ / “ an. Ist mehr als 1 Kabel mit der Funktion assoziiert, sollen nur 2 Kabel angezeigt werden.
Rb100;A5;	An einem Gerät: gibt den Namen der assoziierten Funktion an.
Rb100;A5;"(;A25;"");	An einem Gerät: Funktionsname und Kommentar der zugeordneten Funktion.
R-3" / ";A5;	Listet alle aggregierten Objekte (Kindobjekte), getrennt mit „ / “ auf.  <pre> graph TD Aggregation --> A1 Aggregation --> A2 Aggregation --> A3 A1 --> 1.2 A1 --> 11.2 A1 --> 111.2 1.2 --> 11.2 11.2 --> 111.2 111.2 --> 1111.2 A2 --> 2.2 </pre> <p>Ergebnis der Formel: A1 / A2 / A3</p>

Aufgabe:

Darstellung der an einem Draht angeschlossenen Geräte mit deren Referenzbezeichner, Materialnummer und zugeordneter Funktion (in Klammern).



[Rb108;U113;Ar;" ";A320;Rb100;" (";A5;" ");] -- ";Rb109;U113;Ar;" ";A320;Rb100;" (";A5;" ");



Ergebnis der Formel:

++Baueinheit 1 -D1 Material A (.PD) -- ++Baueinheit 2 -M2 Material B (.S)

2.6.2 Relation Multi

Schlüsselkennung RM	Behandlung von Mehrfach-Zielen (R elation M ulti).
Syntax: RM [p s] <counter> [: <modcounter>] [" <text> "];	
p	Legt fest, dass der eingegebene Text <text> bei Erreichen des Grenzwerts (counter) P räfix werden soll.
s	Legt fest, dass der eingegebene Text <text> bei Erreichen des Grenzwerts (counter) S uffix werden soll.
counter	Definiert den Grenzwert, ab dem die Sonderbehandlung greifen soll. Erreicht die Anzahl der Objekte den Wert <counter>, so wird die Anzahl auf <modcounter> geändert. <modcounter=0> hat eine Sonderfunktion: <ul style="list-style-type: none"> • der Präfix/Suffix wird zum Ergebnisstring. • das ‚aktuelle‘ Objekt wird auf ‚nothing‘ gesetzt. • die Anzahl bisheriger Multi-Objekte reduziert sich auf 0. Damit wird die weitere Auswertung der folgenden Statements abgebrochen und erst bei einer folgenden Sprungmarke fortgesetzt
modcounter	Definiert die modifizierte Anzahl zu behandelnder Objekte. <modcounter> muss niedriger sein als <counter>.
text	Text, der zum Präfix oder Suffix werden soll.

<u>Beispiele:</u>	
...;RMpm"Text";...	Modifikation der Anzeige gefundener assoziierter Objekte. Ist die Anzahl größer oder gleich m, erhalten die erzeugten Zeilen am Anfang den Text angefügt (p= Präfix).
...;RMm:n;...	Einschränkung der Anzeige der gefundenen assoziierten Objekte. Ist die Anzahl größer oder gleich m, werden nur n Objekte angezeigt. Ist die Anzahl kleiner m, werden alle Objekte angezeigt.

2.6.3 Relation Objektgruppen

Sie können eine Objektgruppe verwenden, um eine Formelauswertung auf eine bestimmte Menge von Objekten zu beschränken. Die Festlegung der Objektmenge kann nicht durch einfache Navigation erreicht werden. Mittels einer Objektgruppe können Sie z. B. verhindern, dass ein Objekt mehrfach behandelt wird.

Es gibt nur eine Objektgruppe pro Formelattribut.

Beim Hinzufügen von Objekten zur Gruppe können Sie optional ein Attribut festlegen, dessen Wert auf Gleichheit geprüft wird, bevor das Objekt der Gruppe hinzugefügt wird. Das gleiche gilt für das Löschen von Objekten aus der Gruppe. Das erste Objekt, das mit einer Attributkennung in die Objektgruppe hinzugefügt wird, definiert die Bedingung. Eine Objektgruppe kann Elemente enthalten, die auf verschiedene Arten hinzugefügt wurden; d.h. mit oder ohne Prüfung einer Bedingung.

Schlüsselkennung GO	Behandlung von Objektgruppen (Grouped Objects)
Syntax: GO [+ [A <id>]] [- [A <id>]] t c ;	
GO+A_{id}	Das erste Objekt einer Gruppe definiert die Bedingung und wird selber in die Gruppe aufgenommen.
GO-A_{id}	Das erste Objekt einer Gruppe definiert die Bedingung und wird selber nicht in die Gruppe aufgenommen.
+	Objekt hinzufügen nach Prüfung der Bedingung. Das erste Objekt der Gruppe legt die Bedingung fest.
-	Objekt löschen.
t	Testen, ob das Objekt bereits in der Gruppe enthalten ist. Der Test gibt einen booleschen Wert (0 oder 1) zurück.
c	Gruppe löschen.

Aufgabe:

Es soll eine Liste erstellt werden, die alle Pins und zugeordnete Spezifikationsgeräte getrennt durch „/“ enthält (CID 118 = Pin, CID 113= Gerät, Rolle 119 = Spezifikation).

Verwendete Formel:

[+R-3:118;GO+;][+R-3:113;Rb119;GO+;]RO;Ar;" / ";

Navigationsklammer **[+**
 Erzwingt Weiterbearbeitung der Formel falls Objektnavigation innerhalb der Klammer fehlschlägt
 Wähle Kindobjekt mit Einschränkung auf CID = 118 (Pin)
 Kindobjekt zur Gruppe hinzufügen
 Schließen Navigationsklammer **]**
 Navigationsklammer **[+**
 Wähle Kindgerät
 (Einschränkung auf CID = 113)
 Wähle dem Kindgerät zugeordnete Spezifikationsgeräte (Rolle=119)
 Spezifikationsgerät zur Gruppe hinzufügen
 Schließen Navigationsklammer **]**
 Auflistung der Elemente der aktiven Objektgruppe
 Referenzbezeichner pro Element der Objektgruppe
 Freien Text „ / “ einfügen

Ergebnis der Formel:

+F1 MT1 2 / +F1 MT1 8 / +F1 MT1 4 / +F1 MT1 1 / +F1 MT1 3 / +F1 MT1 6 / +F1 MT1 7 / +F1 MT1 9 / +F1 MT1 5 / Specifications Sensoren T - Temperatur Temperaturwiderstand mit Transmitter / Specifications Sensoren L - Niveau E&H_FMB70 /

2.7 Einbindung von mathematischen Operationen

Bei der Verwendung von Attributwerten in mathematischen Operationen ist zu beachten, dass nicht verwendbare Attributwerte im Default mit dem Wert 1 in die Berechnung eingehen. Mit Parametern kann festgelegt werden, dass für verschiedene Fälle der Wert 0 zurückgegeben wird.

Mathematische Operationen können nur mit Attributen aus dem aktuellen Objekt vorgenommen werden. Um Daten von anderen Objekten zu verwenden, müssen diese in Registern zwischengespeichert (**r**) und abgerufen (**R**) werden. Mit Hilfe der Register können dann Werte aus anderen Objekten in derselben Formel in die aktuelle Berechnung einbezogen werden.

Schlüsselkennung =	mathematische Operationen / Operatoren
Syntax:	
	= [[r R] <RegSel>] (d f u) <expression> ["<formatstring> "];
r RegSel	Der errechnete Wert wird in einem math. Register (Zwischenspeicher) gespeichert. Register: Werte von 1 – 5 sind möglich.
R RegSel	Der Wert eines Registers (1 – 5) wird ausgelesen. Register: Werte von 1 – 5 sind möglich.
d	Festlegung der Ganzzahlberechnung (Double)
f	Festlegung der Fließkommaberechnung (Float)
u	Festlegung der Fließkommaberechnung inklusive zugehöriger Einheit (Unit). Es wird nur die Verwendung von mathematischen Operatoren und Quadratwurzel empfohlen. Hat nur ein Operand eine Einheit, wird diese verwendet. Haben beide Operanden eine Einheit, wird die zuerst genannte verwendet, die Vorgabe der Einheit erfolgt von links nach rechts.
expression	(+ - * / & ^ Q S s s xy exp) <argument> [<argument>]
	+, -, *, / Mathematische Operatoren
	&, , ^ Bitweise Operatoren
	Q Quadratwurzel
	S Sinus (Radiant)
	s Sinus (Grad)
	xy Potenz
	exp Exponential zur Basis e

	argument:	
	<attribute> <constant> <expression> <relation>	
attribute	A [n] [e] [f] <aid> [:<charpos>]	
	n	Fehlt das Attribut am Objekt, wird 0 statt 1 zurückgegeben.
	e	Ist das Attribut am Objekt leer, wird 0 statt 1 zurückgegeben.
	f	Ist kein auswertbarer numerischer Teil im Attribut am Objekt, wird 0 statt 1 zurückgegeben.
	charpos	Position in der Zeichenkette, ab der zu beginnen ist.
constant	K <value> pi e Value ist eine feste Zahlenfolge. Bei Fließkommazahlen muss ein Dezimalpunkt verwendet werden. Es sind keine Tausendertrenner erlaubt. e = eulersche Zahl	
expression	Siehe oben.	
relation	# [P] <role>; Gibt die Anzahl der Objekte zurück, die miteinander verknüpft sind (die über die Rolle gefunden werden). P : die Rolle wird ausgehend vom Parent berechnet. role : Relationsidentifikation. <u>Eine Liste der verwendbaren Rollen finden Sie unter References in der EB-WebHelp.</u> >0: Relationsrichtung vorwärts <0: Relationsrichtung rückwärts	
formatstring	Ausgabestring mit Formatierungsinformationen. Die möglichen Formatierungen sind im Anhang aufgeführt.	

Beispiele für Formeln mit mathematischen Operationen:

<u>Formel:</u>	<u>Wert des Formelattributes (Zeichenfolge):</u>
=#-3;	Gibt die Anzahl der Kindobjekte aus, die dem aktuellen Objekt zugeordnet sind
=#107;	Anzahl der Blätter die der Baueinheit zugeordnet sind (Rolle 107 = UnitToDiagram).
=#-107;	Anzahl der Baueinheit, die dem Blatt zugeordnet sind.
=f+A1234A5678;	Addiert die Werte der Attribute mit der ID 1234 und 5678. Darstellung als Fließkommazahl

=u+A1234A5678;	Addiert die Werte der Attribute mit der ID 1234 und 5678. Beispiel: A1234 = 3 cm A5678= 4 m Der Wert des Formelattributs ist 403 cm. A1234= 4,0 m A5678=3 cm Der Wert des Formelattributs ist 4,0 m.
=d*A1234A5678;	Multipliziert die Werte der Attribute mit der ID 1234 und 5678. Darstellung als ganze Zahl.
=f*A1234K1.2;	Multipliziert den Werte des Attributs mit der ID 1234 mit der Konstanten 1,2.
=fxyA1234K3;	Potenziert das Attribut (Basis) mit der Konstanten 3 (Exponent)
=fxyA1A2;	Potenziert das Attribut A1 (Basis) mit dem Attribut A2 (Exponent)
=fexpA1;	Bestimmt das Exponential des Attributs A1 (Exponent) zur Basis e
=d*A10061#P126;	Bestimmung der Menge eines Materials zu einem Hook Up. Es wird die Menge des Materials mit der Nutzung des Hook Up multipliziert.
=d*A1234#-3;	Multiplikation des Attributwerts mit der Anzahl der Kindobjekte.
=r3d*A1234#-3;	Multiplikation des Attributwerts mit der Anzahl der Kindobjekte. Das Ergebnis wird in Register 3 gespeichert.
=d*R3K2;	Den Inhalt aus Register 3 mit 2 multiplizieren.
=d+*A123K3A456;	Entspricht der mathematischen Formel $(A123*3)+A456$
=d*K2+A123A456;	Entspricht der mathematischen Formel $2*(A123+A456)$
=d*Ane123A456;	Entspricht der mathematischen Formel $A123*A456$ Bzw. $0*A456$, falls A123 nicht am Objekt oder leer ist.
=r1d*K2K2;=r2d+K3K3;=d+R1R2;	Das Ergebnis der Multiplikation $2*2 (=4)$ wird im Register 1 gespeichert. Das Ergebnis der Addition $3+3 (=6)$ wird im Register 2 gespeichert. Die Inhalte der Register 1 und 2 werden addiert. Ergebnis: 10

2.8 Objektprüfung

Schlüsselkennung VO	Objekt für eine Formel sicherstellen (V erify O bject) Bei negativem Prüfungsergebnis wird die Formel beendet.
Syntax: VO s S d D a A <cid>;	
s	Prüfung, ob das aktuelle Objekt des Formelattributs ein Symbol repräsentiert.
S	Prüfung, ob das aktuelle Objekt des Formelattributs kein Symbol repräsentiert.
d	Prüfung, ob das aktuelle Objekt des Formelattributs ein Device repräsentiert.
D	Prüfung, ob das aktuelle Objekt des Formelattributs kein Device repräsentiert.
a	Prüfung, ob das aktuelle Objekt des Formelattributs ein SAO repräsentiert (Symbol ohne AO).
A	Prüfung, ob das aktuelle Objekt des Formelattributs kein SAO repräsentiert (Symbol ohne AO).
cid	Prüfung, ob das aktuelle Objekt des Formelattributs der vorgegebenen Cover-ID entspricht.

2.9 Bedingungen

In Formelattributen können Bedingungen formuliert werden. Bedingungen werden durch geschweifte Klammern **{ }** gekennzeichnet. **Bedingungen werden in der Art einer case-Anweisung geschrieben.**

Es gibt grundsätzlich mehrere Alternativen je Bedingung. Eine Schachtelung der Bedingungen ist möglich.

Nach dem Vergleichswert einer Bedingung muss im Formelattribut ein Leerzeichen stehen!

Schlüsselkennung { }	Bedingung
Syntax: { [j] <Bedingungsformel> [{ [= < > <> : ("<Wertestring>" <aid>)]¹⁺ Leerzeichen <Ausführungsformel> }]¹⁺ }	
j	nach Abarbeitung der Bedingung springe (j ump) zum Objekt vor der Ausführung.
Bedingungsformel	Formelstring, dessen Ergebnis den Prüfstring ergibt.
=, <, >, <>	Prüfung auf Gleichheit, Ungleichheit, größer oder kleiner. Zahlen werden auch als Stringvergleich geprüft.
Wertestring	Die Zeichenfolge, die mit dem Prüfstring verglichen wird. Hier kann auch auf Registerinhalte zugegriffen werden.
aid	Alternativer Wert eines Attributs am Startobjekt, der mit dem Prüfstring verglichen wird.
:''''	Dieses Statement (ohne Bedingung) legt den Sonst-Fall der case-Anweisung fest. Hier können Sie festlegen was ausgeführt werden soll, wenn alle anderen Bedingungen nicht zutreffen.
Ausführungsformel	Anweisung, die ausgeführt wird, wenn die Prüfung erfolgreich war. Als Ausführungsformel ist jede Folge der möglichen Befehle erlaubt. Zum Beispiel: - Text in Hochkomma (" ") eingeschlossen - Attributwert - Formel -

Beispiele für Formeln mit Bedingungen:

Vergleich von 2 Attributwerten:

Das Attribut mit der ID=245 enthält die Breite und das mit der ID = 246 die Höhe eines Objekts.

- **{A245;{=246 "gleich";}{:'''' "ungleich";}};**
Ergebnis für Breite = 20,00 mm und Höhe = 20,00 mm „gleich“
Ergebnis für Breite = 20,00 mm und Höhe = 30,00 mm „ungleich“
- **{A245;{=246 "2 x ";A245;}{:'''' =*A245A246; " mm";}};**
Ergebnis für Breite = 20,00 mm und Höhe = 20,00 mm: „2 x 20,00 mm“
Ergebnis für Breite = 20,00 mm und Höhe = 30,00 mm: „600,00 mm“
- **{=f+A245A246;{="40" "vierzig";}{:'''' =f+A245A246; " mm";}};**
Ist die Summe von Höhe und Breite = 40, soll „vierzig“ ausgegeben werden, ansonsten soll die Summe von Breite und Höhe als Fließkomma-Zahl mit „mm“ dargestellt werden.

Prüfung ob ein Attributwert eine bestimmte Bedingung erfüllt:

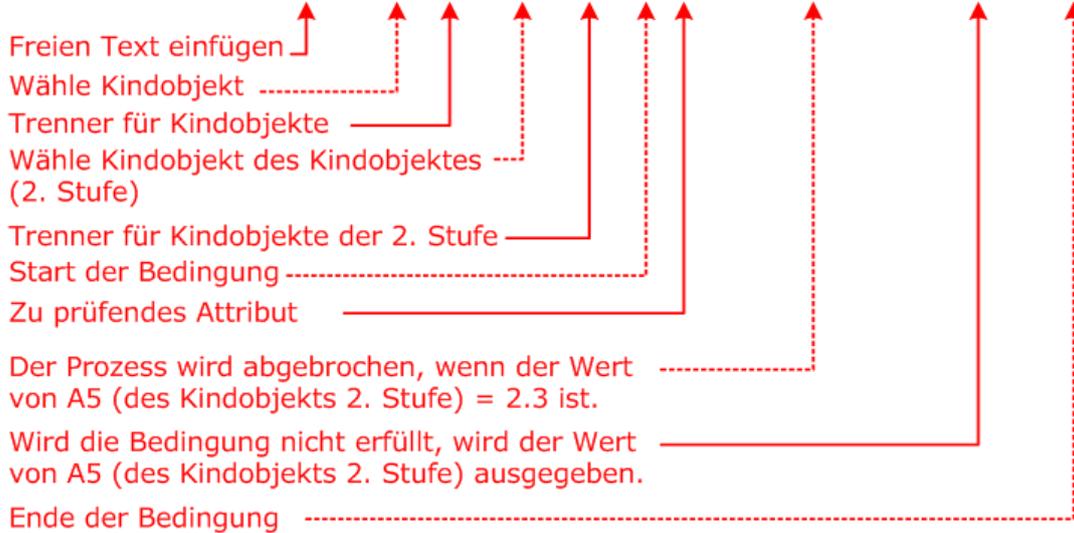
- {AF245(256);{="1" "manuell ";A245;}{:"" A245;}};**
 AF245(256) gibt den Wert „1“ aus, wenn das Attribut mit der ID = 245 manuell eingetragen wurde.
 Ergebnis bei manueller Eingabe des Attributs mit der ID = 245: „manuell 20,00 mm“.
- {A10025;{="1" A10001;}{:"" A10002;}};**
 Das Attribut mit der ID=10025 sei ein Boolesches Attribut.
 Hat dieses den Wert „1“ wird der Wert des Attributs mit der ID=10001 ausgegeben, ansonsten der Wert des Attributs mit der ID=10002.
- {A5;{="20"<"10">77 "im Bereich";}{:"" "nicht im Bereich";}};**
 Ist der Wert des Attributs mit der ID=5 gleich 20 oder kleiner 10 oder größer als der Wert des Attributs mit der ID=77 des Startobjekts, ist das Ergebnis der Formel „im Bereich“. Wird die Bedingung nicht erfüllt, wird „nicht im Bereich“ ausgegeben.
- {A5;{="1"<"2" "<2";}{="10">"24" ">24";}{:"" "default";}};**
 Ist der Wert des Attributs mit der ID =5 gleich 1 oder kleiner 2, wird „<2“ ausgegeben. Ist der Wert gleich 10 oder größer 24, wird „>24“ ausgegeben. Wird keine der Bedingungen erfüllt, wird „default“ ausgegeben.

Aufgabe:

Pro Kindobjekt der 1. Stufe (A1, A2) sollen die Kindobjekte der 2. Stufe (1.2, 1.3, 2.2) aufgelistet werden. Die Kindobjekte der ersten Stufe sollen mit „ / “ getrennt werden, die der 2. Stufe mit „#“. Sobald der Wert des Kindobjekts der 2. Stufe 2.3 erreicht, soll die Ausführung der Formel beendet werden.



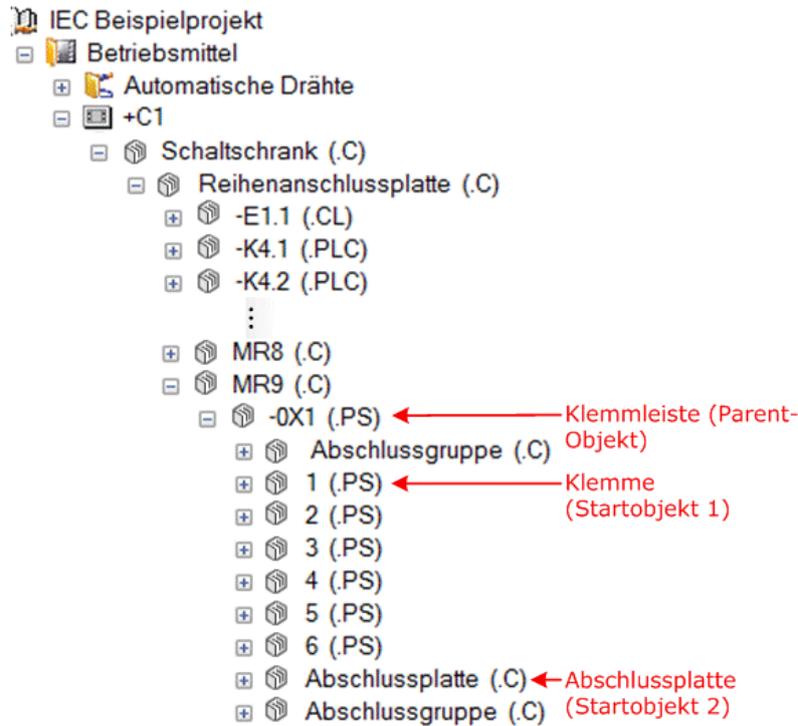
"Start ";R-3" / ";R-3" # ";{A5;{"="2.3" OGE;};{"'" A5;}}";



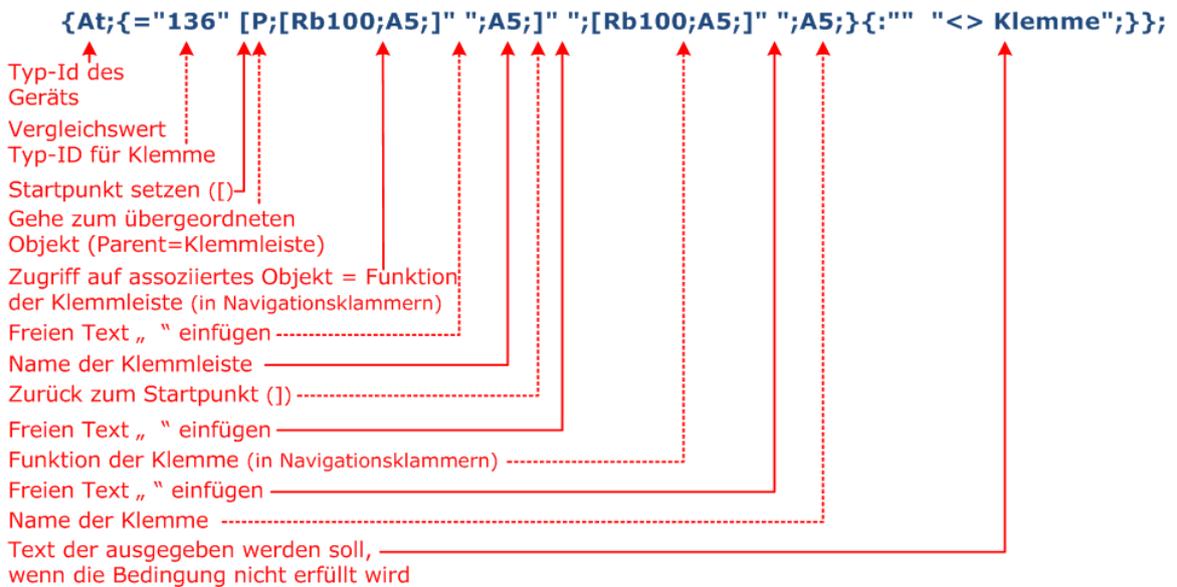
Ergebnis der Formel: START 1.2 # 1.3 / 2.2 #

Aufgabe:

Ist ein Betriebsmittel eine Klemme, soll die Funktion der Klemmleiste, die Klemmleiste, Funktion der Klemme und die Klemme ausgegeben werden. Ansonsten soll der Text „<> Klemme“ ausgegeben werden.



Verwendete Funktion:



Ergebnis der Funktion:

für die Klemme (Startobjekt1): .PS -0X1 .PS 1

für die Abschlussplatte (Startobjekt 2): <>Klemme

2.10 Anhang

2.10.1 Formatierung der Ausgabe

Die Format-Spezifikation besteht aus optionalen und erforderlichen Feldern.

Jedes einzelne Zeichen der Format-Spezifikation steht für eine bestimmte Format-Option. Die einfachste Form wäre das Prozentzeichen gefolgt von einem Zeichen (z.B. %s).

Steht nach dem Prozentzeichen ein Zeichen das keiner Format-Option entspricht, wird das Zeichen auf die Standardausgabe kopiert.

Formatierung der Ausgabe		
<u>Syntax:</u>		
%[flags] [width] [.precision] type		
flags	-	Ausrichtung links, Der Default-Wert ist rechts.
	+	Erzwingt das Vorzeichen (+ oder -). Ansonsten wird nur bei negativen Werten das Vorzeichen angezeigt.
	0	Auffüllen mit führenden Nullen bis zur minimalen Feldlänge. In Verbindung mit „-“ erfolgt kein Auffüllen des Feldes.
	„ “ (Leerzeichen)	Stellt positiven Ausgabewerten ein Leerzeichen voran. In Verbindung mit „+“ wird kein Leerzeichen vorangestellt.
	#	Zusammen mit dem Type = f, wird in jedem Fall ein Dezimalkomma gesetzt. Wird beim Typ = d ignoriert.
width	Legt die minimale Länge der Ausgabe in Zeichen fest. Ist die Ausgabe kürzer, so werden der Ausgabe, abhängig von der Ausrichtung, bis zur Minimallänge links oder rechts Leerzeichen zugefügt. Die Längenangabe verursacht niemals das Kürzen eines Wertes. Alle Zeichen werden ausgegeben.	
.precision	Optional. Maximale Anzahl von Zeichen oder die minimale Anzahl von Ziffern bei ganzzahligen Werten.	
type	d	Ganzzahlige Darstellung (Double) Der Wert gibt die minimale Anzahl von Ziffern an, die ausgegeben werden soll. Ist die Anzahl geringer, werden führende Nullen vorangestellt.
	f	Fließkommadarstellung (Float) Der Wert gibt die Anzahl der Nachkommastellen an. Wird ein Dezimalkomma ausgegeben, wird mindestens eine Ziffer davor angezeigt. Der Wert wird entsprechend der Nachkommastellen gerundet.