

VDA	<p style="text-align: center;">Arbeitskreis "CAD/CAM" im VDA-Rohstoff-Ausschuß (VDA-AK "CAD/CAM") CAD/CAM-Datenaustausch in der Betriebsstättenplanung</p> <p style="text-align: center;">VDABS</p>	4960
<p>Diese Empfehlung definiert eine einheitliche Schnittstelle für den Datenaustausch in der Betriebsstättenplanung, im Rahmen der Fabrikplanung.</p> <p>Die Abkürzung VDABS steht dabei für "Verband der Automobilindustrie - Betriebsstätten Schnittstelle".</p> <p>Diese Empfehlung beschränkt sich auf den Austausch von 2D-Geometriedaten, zusammen mit zeichnungstechnischen Elementen (Annotation) wie Bemaßung, Beschriftung und Schraffur.</p>		
<p>VDA-AK "CAD/CAM"</p> <p>AUDI, Behr, BMW, Bosch, Brose, Continental, Drauz, FLACHGLAS, Ford, FREUDENBERG, GETRAG, Happich, Hella, IVECO, Karmann, KEIPER-RECARO, Kostal, MAN, Mercedes-Benz, Opel, PIERBURG, Porsche, SIEMENS, SWF, Teves, ULO, VDO, VEGLA, Volkswagen, WABCO, Webasto, ZF.</p>		
Herausgeber:	<p>Verband der Automobilindustrie e.V. (VDA) Westendstraße 61 Postfach 170563 Kennwort : VDA-AK "CAD/CAM" 6000 Frankfurt Telefon (069) 75 70-238 Telex 411293</p>	<p>4960 Copyright Nachdruck, auch auszugsweise, nur mit Genehmigung des Herausgebers gestattet</p>

Inhalt

1	Anwendung und Zweck	1
2	Aufbau des Austauschformates	3
3	Aufbau der HEADER - Sektion	6
4	Aufbau der DATA - Sektion	7
4.1	LIBRARY - Bereich	8
4.2	GEOMETRY_2D - Bereich	15
4.3	AEC_ATTRIBUTES - Bereich	25
4.4	PRESENTATION_DRAFTING - Bereich	28
4.5	Gruppierungen	42
5	Beispiel	46
6	Literatur	51

1 Anwendung und Zweck

Der Anwendungsbereich dieser Empfehlung ist der Datenaustausch in der Betriebsstättenplanung Rahmen der Fabrikplanung.

Objekte der Betriebsstättenplanung sind:

- Werksflächen,
- Gebäude und bauliche Anlagen,
- Ausrüstungen und Maschinen,
- Ver- und Entsorgungsanlagen sowie
- Betriebs- und Büroeinrichtungen.

Ein im Jahr 1991 durchgeführter Vergleich von VDABS mit VDAIS bzw. IGES ergab wesentliche Vorteile für VDABS beim CAD-Datenaustausch in der Betriebsstättenplanung. VDABS erlaubt beispielsweise den Austausch von assoziativen Bemaßungen und von Sachdaten. Weiterhin ergeben sich deutlich kürzere Austauschfiles gegenüber VDAIS.

Die Vorzüge dieser Spezifikation in der vorliegenden Form liegen im wesentlichen in:

- Verfügbarkeit von Softwaretools und Übersetzern für marktgängige CAD-Systeme,
- Unterstützung durch die Spitzenverbände des Bauwesens und durch das Bundesbauministerium.

In der zweiten Ausbaustufe wird diese Schnittstelle um den Austausch von 3D-Bauteilen zusammen mit Sachdaten erweitert. Diese zweite Ausbaustufe wird voraussichtlich Ende 1993 definiert sein.

VDABS entspricht sachlich der von den Spitzenverbänden der Bauwirtschaft empfohlenen Richtlinie STEP-2DBS.

VDABS baut auf STEP-Konventionen des Jahres 1988 auf und verwendet das zu diesem Zeitpunkt gültige physikalische Fileformat und geometrische Elemente. Zu diesem Zeitpunkt nicht enthaltene Elemente wie Bemaßung, Beschriftung, Schraffuren und Strukturelemente wurden in Anlehnung an die STEP-Methode entwickelt.

Diese nationale Empfehlung wurde als deutscher Beitrag bei der ISO (International Organization for Standardization) eingereicht. Sie wurde und wird neben anderen Vorschlägen diskutiert. Rückwirkungen sowie eine Ausweitung der Themenstellungen auf die vorliegende Fassung sind daher möglich.

Für weitere Informationen steht Ihnen das FZI

Forschungszentrum Informatik
Forschungsbereich CAD/CAM-Technologie

Haid- und Neu-Straße 10-14

7500 Karlsruhe 1

Tel. 0721-9654-501

Fax 0721-9654-509

zur Verfügung.

2 Aufbau des Austauschformates

Leerzeichen

Leerzeichen, Tabulatoren und Zeilenumbrüche dienen (außer in Texten) der Lesbarkeit eines Austauschfiles. Ihre Benutzung wird empfohlen.

Kommentare

Kommentare sind in /* und */ einzuschließen. Kommentare dürfen nicht geschachtelt werden. Alle Zeichen und Zeichenfolgen außer */ sind erlaubt.

Entities

Entities ohne Attribute haben folgendes Format:

@ *Entity-Kennung* = *Entity-Schlüsselwort* ;

Entities mit Attributen haben folgendes Format:

@ *Entity-Kennung* = *Entity-Schlüsselwort* (*Attribute*) ;

Anmerkungen:

- Mit dem Zeichen '@' wird ein Entity begonnen.
- *Entity-Kennung* ist eine (höchstens 9-stellige) vorzeichenlose ganze Zahl, die der Identifizierung des Entities dient. Die Entities müssen aufsteigend beschrieben werden. Lücken in der Numerierung sind erlaubt.
- Das Zeichen '=' trennt die *Entity-Kennung* vom *Entity-Schlüsselwort*.
- *Entity-Schlüsselwort* ist eines der in den Abschnitten 3 und 4 beschriebenen Entity-Namen.
- Mit dem Zeichen '(' beginnt die Liste der Attribute.
- Die Art und Anzahl der Attribute hängen vom Entity ab und sind in den Abschnitten 3 und 4 beschrieben. Die einzelnen Attribute werden durch Kommata voneinander abgetrennt.
- Das Zeichen ')' beendet die Liste der Attribute.
- Das Zeichen ';' beendet ein Entity.

Referenzen

Innerhalb der Attributliste werden Referenzen auf andere Entities wie folgt notiert:

Entity-Kennung

Alle Referenzen müssen sich auf Entities beziehen, die bereits definiert wurden und somit eine numerisch kleinere Kennung als das referierende Entity haben.

Zahlen (Integer und Real)

Ganze Zahlen bestehen aus einem optionalen Vorzeichen und beliebig vielen Ziffern. Fließkommazahlen bestehen aus einer Dezimalzahl und einem optionalen Exponenten. Die Dezimalzahl muß mindestens eine Ziffer und einen Punkt enthalten. Der Exponent besteht aus 'e' oder 'E' und einer ganzen Zahl.

Texte (String)

werden in " eingeschlossen. ", !: und :! innerhalb von Texten müssen verdoppelt werden. Beliebige nichtdruckbare Zeichen werden wie folgt notiert:

!: *Dezimalcode* , *Dezimalcode* , ... :!

- !: beginnt eine Sequenz von Dezimalcodes
- *Dezimalcode* ist der dezimale ASCII-Code eines Zeichens. Mehrere Dezimalcodes werden durch Kommata voneinander getrennt.
- :! beendet eine Sequenz von Dezimalcodes

Aufbau

Ein Austauschfile ist wie folgt gegliedert:

```

VDABS_START;           1
HEADER;                2
  header entities     3
ENDSEC;
DATA;                  4
  data entities       5
ENDSEC;
VDABS_END;
```

Anmerkungen zu den Zeilen:

- 1 Jedes Austauschfile muß mit `VDABS_START;` beginnen.
- 2 Die Header-Sektion wird mit `HEADER;` begonnen und `ENDSEC;` beendet.
- 3 *header entities* werden in Abschnitt 3 beschrieben.
- 4 Die Data-Sektion wird mit `DATA;` begonnen und `ENDSEC;` beendet.
- 5 *data entities* werden in Abschnitt 4 beschrieben.
- 6 `VDABS_END;` beendet einen Austauschfile. Lediglich Kommentare dürfen noch folgen.

Bei STEP/PDES wird klar zwischen Geometrie-Entities und Entities für Annotation, also für Bemaßung, Beschriftung und Schraffur unterschieden. Diese Unterscheidung wird bei VDABS übernommen. Es gibt also z. B. das Entity `CIRCLE_2D` als Geometrie-Entity und das Entity `ANN_CIRCLE_2D` als Annotation-Entity. Die Attribute beider Entities sind identisch.

3 Aufbau der HEADER - Sektion

Die Entities der Header-Sektion werden derzeit noch nicht ausgewertet. Es ist vorgesehen, sie entsprechend ISO-10303-21 zu definieren, sowie diese internationale Norm verabschiedet wird.

4 Aufbau der DATA - Sektion

Die DATA-Sektion eines Austausch-Files ist in vier Bereiche gegliedert:

- **LIBRARY**

Dieser Bereich wird zur Beschreibung von Bibliothekselementen verwendet. Er umfaßt nicht nur Symbole, sondern auch Textfonts, Strichstärken, Farben und sogenannte **USER_DEFINED_ENTITIES** für bauspezifische technische Attribute.

- **GEOMETRY_2D**

Hier wird die zur Beschreibung des Bauwerks verwendete Geometrie übergeben, in tatsächlichen Abmessungen, also im Maßstab 1:1. Nicht zum Bauwerk gehörige Linien, z. B. Rasterlinien, befinden sich nicht in diesem Bereich, dafür jedoch Symbole z. B. für den Ausbau. Die Attribute für die Linien sind nicht in diesem Bereich.

- **AEC_ATTRIBUTES**

Dieser Bereich enthält die Definition von nichtgrafischen Attributen, wie Material, Kostenelement, Wichte etc. und deren Zuordnung zu den geometrischen Entities.

- **PRESENTATION_DRAFTING**

In diesem Bereich werden den **GEOMETRY_2D**-Entities grafische Attribute zugewiesen und die Zeichnung um Bemaßung, Beschriftung und Schraffur ergänzt.

4.1 LIBRARY - Bereich

4.1.1 Entity LIBRARY

Mit diesem Entity wird der Beginn des LIBRARY-Bereichs angezeigt. Anfang und Ende des Bereiches werden mit SCOPE und ENDScope markiert. SCOPE und ENDScope sind in Abschnitt 4.5.1 beschrieben.

Attribute: keine

Beispiel:

```
@10 = LIBRARY SCOPE ;
```

4.1.2 Entity UNITS

Mit diesem Entity werden die im Austauschfile verwendeten Dimensionen übergeben.

Attribute:	String	Längeneinheit zulässige Werte sind "MM", "CM", "M" und "INCH"
	String	Winkleinheit zulässige Werte sind "BOGENMASS" und "ALTGRAD"

Beispiel:

```
@20 = UNITS ( "INCH" , "BOGENMASS" ) ;
```

4.1.3 Entity CONVERSION

Mit diesem Entity werden die im Austauschformat übergebenen Einheiten in die Standardeinheiten mm und Bogenmaß umgerechnet.

Typischer Anwendungsfall ist die Umrechnung von z. B. Inch in mm mit dem Faktor 2.54. Die Attribute sind in derselben Reihenfolge angeordnet wie beim Entity UNITS.

Attribute:	Real	Faktor für die Umrechnung der im Austauschfile verwendeten Längeneinheiten in mm
	Real	Faktor für die Umrechnung der im Austauschfile verwendeten Winkleinheiten in Bogenmaß

Anmerkung: UNITS oder CONVERSION darf nur einmal im Austauschfile vorkommen.

Beispiel:

```
@21 = CONVERSION ( 2.54 , 1 ) ;
```

4.1.4 Entity LINE_WIDTH

Mit diesem Entity werden Linienbreiten definiert. Mit der übergebenen Bezeichnung können die Linienbreiten auf die internen Nummern des empfangenden CAD-Systems eingestellt werden.

Attribute:	Integer	Interne Nummer der Linienbreite
	String	Name der Linienbreite
	Real	(optional) Wert der Linienbreite

Anmerkung: Derzeit ist das Format für externe Libraries noch nicht definiert. Der Wert der Linienbreite muß also, obwohl er formal optional ist, noch angegeben werden.

Anmerkung: Die Beispiele enthalten die aus dem "Leitfaden für den erfolgreichen Datenaustausch" empfohlenen Attribute.

Beispiel:

```
@23 = LINE_WIDTH ( 1 , "DIN 1356 25" , 0.25 ) ;
@24 = LINE_WIDTH ( 2 , "DIN 1356 35" , 0.35 ) ;
@25 = LINE_WIDTH ( 3 , "DIN 1356 50" , 0.50 ) ;
@26 = LINE_WIDTH ( 4 , "DIN 1356 70" , 0.70 ) ;
@27 = LINE_WIDTH ( 5 , "DIN 1356 100" , 1.0 ) ;
@28 = LINE_WIDTH ( 6 , "DIN 1356 140" , 1.4 ) ;
```

4.1.5 Entity LINE_STYLE

Mit diesem Entity werden Linienarten als Strichfolgen definiert.

Attribute:	Integer	Interne Nummer des Linientyps Es gelten folgende Vorbelegungen: 1 Volllinie 2 unsichtbare Linie
	String	Name der Linienart
	Liste von	
	Real	Länge des Striches
	Real	Länge der Lücke
	usw.	
	Real	(optional) Reziproker Zielmaßstab, für den diese Linienart gilt. Falls keine Angabe gemacht wird, gilt der Strichtyp für alle Maßstäbe.

Anmerkung: Die Beispiele enthalten die aus dem "Leitfaden für den erfolgreichen Datenaustausch" empfohlenen Attribute.

Beispiel:

```
@31=LINE_STYLE(3,"STRICH-KURZ" , (2,1) );
@32=LINE_STYLE(4,"STRICH" , (4,1.5) );
@33=LINE_STYLE(5,"STRICH-LANG" , (6,2) );
@34=LINE_STYLE(6,"STRICH PUNKT" , (7,1,1,1));
@35=LINE_STYLE(7,"STRICH PUNKT PUNKT", (7,1,1,1,1,1));
@36=LINE_STYLE(8,"STRICH PUNKT-LANG", (15,2,2,2));
@37=LINE_STYLE(9,"STRICH PUNKT PUNKT-LANG",
(15,2,2,2,2,2));
@38=LINE_STYLE(10,"PUNKT" , (0.5,1.5));
```

4.1.6 Entity COLOR

Mit diesem Entity wird die Zuordnung der Farben zu den internen Nummern im Austauschfile vorgenommen. Mit dieser Zuordnung kann das empfangende System eine Farbe seiner internen Nummer zuweisen.

Attribute: Integer Interne Nummer der Farbe
 String Name der Farbe

Anmerkung: Die Beispiele enthalten die aus dem "Leitfaden für den erfolgreichen Datenaustausch" empfohlenen Attribute.

Beispiel:

```
@40 = COLOR ( 1, "SCHWARZ" ) ;
@41 = COLOR ( 2, "ROT" ) ;
@42 = COLOR ( 3, "GRÜN" ) ;
@43 = COLOR ( 4, "BLAU" ) ;
@44 = COLOR ( 5, "GELB" ) ;
@45 = COLOR ( 6, "MAGENTA" ) ;
@46 = COLOR ( 7, "CYAN" ) ;
@47 = COLOR ( 8, "WEISS" ) ;
```

4.1.7 Entity TEXT_FONT

Mit diesem Entity werden internen Fontnummern die entsprechenden Bezeichnungen zugeordnet.

Attribute: Integer Interne Nummer des Textfonts
 String Name des Textfonts

Anmerkung: Die Beispiele enthalten die aus dem "Leitfaden für den erfolgreichen Datenaustausch" empfohlenen Attribute.

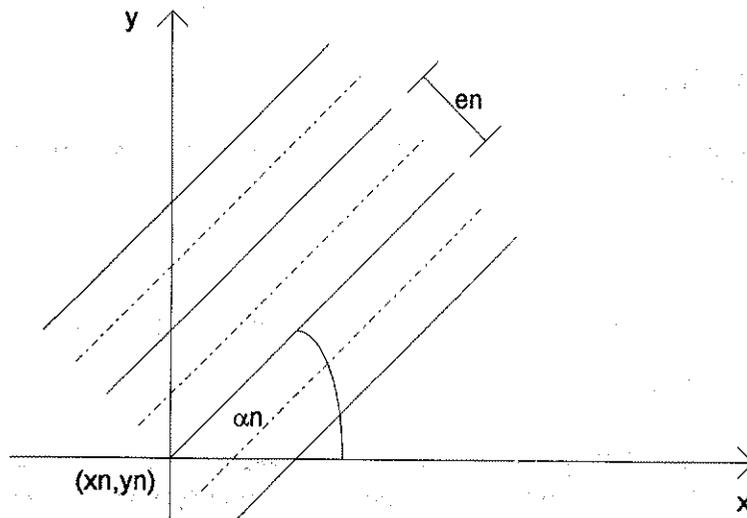
Beispiel:

```
@50 = TEXT_FONT ( 1, "DIN 6776" ) ;
@51 = TEXT_FONT ( 2, "DIN 6776 KURSIV" ) ;
```

4.1.8 Entity HATCH

Mit diesem Entity werden internen Schraffurnummern Schraffurbezeichnungen zugeordnet. Damit wird eine Zuordnung der im Austauschformat verwendeten internen Schraffurnummern zu den Schraffurkatalogen des empfangenden Systems möglich. Optional wird zusätzlich die Schraffur selbst definiert.

Attribute:	Integer	Interne Schraffurnummer
	String	Name der Schraffur
	Liste von	(optional)
	Integer-	Interne Nummer der Strichart
	Real	x_n -Koordinate des Ursprungs
	Real	y_n -Koordinate des Ursprungs
	Real	Winkel α_n der Schraffurlinie
	Real	Abstand e_n zur nächsten Schraffurlinie
	usw.	
	Real	(optional) Reziproker Zielmaßstab, für den die Angaben der Liste gelten. Falls keine Angaben gemacht werden, gelten die Angaben unverändert für alle Maßstäbe.



Beispiel :

```
@60 = HATCH ( 1 , "BETON" , ( 1 , 0 , 0 , 45 , 10 ,
( 6 , 0.705 , 0 , 45 , 10 ) ) );
```

4.1.9 Entity SYMBOL

Mit diesem Entity werden interne Nummern von Bibliothekssymbolen z. B. für Möblierung oder Haustechnik ihren Bezeichnungen zugeordnet. Das empfangende System wird damit in die Lage versetzt, die Zuordnung zu seinen internen Symboltabellen herzustellen. Optional werden die Symbole zusätzlich grafisch übertragen.

Attribute:

Integer	Interne Nummer des Symbols
String	Name des Symbols
Liste von	(optional) Referenz auf Entities COMPOSITE_CURVE_2D, aus denen das Symbol besteht. Die Angaben sind im Maßstab 1:1. usw.

Beispiel:

```
@60 = COMPOSITE_CURVE_2D ( ... ) ;
@61 = COMPOSITE_CURVE_2D ( ... ) ;
@62 = SYMBOL ( 1 , "WASCHBECKEN" , ( #60 , #61 ) ) ;
```

4.1.10 Entity TERMINATOR_SYMBOL

Mit diesem Entity werden Maßbegrenzungssymbole definiert als Verweis auf ein Entity ANN_COMPOSITE_CURVE_2D.

Attribute:

Integer	Interne Nummer des Maßbegrenzungssymbols
String	Name des Maßbegrenzungssymbols
Referenz	auf ein Entity vom Typ ANN_COMPOSITE_CURVE_2D

Beispiel:

```
@70 = ANN_COMPOSITE_CURVE_2D ( ... ) ;
@71 = TERMINATOR_SYMBOL (1, "BEGRENZUNGSPFEIL" , #70 ) ;
```

4.1.11 Entity USER_DEFINED_AEC_ENT

Mit diesem Entity können spezielle bauspezifische Entities für technische Attribute wie Material, Kostenelemente etc. definiert werden. Dieses Entity ist ein Provisorium, da allgemein anerkannte Entities nicht verfügbar sind. Die einzelnen Systemanbieter gehen bei den technischen Attributen auch unterschiedliche Wege. Nützlich wären für diesen Zweck Data-Dictionary-Entities. Da jedoch die wenigsten CAD-Systeme über die Kopplung mit Datenbanken verfügen, ist dieser Weg in der ersten Ausbaustufe der geplanten Schnittstelle nicht vorgesehen.

Attribute: String Name des zu definierenden Entities

Liste von

String	Name des Attributes
String	Typ des Attributes; zulässig sind "INTEGER", "REAL", "STRING" und "POINTER"

USW.

Beispiel:

```
@80 = USER_DEFINED_AEC_ENT ( "MATERIAL" ,
    ( "NAME" , "STRING" ,
      "E-MODUL" , "REAL" ,
      "WICHTE" , "REAL" ) ) ;
```

Der so erzeugte ENTITY Typ hat den Namen MATERIAL mit den Attributen NAMEN als String und E-MODUL und WICHTE jeweils als Real. Wird dieses Entity später verwendet, so wird dem Entityschlüsselwort ein Ausrufezeichen vorangestellt:

```
@81 = !MATERIAL ( "BETON" , 1 , 5 ) ;
```

4.2 GEOMETRY_2D - Bereich

In diesem Bereich werden die Geometrie-Entities beschrieben. Anfang und Ende des Geometrie-Bereiches werden mit SCOPE, ENDScope markiert. SCOPE und ENDScope sind in Abschnitt 4.5.1 beschrieben.

4.2.1 Entity GEOMETRY_2D

Mit diesem Entity wird der Beginn des Geometrie-Bereichs angezeigt.

Attribute: keine

Beispiel:

```
@200 = GEOMETRY_2D SCOPE;
```

4.2.2 Entity POINT_2D

Mit diesem Entity werden Punktkoordinaten beschrieben.

Attribute: Real X-Koordinate

Real Y-Koordinate

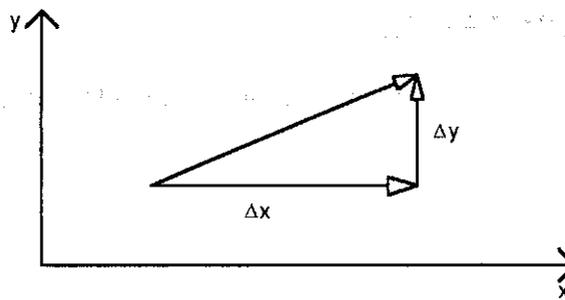
Beispiel:

```
@201 = POINT_2D ( 1.2 , 3.2 ) ;
```

4.2.3 Entity DIRECTION_2D

Mit diesem Entity wird eine Richtung beschrieben.

Attribute: Real Δx
 Real Δy

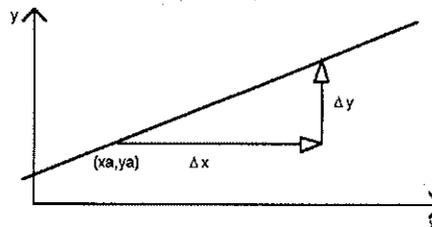


Beispiel:

```
@202 = DIRECTION_2D ( 1 , 0.3 ) ;
```

4.2.4 Entity LINE_2D

Mit diesem Entity wird eine Gerade in Parameterdarstellung beschrieben.



Parameterdarstellung der Geraden:

$$x = x_a + t \Delta x$$

$$y = y_a + t \Delta y$$

$$[-\infty < t < +\infty]$$

Attribute:	Real	x_a Koordinate eines Linienpunktes
	Real	y_a Koordinate des Linienpunktes oder Referenz auf POINT_2D Entity
	Real	Δx Richtung der Linie
	Real	Δy Richtung der Linie oder Referenz auf DIRECTION_2D Entity

Beispiel:

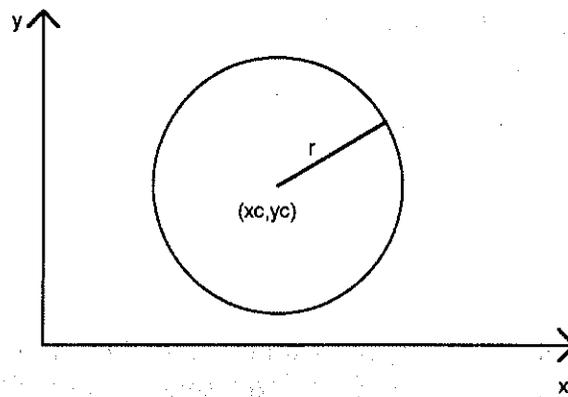
```
@ 210 = LINE_2D ( 1.2 , 3.2 , 1 , 0.3 ) ;
```

```
@ 211 = LINE_2D ( #201 , #202 ) ;
```

4.2.5 Entity CIRCLE_2D

Mit diesem Entity wird ein Vollkreis in Parameterdarstellung beschrieben. Die Koordinaten des Kreismittelpunktes können direkt oder als Verweis auf POINT_2D-Entities angegeben werden.

Das Entity CIRCLE-2D wird in zweierlei Weise gebraucht: einmal, um einen Vollkreis zu zeichnen und zum anderen, um mit Hilfe von TRIMMED_CURVE_2D (s. Kap. 4.2.7) einen Kreisbogen zu zeichnen. Im ersten Fall muß der Kreis innerhalb eines Layers (s. Kap. 4.5.2) stehen; im zweiten Fall, in dem der Kreis lediglich als Hilfsgeometrie dient, muß er sich außerhalb eines Layers befinden; die ihn referierende TRIMMED_CURVE_2D selbst steht immer innerhalb eines Layers.



Parameterdarstellung des Kreises:

$$x = x_c + r \cos(t)$$

$$y = y_c + r \sin(t)$$

$$[0 \leq t \leq 2 \pi]$$

Attribute:	Real	Radius r
	Real	x_c Koordinate des Kreismittelpunktes
	Real	y_c Koordinate des Kreismittelpunktes oder Referenz auf ein POINT_2D Entity

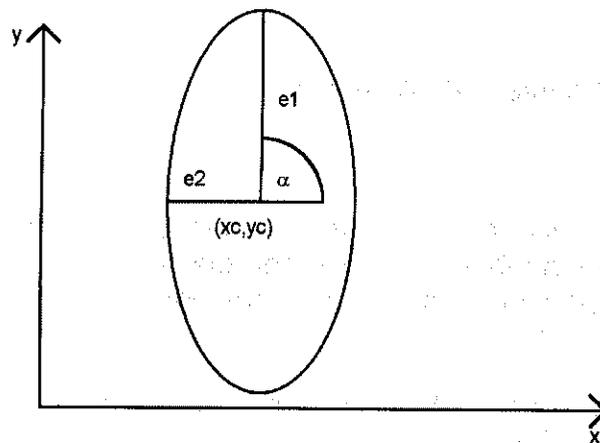
Beispiel:

```
@220 = CIRCLE_2D ( 5.0 , #201 ) ;
```

4.2.6 Entity ELLIPSE_2D

Mit diesem Entity wird eine volle Ellipse in Parameterdarstellung beschrieben. Die Koordinaten des Mittelpunktes können direkt oder als Verweis auf ein POINT_2D Entity beschrieben werden.

Das Entity ELLIPSE-2D wird in zweierlei Weise gebraucht: einmal, um eine volle Ellipse zu zeichnen und zum anderen, um mit Hilfe von TRIMMED_CURVE_2D (s. Kap. 4.2.7) einen Ellipsenbogen zu zeichnen. Im ersten Fall muß die Ellipse innerhalb eines Layers (s. Kap. 4.5.2) stehen; im zweiten Fall, in dem die Ellipse lediglich als Hilfsgeometrie dient, muß sie sich außerhalb eines Layers befinden; die sie referierende TRIMMED_CURVE_2D selbst steht immer innerhalb eines Layers.



Parameterdarstellung der Ellipse:

$$x = x_c + e_1 \cos(\alpha) \sin(t) - e_1 \sin(\alpha) \cos(t)$$

$$y = y_c + e_2 \sin(\alpha) \cos(t) + e_2 \cos(\alpha) \sin(t)$$

$$[0 \leq t \leq 2 \pi]$$

Attribute: -	Real	Große Halbachse e1
	Real	Kleine Halbachse e2
	Real	Drehwinkel α der großen Halbachse gegen die x-Achse in Bogenmaß
	Real	x_c Koordinate des Mittelpunktes
	Real	x_c Koordinate des Mittelpunktes oder Referenz auf ein POINT_2D Entity

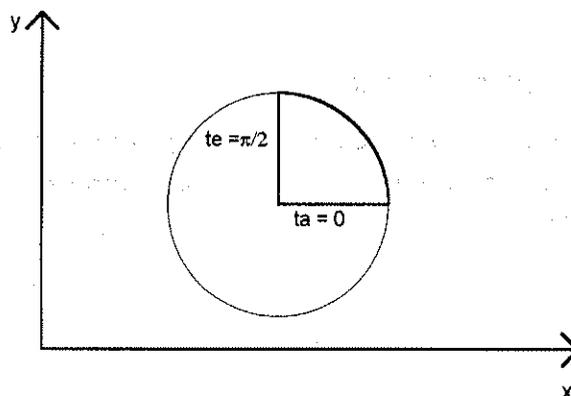
Beispiel:

```
@230 = ELLIPSE_2D ( 10., 5. , 1.571 , 0 , 0 ) ;
```

4.2.7...Entity TRIMMED_CURVE_2D

Mit diesem Entity wird eine Strecke oder das Segment eines Kreises oder einer Ellipse beschrieben. Dazu werden die Anfangs- und Endparameter t_a und t_e angegeben, sowie der Verweis auf die entsprechende Basiskurve.

Das Entity TRIMMED-CURVE_2D muß immer innerhalb eines Layers stehen. (s. Kap. 2.5.2)



Attribute:	Referenz	auf die Basiskurve nur LINE_2D, CIRCLE_2D oder ELLIPSE_2D sind erlaubt
	Real	Anfangsparameter t_a
	Real	Endparameter t_e

Beispiel:

```
@240 = TRIMMED_CURVE_2D ( #220 , 0 , 1.571 ) ;
```

4.2.8 Entity POLYGON_2D

Mit diesem Entity werden Polygonzüge als Folge von Punkten beschrieben. Die Polygonzüge können offen sein. Die Punkte werden entweder als Koordinatenpaare x,y eingegeben oder als Verweise auf Entities POINT_2D. Der Vorteil gegenüber der Beschreibung mit dem Entity COMPOSITE_CURVE_2D (s. Kap. 4.2.9) besteht darin, daß jeder Punkt bei der Verwendung von POLYGON_2D nur einmal angegeben wird.

Das Entity POLYGON_2D muß immer innerhalb eines Layers stehen. (s. Kap. 2.5.2)

Attribute:	Liste von	
	Real	x Koordinate
	Real	y Koordinate oder Referenz auf ein Entity POINT_2D
		usw.

Beispiel:

```
@250 = POLYGON_2D ((0,0, 100,0, 100,100, 0,100, 0,0));
```

4.2.9 Entity COMPOSITE_CURVE_2D

Mit diesem Entity werden Linienzüge als Folge von Verweisen auf Entities vom Typ CIRCLE_2D, ELLIPSE_2D, TRIMMED_CURVE_2D oder POLYGON_2D beschrieben. Die erzeugte Kurve muß weder zusammenhängend noch geschlossen sein.

Attribute: Liste von Referenz auf Teilkurven lediglich CIRCLE-2D, ELLIPSE-2D; TRIMMED_CURVE_2D und POLYGON_2D sind erlaubt.
usw.

Beispiel:

```
@260 = COMPOSITE_CURVE_2D ( ( #240 , #250 ) ) ;
```

4.2.10 Entity AREA_2D

Mit diesem Entity werden Flächen als Folgen geschlossener Linienzüge definiert. Diese Flächen können schraffiert werden. Der Inhalt dieser Flächen kann berechnet werden und den Flächen können Attribute wie Nutzfläche, Verkehrsfläche, Abteilungszugehörigkeit etc. zugeordnet werden. Diese Flächendefinition hat keine topologische Bedeutung und ist deswegen nicht zu verwechseln mit dem FACE-Entity in STEP.

Attribute: Liste von Referenz auf COMPOSITE_CURVE_2D Entity usw.

Die Linienzüge jeder COMPOSITE_CURVE_2D müssen geschlossen sein. Eine bestimmte Reihenfolge ist jedoch nicht vorgeschrieben.

Der erste Verweis bezieht sich auf die äußere Begrenzungslinie, die folgenden Veweise beziehen sich auf innerliegende Begrenzungslinien.

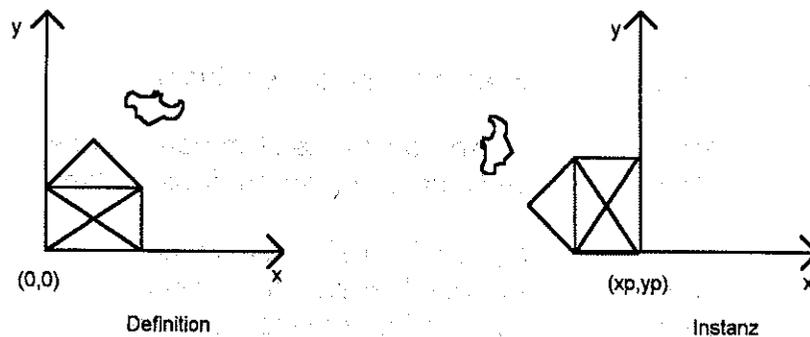
Die Kurven sollen einander nicht überschneiden.

Beispiel:

```
@260 = CIRCLE_2D ( 100 , 0 , 0 ) ;
@261 = COMPOSITE-CURVE-2D ( ( #260 ) ) ;
@261 = AREA_2D ( ( #261 ) ) ;
```

4.2.11 Entity INSTANCE_2D

Mit diesem Entity werden Gruppen (s. Kap. 4.5.3) instanziiert und COMPOSITE_CURVE_2D kopiert. Dabei bestimmt eine Matrix, inwieweit dabei rotiert und skaliert wird.



Transformation:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} a_{xx} & a_{xy} \\ a_{yx} & a_{yy} \end{pmatrix} * \begin{pmatrix} x \\ y \end{pmatrix}$$

Attribute:	Referenz	auf ein GROUP_2D oder COMPOSITE_CURVE_2D Entity
	Real	x _p Koordinate des Einfügepunktes
	Real	y _p Koordinate des Einfügepunktes
	Real	a _{xx} der Transformationsmatrix
	Real	a _{xy} der Transformationsmatrix
	Real	a _{yx} der Transformationsmatrix
	Real	a _{yy} der Transformationsmatrix

Beispiel:

```
@270 = GROUP_2D ( ( ... ) ) ; /* S. KAP 4.5.3 */
@271 = INSTANCE_2D ( #270 , 0,0 , 0,-1,1,0 ) ;
/* DREHUNG UM 90 GRAD */
```

4.2.12 Entity INSTANCE_SYMBOL

Mit diesem Entity werden Symbole instanziiert. Der Transformationsmechanismus ist bei INSTANCE_2D (Kap. 4.2.11) beschrieben.

Attribute:	Integer	Interne Nummer des Symbols
	Real	x _p Koordinate des Einfügepunktes
	Real	y _p Koordinate des Einfügepunktes
	Real	a _{xx} der Transformationsmatrix
	Real	a _{xy} der Transformationsmatrix
	Real	a _{yx} der Transformationsmatrix
	Real	a _{yy} der Transformationsmatrix

Beispiel:

```
@272 = INSTANCE_SYMBOL ( 1 , 0,0 , 2,0,0,2 ) ;
/* SKALIERUNG UM FAKTOR 2 */
```

4.3 AEC_ATTRIBUTES - Bereich

In diesem Bereich werden die technischen Attribute zu den grafischen Elementen übergeben, d. h. hier wird festgelegt, daß ein Rechteck eine Wand ist, mit dem Material Beton, mit Kosten etc.

Anfang und Ende des Bereiches werden mit SCOPE und ENDScope markiert. SCOPE und ENDScope sind in Abschnitt 4.5.1 beschrieben.

Da es für diesen Bereich keinen allgemein akzeptierten Standard gibt, werden vorläufig nur die Mechanismen angegeben, wie nichtgrafische Attribute an geometrische Entities angehängt werden können. Diese Lösung ist eine Übergangslösung, die dann funktioniert, wenn sich "Sender" und "Empfänger" über Inhalt und Aufbau der Entities verständigen.

4.3.1 Entity AEC_ATTRIBUTES

Mit diesem Entity wird der Beginn des Bereiches der technischen Attribute und deren Zuweisung zu den Geometriedaten angezeigt.

Attribute: keine

Beispiel:

```
@300 = AEC_ATTRIBUTES SCOPE ;
```

4.3.2 User Defined AEC Entities

Im LIBRARY-Bereich wurden unter Kap. 4.1.11 Entity Typen definiert, die hier verwendet werden dürfen. Dem Entity-Schlüsselwort wird dabei ein Ausrufezeichen vorangestellt.

Attribute: Sie ergeben sich aus der Entity-Definition entsprechend 4.1.11

Beispiel:

Im Library-Bereich wurde folgendes Entity definiert:

```
@80 = USER_DEFINED_AEC_ENT ( "MATERIAL" ,  
    ( "NAME" , "STRING" ,  
      "E-MODUL" , "REAL" ,  
      "WICHTE" , "REAL" ) ) ;
```

Nun kann es mit Attributen verwendet werden:

```
@301 = !MATERIAL ( "BETON" , 1 , 5 ) ;
```

4.3.3 Entity ATTACH_AEC_ATTRIBUTES

Mit diesem Entity wurden einem geometrischen Entities ein oder mehrere USER_DEFINED_AEC Entities angefügt. Dieser Mechanismus entspricht dem ursprünglich in STEP vorgesehenen LINK-Mechanismus. Da er jedoch strittig ist, wurde dieses Entity definiert.

Attribute: Referenz auf ein Entity ; dabei sind nur die Entities des GEOMETRY Bereichs erlaubt.

Liste von

Referenz auf ein mit USER_DEFINED_AEC_ENT erzeugtes Entity.

USW.

Beispiel:

```
@302 = ATTACH_AEC_ATTRIBUTES ( #261, ( #30 ) ) ;  
/* DIE FLÄCHE 261 ERHÄLT DAS MATERIAL BETON*/
```

4.4 PRESENTATION_DRAFTING - Bereich

In diesem Bereich werden den im Geometrieteil erzeugten Linien Linienattribute zugewiesen, sowie die zeichnungstechnischen Elemente Bemaßung, Beschriftung, und Schraffur hinzugefügt.

Anfang und Ende des Bereiches werden mit SCOPE und ENDScope markiert. SCOPE und ENDScope sind in Abschnitt 4.5.1 beschrieben.

In diesem Bereich dürfen nicht dieselben Entities wie im Geometriebereich verwendet werden. Linien als Begrenzungslinien von Bauteilen und Bemaßungslinien dürfen also nicht mit denselben Entitytypen beschrieben werden. Die entsprechenden Annotation-Entities sind in Kap. 4.4.2 beschrieben.

4.4.1 Entity PRESENTATION_DRAFTING

Mit diesem Entity wird der Beginn des Bereichs PRESENTATION_DRAFTING angezeigt.

Attribute: keine

Beispiel:

```
@400 = PRESENTATION_DRAFTING SCOPE ;
```

4.4.2 ANN_Entities

Zur Darstellung von Bemaßungslinien, Schraffuren, Rasterlinien und sonstigen zeichnungstechnischen Elementen, also von Linien, die keine projizierte Gebäudegeometrie darstellen, werden spezielle Entities verwendet. Sie beginnen alle mit ANN_. Es gibt insgesamt folgende Entitytypen:

- ANN_POINT_2D
- ANN_DIRECTION_2D
- ANN_LINE_2D
- ANN_CIRCLE_2D
- ANN_TRIMMED_CURVE_2D
- ANN_COMPOSITE_CURVE_2D

Attribute: Die Parameter sind identisch mit den Parametern der entsprechenden Geometrie-Entities ohne den Vorsatz ANN_. Die einzige Ausnahme ist, daß ANN_Entities ihrerseits nur auf ANN_Entities verweisen dürfen.

4.4.3 Entity LINE_ATTRIBUTES

Mit diesem Entity werden die Linienattribute, nämlich Linienbreite (LINE_WIDTH), Linienart (LINE_STYLE) und Farbe (COLOR) zusammengefaßt. Diesen drei Attributen können dann Linien zugeordnet werden.

Attribute:	Integer	Interne Nummer der Linienbreite
	Integer	Interne Nummer der Linienart
	Integer	Interne Nummer der Farbe

Beispiel:

```
@402 = LINE_ATTRIBUTES ( 1 , 1 , 1 ) ;
/* 0.25 DURCHGEZ. SCHWARZ */
```

4.4.4 Bereich Text

Dieser Bereich beschreibt die Darstellung von Texten.

4.4.4.1 Entity TEXT

Dieses Entity definiert einen String als Text.

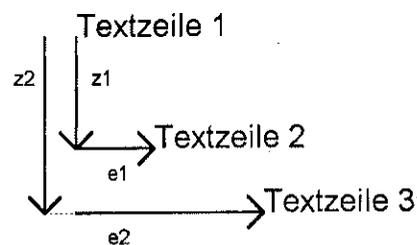
Attribute:	String	zu zeichnender Text (s. Kap 4.4.7)
------------	--------	------------------------------------

Beispiel:

```
@410 = TEXT ( "KÜCHE" ) ;
```

4.4.4.2 Entity TEXT_BLOCK

Mit diesem Entity werden Textblöcke beschrieben. Die einzelnen Textzeilen stehen stets untereinander. Die Zeilen dürfen jedoch eingerückt sein. Der Textblock kann einen Rahmen haben.



Attribute: Real Reziproker Maßstab, für den z und e gelten

Liste von

Referenz auf Entity TEXT

Real Zeilenabstand z

Real Einrückung e

usw.

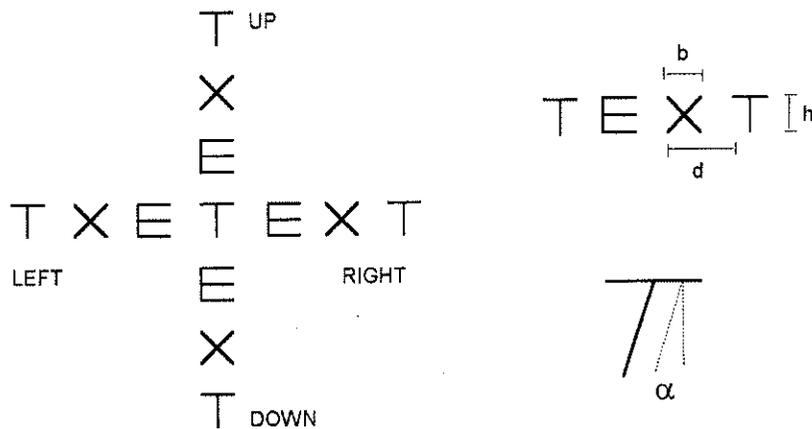
Referenz (optional) Verweis auf den Textrahmen nur
ANN_COMPOSITE_CURVE_2D ist zulässig

Beispiel:

```
@420 = TEXT ( "Textzeile 1" ) ;
@421 = TEXT ( "Textzeile 2" ) ;
@422 = TEXT ( "Textzeile 3" ) ;
@423 = TEXT_BLOCK ( 1 , ( #420 , 0 , 0 ,
                          #421 , 10 , 8 ,
                          #422 , 15 , 29 ) ) ;
```

4.4.4.3 Entity TEXT_ATTRIBUTES

Mit diesem Entity werden Textattribute zusammengefaßt.



Attribute:	String	(optional) Name des Textattributs
	Integer	Interne Nummer des Textfonts
	Real	Buchstabenhöhe h
	Real	Buchstabenbreite b
	Real	Abstand der Buchstaben untereinander d
	Real	Neigung der Buchstaben α
	String	Orientierung des Textes; zulässig sind "RIGHT", "LEFT", "UP" und "DOWN"
	Real	(optional) Reziproker Maßstab, für den diese Attribute gelten

Beispiel:

```
@430 = TEXT_ATTRIBUTES ( , 1, 10, 7, 8, 0, "RIGHT" );
```

4.4.4.4 Entity INSTANCE_TEXT

Mit diesem Entity wird Text instanziiert. Dieses Entity muß in einem Layer (s. Kap. 4.5.2) eingeschlossen sein.

Attribute:	Referenz	auf den Text zulässig sind hier nur TEXT und TEXT_BLOC
	Real	x Koordinate
	Real	y Koordinate
	Real	β Neigung des Textes



Beispiel:

```
@440 = INSTANCE_TEXT ( #423 , 48 , 12 , 0 ) ;
```

4.4.5 Bereich Bemaßung

In diesem Bereich werden Maßlinien, Maßhilfslinien, Maßzahlen und Maßbegrenzungssymbole instanziiert. Die Maßlinien und Maßhilfslinien werden dabei mit den ANN_ Entities erzeugt, die Maßzahlen mit INSTANCE_TEXT.

4.4.5.1 Entity DIMENSION_LINE

Mit diesem Entity werden Maßlinien als Verweis auf entsprechende Entities definiert.

Attribute: Liste von
 Referenz auf ANN_TRIMMED_CURVE_2D
 usw.

Beispiel:

```
@450 = DIMENSION_LINE ( ( #110 ) ) ;
```

4.4.5.2 Entity EXTENSION_LINE

Mit diesem Entity werden die Maßhilfslinien als Verweis auf Entities des Typs ANN_TRIMMED_CURVE_2D definiert.

Attribute: Liste von
 Referenz auf ANN_TRIMMED_CURVE_2D
 usw.

Beispiel:

```
@451 = EXTENSION_LINE ( ( #111 , #112 ) ) ;
```

4.4.5.3 Entity DIMENSION_TEXT

Mit diesem Entity werden die Maßzahlen einer Maßkette durch Verweise auf entsprechende Entities des Typs INSTANCE_TEXT definiert.

Attribute: Liste von
 Referenz auf INSTANCE-TEXT
 usw.

Beispiel:

```
@452 = DIMENSION_TEXT ( ( #113 ) ) ;
```

4.4.5.4 Entity INSTANCE_TERMINATOR_SYMBOL

Mit diesem Entity werden Maßbegrenzungssymbole instanziiert.

Attribute:	Integer	Interne Nummer des Maßbegrenzungssymbols
	Liste von	
	Real	x Koordinate des Symbols
	Real	y Koordinate des Symbols
	usw.	

Beispiel:

```
@453 = INSTANCE_TERMINATOR_SYMBOL ( 1, (10,0 , 10,5) );
```

4.4.5.5 Entity DIMENSION_BLOC

Mit diesem Entity wird eine Maßkette (gegebenenfalls auch ein Einzelmaß) als zusammengehörige Einheit definiert, indem die zugehörigen Entities angegeben werden.

Attribute:	Referenz	auf die zugehörige DIMENSION_LINE
	Referenz	auf die zugehörige EXTENSION_LINE
	Referenz	auf den zugehörigen INSTANCE-TEXT
	Referenz	auf das zugehörige INSTANCE-TERMINATOR-SYMBOL

Beispiel:

```
@460 = DIMENSION_BLOC ( #450 , #451 , #452 , #453 ) ;
```

4.4.5.6 Entity ASSOC_DIMENSION_LINE

Mit diesem Entity wird die Schnittbemaßung als Assoziativität definiert.

Attribute:

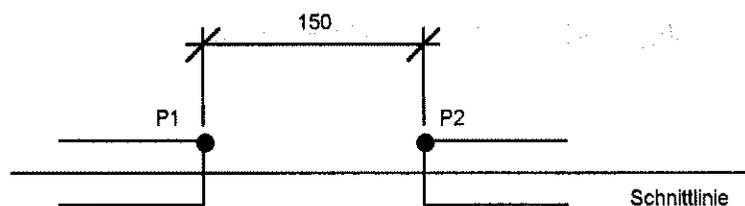
- Referenz auf ein Entity ANN_COMPOSITE_CURVE_2D, das die Schnittlinie definiert
- Referenz auf den DIMENSION BLOC, der so erzeugt wurde
- Liste von String usw. Layer, der zu berücksichtigen ist

Beispiel:

```
@470 = ASSOC_DIMENSION_LINE ( #100 , #460 ,
    ( "Waende" , "Fenster" ) ) ;
```

4.4.5.7 Entity ASSOC_DIMENSION_POINT

Mit diesem Entity wird für die Punktbemaßung die Assoziativität festgelegt. Sie gilt für die Abstandsbemaßung als Einzelbemaßung wie im Bild dargestellt.



Attribute: Referenz auf POINT_2D P1
 Referenz auf POINT_2D P2
 Referenz auf den zugehörigen DIMENSION_BLOC

Beispiel:

```
@480 = ASSOC_DIMENSION_POINT ( #20 , #21 , #460 ) ;
```

4.4.6 Entity FILL_AREA

Mit diesem Entity werden Flächen mit Schraffuren gefüllt.

Attribute: Integer Interne Schraffurnummer
 Liste von
 Referenz auf zu schraffierende Fläche AREA_2D
 usw.

Beispiel:

```
@490 = FILL_AREA ( 1, ( #261 ) ) ;
```

4.4.7 Entity FILL_AREA_COLOR

Mit diesem Entity werden Flächen mit Farben gefüllt.

Attribute: Integer Interne Farbnummer.

Liste von

Referenz auf die zu füllende Fläche AREA_2D

usw.

Beispiel:

```
@491 = FILL_AREA_COLOR ( 1, ( #261 ) ) ;
```

Anmerkung:

Dieses Entity existiert nicht in STEP_2DBS.

4.4.8 Entity PRESENT_LAYER

Mit diesem Entity werden einem Layer pauschal Attribute für Linien und Texte zugeordnet. Alle Entities des angegebenen Layers, die keine eigenen Zeichnungsattribute haben, erhalten die Zeichnungsattribute des Layers, auf dem sie existieren.

Attribute: String Name des Layers

Referenz auf ein Entity LINE_ATTRIBUTES

Referenz auf ein Entity TEXT_ATTRIBUTES

Beispiel:

```
@500 = PRESENT_LAYER ( "Waende" , #402 , #430 ) ;
```

4.4.9 Entity PRESENT_LINES

Mit diesem Entity wird ein Entity LINE_ATTRIBUTES einer Folge von Linien als Verweis auf Geometrie- und Text-Entities zugeordnet.

Attribute: Referenz auf ein Entity LINE_ATTRIBUTES

Liste von

Referenz auf ein Entity, für das dieses
Linienattribut gilt

usw.

Beispiel:

```
@501 = PRESENT_LINES ( #402 , ( #230 , #250 ) ) ;
```

4.4.10 Entity PRESENT_TEXT

Mit diesem Entity werden den INSTANCE_TEXT Entities individuell TEXT_ATTRIBUTES zugeordnet.

Attribute: Referenz auf ein Entity TEXT_ATTRIBUTES

Liste von

Referenz auf INSTANCE_TEXT, für das dieses
Textattribut gilt

usw.

Beispiel:

```
@502 = PRESENT_TEXT ( #330, ( #440 ) ) ;
```

4.4.11 Entity COMPOSE_PLAN

Dieses Entity wurde nicht von STEP_2DBS übernommen.

4.5 Gruppierungen

4.5.1 SCOPE, ENDScope

SCOPE und ENDScope werden gegenüber STEP in einer eingeschränkten Bedeutung verwendet. Mit den Schlüsselwörtern SCOPE und ENDScope werden nur Anfang und Ende der Bereiche LIBRARY, GEOMETRY_2D, AEC_ATTRIBUTES und PRESENTATION_DRAFTING markiert. Im Normalfall müssen alle Entities innerhalb eines durch SCOPE - ENDScope markierten Bereiches, die auch von außerhalb dieses Bereiches referiert werden sollen, als Attribute in ENDScope aufgeführt werden.

Beispiel:

```
@110 = GEOMETRY_2D
SCOPE ;
@111 = POINT_2D ( 0 , 0 ) ;
@112 = DIRECTION_2D ( 1 , 0 ) ;
@113 = LINE_2D ( #111 , #112 ) ;
@114 = TRIMMED_CUREVE_2D ( #113 , 0 , 1 ) ;
@115 = TRIMMED_CUREVE_2D ( #113 , 1 , 2 ) ;
ENDSCOPE ( #114 ) ;
...
@500 = PRESENTATION_DRAFTING SCOPE ;
...
@502 = PRESENT-LINES ( ... , ( #114 , #115 ) );
```

In diesem Beispiel kann in Entity 502 nur Entity 114 referiert werden, da es in der ENDScope-Liste aufgeführt wurde. Die Referenz auf Entity 115 ist falsch.

Da bei der Verwendung innerhalb von VDABS im Regelfall alle Entities innerhalb SCOPE und ENDScope referierbar sein sollen, wird dafür die Notation

```
ENDSCOPE ( "ALL" ) ;
```

zugelassen.

4.5.2 Layer

Die Layertechnik ist eine im Bauwesen weitverbreitete CAD-Technik. Sie wird nicht nur zur Strukturierung der projizierten Gebäudegeometrie, sondern auch bei der Bemaßung, Beschriftung und Schraffur eingesetzt. Layer können jederzeit und mehrfach geöffnet und geschlossen werden. Allerdings darf an jeder Stelle immer nur ein Layer offen sein.

Einige Entities müssen immer in einem geöffneten Layer liegen. Diese sind:

- TRIMMED_CURVE_2D und ANN_TRIMMED_CURVE_2D
- POLYGON_2D
- INSTANCE_2D
- INSTANCE_SYMBOL_2D
- INSTANCE_TEXT
- INSTANCE_TERMINATOR_SYMBOL

Einige Entities können in einem geöffneten Layer liegen. Diese sind:

- CIRCLE_2D und ANN_CIRCLE_2D
- ELLIPSE_2D und ANN_ELLIPSE_2D

Liegen sie innerhalb des Layers, so werden sie gezeichnet. Dienen sie jedoch nur als Hilfsgeometrie für ein TRIMMED_CURVE_2D bzw. ANN-TRIMMED-CURVE_2D, so sollten sie außerhalb des Layers liegen, um ihren getrimmten Anteil nicht zu überdecken.

4.5.2.1 Entity OPEN_LAYER

Mit diesem Entity wird der Beginn eines Layers angezeigt. Bis zum nächsten Entity CLOSE_LAYER gehören alle Entities zu diesem Layer.

Attribute: String Name des Layers

Beispiel:

```
@100 = OPEN_LAYER ( "WAENDE" ) ;
```

4.5.2.1 Entity CLOSE_LAYER

Mit diesem Entity wird das Ende des offenen Layers angezeigt.

Attribute: keine

Beispiel:

```
@160 = CLOSE_LAYER ;
```

4.5.3 Entity GROUP_2D

Mit diesem Entity werden Gruppen definiert, die später bei der Planmontage beliebig oft mittels INSTANCE_2D instanziiert werden können. Eine Gruppe kann jegliche Art von Informationen enthalten wie Geometrie, zeichnungstechnische Elemente, Bemaßung, Beschriftung, Schraffur etc. Die Schachtelung von Instanzierungen von Gruppen ist uneingeschränkt.

Attribute: String Name der Gruppe
 Liste von
 Referenz auf beliebiges (außer GROUP_2D)
 Entity
 usw.

Beispiel:

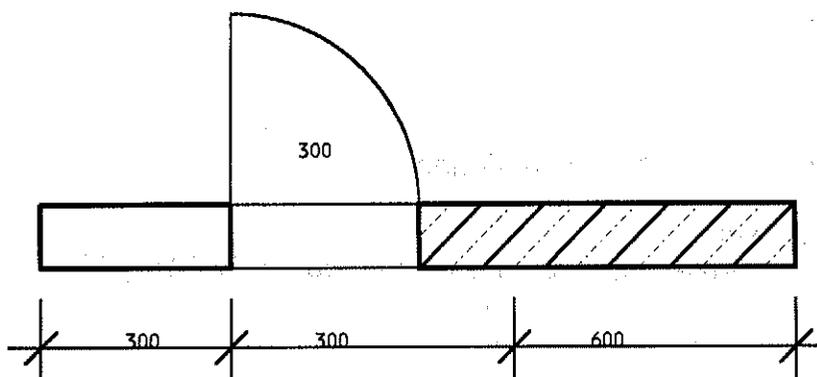
```
@120 = OPEN_LAYER ( "FENSTER" ) ;
@122 = TRIMMED_CURVE_2D ( #101 , 0 , 1 ) ;
@124 = TRIMMED_CURVE_2D ( #103 , 0 , 1 ) ;
@126 = INSTANCE_TEXT ( #105 , 10 , 10 , 0 ) ;
@128 = CLOSE_LAYER ;
@140 = GROUP_2D ("RUNDES FENSTER", (#101, #103,#105));
```

Die so definierte Gruppe kann später wie folgt benutzt werden:

```
@200 = OPEN_LAYER ( "AUSSCHNITT" ) ;
@202 = INSTANCE_2D (#140,0,0,0,-1,1,0); /* GEDREHT */
@204 = INSTANCE_2D (#140,0,20,2,0,0,1); /* SKALIERT */
@206 = CLOSE_LAYER ;
```

5 Beispiel

Das in der folgenden Figur dargestellte Beispiel zeigt eine Wand mit einer Tür:



Der Austauschfile beginnt mit dem leeren Header

```
VDABS_START;
HEADER;
ENDSEC;
DATA;
```

Der Bereich Library enthält die Bibliothekselemente

```
@1 = LIBRARY SCOPE;

@2 = UNITS ("MM", "BOGENMASS");

@4 = LINE_WIDTH (1, "LINIENBREITE 13", 0.13);
@5 = LINE_WIDTH (3, "LINIENBREITE 25", 0.25);
@6 = LINE_WIDTH (4, "LINIENBREITE 35", 0.35);

@7 = LINE_STYLE (3, "STRICHPUNKTIERT",
(0.4, 0.2, 0.1, 0.2), 50);
```

```
@8 = COLOR (1, "SCHWARZ");
@9 = TEXT_FONT (1, "SENKRECHTE NORMSCHRIFT");
@10 = HATCH (1, "BETON",
            ( 1, 0, 0, 0.78548, 50,
              3, 0, 35.355339, 0.78548, 50) );
```

Definition des Symbols TUER

```
@21 = LINE_2D (0, 0, 0, 300);
@22 = CIRCLE_2D (300, 0, 0);
@23 = OPEN_LAYER ("TUER");
@24 = TRIMMED_CURVE_2D (#22, 0, 1.570796);
@25 = TRIMMED_CURVE_2D (#21, 0, 1);
@26 = CLOSE_LAYER ;
@27 = COMPOSITE_CURVE_2D ((#24, #25));
@28 = SYMBOL (1, "TUER", (#27));
```

Definition des Maßbegrenzungssymbols

```
@31 = ANN_LINE_2D (0,0,1,1);
@32 = OPEN_LAYER ("BEMASSUNG");
@33 = ANN_TRIMMED_CURVE_2D (#31, -10, 10);
@34 = CLOSE_LAYER ;
@35 = ANN_COMPOSITE_CURVE_2D ((#33));
@36 = TERMINATOR_SYMBOL (1, "SCHRAEGSTRICH", #35);
```

```
ENDSCOPE ("ALL");
```

Der Bereich Geometry beginnt mit der Definition hilfreicher Punkte und Richtungen

```
@101 = GEOMETRY_2D SCOPE;

@102 = POINT_2D (0, 0);
@103 = POINT_2D (300, 0);
@104 = POINT_2D (600, 0);
@105 = POINT_2D (1200, 0);
@106 = POINT_2D (0, 100);

@107 = DIRECTION_2D (300, 0);
@108 = DIRECTION_2D (0, 100);
```

Mit diesen Punkten und Richtungen werden dann die Geraden definiert

```
@109 = LINE_2D (#102, #107);
@110 = LINE_2D (#106, #107);
@111 = LINE_2D (#102, #108);
@112 = LINE_2D (#103, #108);
@113 = LINE_2D (#104, #108);
@114 = LINE_2D (#105, #108);
```

Aus den Geraden werden zeichenbare Liniensegmente ausgeschnitten

```
@115 = OPEN_LAYER ("KONSTRUKTIONSLINIEN");
@116 = TRIMMED_CURVE_2D (#109, 0, 1);
@117 = TRIMMED_CURVE_2D (#109, 1, 2);
@118 = TRIMMED_CURVE_2D (#109, 2, 4);
@119 = TRIMMED_CURVE_2D (#110, 0, 1);
@120 = TRIMMED_CURVE_2D (#110, 1, 2);
@121 = TRIMMED_CURVE_2D (#110, 2, 4);
@122 = TRIMMED_CURVE_2D (#111, 0, 1);
@123 = TRIMMED_CURVE_2D (#112, 0, 1);
@124 = TRIMMED_CURVE_2D (#113, 0, 1);
@125 = TRIMMED_CURVE_2D (#114, 0, 1);
@126 = CLOSE_LAYER ;
```

Diese wiederum werden zu Linienzügen zusammengefaßt

```
@127 = COMPOSITE_CURVE_2D ((#116, #117, #118, #125,
    #121, #120, #119, #122));
@128 = COMPOSITE_CURVE_2D ((#117, #124, #120, #123));
```

Die im Bereich Library definierte Tür wird hier instanziiert

```
@129 = OPEN_LAYER ("TUER");
@130 = INSTANCE_SYMBOL (1, 300, 100, 1, 0, 0, 1);
@131 = CLOSE_LAYER ;
```

```
ENDSCOPE ("ALL");
```

Im Bereich Presentation_Drafting werden zunächst Linien- und Textattribute zusammengefaßt

```
@140 = PRESENTATION_DRAFTING SCOPE;  
  
@141 = LINE_ATTRIBUTES (1, 1, 1);  
@142 = LINE_ATTRIBUTES (3, 1, 1);  
@143 = LINE_ATTRIBUTES (4, 1, 1);  
@144 = TEXT_ATTRIBUTES (, 1, 4, 2, 3, 0, "RIGHT", 10);  
@145 = LINE_ATTRIBUTES (1, 2, 1);
```

Beschriftung des Türbogens

```
@150 = OPEN_LAYER ("TEXT");  
@151 = TEXT ("300");  
@152 = INSTANCE_TEXT (#151, 400, 200, 0);  
@153 = CLOSE_LAYER ;
```

Schraffur des rechten Wandstückes

```
@160 = OPEN_LAYER ("SCHRAFFUR");  
@161 = COMPOSITE_CURVE_2D ((#118, #125, #121, #124));  
@162 = AREA_2D ((#161));  
@163 = FILL_AREA (1, (#162));  
@164 = CLOSE_LAYER ;
```

Definition der Bemaßung

```
@170 = ANN_POINT_2D (-10, -100);  
@171 = ANN_DIRECTION_2D (1220, 0);  
@172 = ANN_LINE_2D (#170, #171);  
@173 = ANN_POINT_2D (0, -120);  
@174 = ANN_POINT_2D (300, -120);  
@175 = ANN_POINT_2D (600, -120);  
@176 = ANN_POINT_2D (1200, -120);  
@177 = ANN_DIRECTION_2D (0, 70);  
@178 = ANN_LINE_2D (#173, #177);  
@179 = ANN_LINE_2D (#174, #177);  
@180 = ANN_LINE_2D (#175, #177);  
@181 = ANN_LINE_2D (#176, #177);  
@182 = TEXT ("300");  
@183 = TEXT ("600");
```

```

@184 = OPEN_LAYER ("BEMASSUNG");
@185 = ANN_TRIMMED_CURVE_2D (#172, 0, 1);
@186 = ANN_TRIMMED_CURVE_2D (#178, 0, 1);
@187 = ANN_TRIMMED_CURVE_2D (#179, 0, 1);
@188 = ANN_TRIMMED_CURVE_2D (#180, 0, 1);
@189 = ANN_TRIMMED_CURVE_2D (#181, 0, 1);
@190 = INSTANCE_TEXT (#182, 100, -100, 0);
@191 = INSTANCE_TEXT (#182, 400, -100, 0);
@192 = INSTANCE_TEXT (#183, 850, -100, 0);
@194 = INSTANCE_TERMINATOR_SYMBOL (1,
    ( 0, -100, 300, -100,
      600, -100, 1200, -100 ));
@195 = CLOSE_LAYER ;

@294 = DIMENSION_LINE ((#185));
@295 = EXTENSION_LINE ((#186, #187, #188, #189));
@296 = DIMENSION_TEXT ((#190, #191, #192));
@297 = DIMENSION_BLOC (#294, #295, #296, #194);

```

Definition der Assoziativität der Bemaßung

```

@310 = ANN_POINT_2D (-10, 50);
@311 = ANN_LINE_2D (#310, #171);
@312 = OPEN_LAYER ("KONSTRUKTIONSLINIEN");
@313 = ANN_TRIMMED_CURVE_2D (#311, 0, 1);
@314 = CLOSE_LAYER ;
@315 = ANN_COMPOSITE_CURVE_2D ((#313));
@316 = ASSOC_DIMENSION_LINE (#315, #297,
    ("KONSTRUKTIONSLINIEN"));

```

Zuordnung von Linien- und Textattributen zu den Entities

```

@321 = PRESENT_LAYER ("KONSTRUKTIONSLINIEN",
    #143, #144);
@322 = PRESENT_LINES (#141, (#117, #120, #152));
@323 = PRESENT_LAYER ("TUER", #141, #144);
@324 = PRESENT_LAYER ("TEXT", #143, #144);
@325 = PRESENT_LAYER ("SCHRAFFUR", #142, #144);
@326 = PRESENT_LAYER ("BEMASSUNG", #141, #144);
@327 = PRESENT_LINES (#145, (#313));

```

Abschluß des Austauschfiles

```

ENDSCOPE ("ALL");
ENDSEC;
VDABS_END;

```

6 Literatur

- [1] Altmueller, J., The STEP-File Structure Version 10.0 August 1987, Document N 160
- [2] Wilson, P. R., Kennicott, P. R., STEP/PDES Testing Draft St. Louis Edition, September 1987, Document N 165
- [3] Schenck, D., Express, A Language for Information Modeling April 1987, Document N 119
- [4] Schenck, D.; A. Position, Scope Link and Property, September 1987, Document N 162.
- [5] ISO 10303-21 Clear Text Encoding of Exchange Structure; January 93, Document N 180.