








ISOBUS Virtual Terminal: Benutzeroberfläche und Objekte (2)



Franz Höpfinger


Inhaltsverzeichnis

	myst: enable_extensions: [„colon_fence“] html_meta: „description lang=en“: 9 „metadata description“ „description lang=de“: „metadata description“ „description lang=fr“: „description des métadonnées“ „keywords“: „Sphinx, MyST“ „property=og:locale“: „de“	
1	Wiki 2: ISOBUS Virtual Terminal	11
1.1	Meisterschulen am Ostbahnhof, München	11
2	AuxAssignOfApp	13
3	Bitmaps einfügen	15
3.1	PNG finden	15
3.2	PNG herunterladen	15
3.3	ImageMagick®	15
3.3.1	1. Alle Dateien unter img_original werden um 2 Pixel ringsum zugeschnitten:	15
3.3.2	2. Alle Dateien unter img_cropped werden auf 64x64 skaliert	15
3.3.3	3. umwandlung auf 2 Farben (Monochrome)	16
3.3.4	4. Konvertierung ins Format BMP	16
3.4	Paint	16
3.4.1	PNG in BMP umwandeln	16
3.4.2	BMP Skalieren	16
3.4.3	BMP Farbtiefe reduzieren	17
3.5	BMP in ISO-Designer einfügen	17
4	Einträge von Schülern	19
5	Welcome to the ISOBUS-VT-Objects wiki!	21
5.1	 Podcast	21
6	ISOBUS Objekt IDs	23
7	ISOBUS-Objekte-Versionen	29
8	ISOBUS Objekte	31
8.1	Programming	31
8.2	VT Levels	32
9	ISOBUS Wiki Links	33
10	Jetter ISO Designer Workspace öffnen	35
11	Jetter ISO-Designer	39

12	Leere Objekte (Minimum Child Objects)	43
12.1	Objekte, die NICHT leer sein dürfen (Minimum 1 Child)	43
12.2	Objekte, die leer sein dürfen (Minimum 0 Children)	43
12.2.1	Besonderheit: Input List (ID 10)	44
13	Objekthierarchie (Parent-Child Beziehungen)	45
13.1	Parent: Working Set object (ID 0)	45
13.2	Parent: Data Mask object (ID 1)	45
13.3	Parent: Alarm Mask object (ID 2)	46
13.4	Parent: Container object (ID 3)	47
13.5	Parent: Window Mask object (ID 34)	48
13.6	Parent: Soft Key Mask object (ID 4)	48
13.7	Parent: Key object (ID 5)	49
13.8	Parent: Button object (ID 6)	49
13.9	Parent: Key Group object (ID 35)	50
13.10	Parent: Input List object (ID 10)	50
13.11	Parent: Output List object (ID 37)	51
13.12	Parent: Auxiliary Function Type 1 object (ID 29)	51
13.13	Parent: Auxiliary Input Type 1 object (ID 30)	52
13.14	Parent: Auxiliary Function Type 2 object (ID 31)	52
13.15	Parent: Auxiliary Input Type 2 object (ID 32)	52
13.16	Parent: Object Label graphic representation	53
13.17	Parent: Object Label Reference List object (ID 40)	53
13.18	Parent: Animation object (ID 44)	55
14	Ungültige Objektpools (Invalid Object Pools)	57
14.1	1. Strukturelle Integrität (Parsing-Fehler)	57
14.2	2. Referenzierungs-Fehler (Dangling Pointers)	57
14.3	3. Logische Fehler & Rekursion	57
14.4	4. Spezifische Objekt-Regeln (Hard Constraints)	57
14.5	5. Ressourcen-Limits	58
14.6	Tipps zur Fehlersuche	58
15	Vektorgrafiken erstellen	59
15.1	Coral Draw X5	59
15.2	Solid Edge	59
15.3	Libre CAD	59
15.4	ISO-Designer	60
15.5	Alternative mit Autodesk Fusion 360 und Plugin	61
15.5.1	Fusion 360 anwenden.	61
16	 Videos	65
16.1	ISO-Designer: Deutsch	65
16.2	ISO-Designer: englisch	65

17	logiBUS®	67
17.1	 Podcast	67
17.2	 Video	67
18	Objekte	69
18.1	Attribute	69
18.1.1	Empfohlene Lektüre:	69
18.1.2	 Podcast	70
18.2	Ausgabe	70
18.3	Auxiliary Funktionen	70
18.4	Bedienelemente	70
18.5	Bilder	70
18.6	Container	70
18.6.1	 Podcast	70
18.7	Eingabe	71
18.8	Externe Objekte	71
18.9	Funktionen	71
18.10	Masken	71
18.10.1	 Podcast	71
18.11	Spezial & Sonstiges	71
18.12	Variablen	71
19	Referenztabellen (Annex A)	73
19.1	Kommando-Übersicht (ISO 11783-6 Table A.5)	73
19.2	Event IDs (ISO 11783-6 Table A.3)	74
19.3	Standard Farbpalette (ISO 11783-6 Table A.4)	75
20	Isobus objects	77
20.1	ID 0 – Working set – ISO 11783-6 – B.1	77
20.1.1	Wichtige Eigenschaften (gemäß ISO 11783-6, Anhang B.1)	77
20.1.2	Bedeutung für die Implementierung	79
20.2	ID 1 – Data mask – ISO 11783-6 – B.2	79
20.2.1	Ereignisse (Events - Tabelle B.3)	81
20.2.2	Verhalten und Einschränkungen	81
20.2.3	Bedeutung für die Implementierung	81
20.3	ID 2 – Alarm Mask – ISO 11783-6 – B.3	81
20.3.1	Ereignisse (Events - Tabelle B.5)	82
20.3.2	Verhalten bei mehreren Alarmen	83
20.3.3	Bedeutung für die Implementierung	83
20.4	ID 3 – Container – ISO 11783-6 – B.4	83
20.4.1	Ereignisse (Events - Tabelle B.7)	84
20.4.2	Nutzung in der Praxis	85
20.4.3	Bedeutung für die Implementierung	85
20.5	ID 4 – Soft Key Mask – ISO 11783-6 – B.5	86
20.5.1	Ereignisse (Events - Tabelle B.9)	87
20.5.2	Zusammenspiel mit Datenmasken	87
20.5.3	Bedeutung für die Implementierung	87

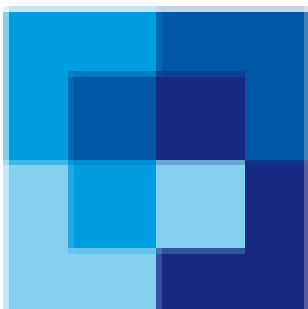
20.5.4	 Podcast	87
20.6	ID 5 – Key (Soft Key) – ISO 11783-6 – B.6	88
20.6.1	 Podcast	88
20.6.2	Ereignisse (Events - Tabelle B.11)	89
20.6.3	Der Key Code	89
20.6.4	Bedeutung für die Implementierung	90
20.7	ID 6 – Button – ISO 11783-6 – B.7	90
20.7.1	Ereignisse (Events - Tabelle B.13)	92
20.8	ID 7 – Input boolean – ISO 11783-6 – B.8.2	94
20.8.1	Ereignisse (Events - Tabelle B.15)	95
20.8.2	Bedeutung für die Implementierung	96
20.9	ID 8 – Input string – ISO 11783-6 – B.8.3	96
20.9.1	Funktionsweise und Optionen	98
20.9.2	Ereignisse (Events - Tabelle B.15)	98
20.9.3	Bedeutung für die Implementierung	98
20.10	ID 9 – Input number – ISO 11783-6 – B.8.4	98
20.10.1	Die Skalierungslogik	100
20.10.2	Validierung	100
20.10.3	Ereignisse (Events - Tabelle B.15)	100
20.10.4	Real Time Editing (AID 15, Bit 1)	101
20.10.5	Bedeutung für die Implementierung	101
20.11	ID 10 – Input list – ISO 11783-6 – B.8.5	101
20.11.1	Funktionsweise und Darstellung	102
20.11.2	Ereignisse (Events - Tabelle B.19)	102
20.11.3	Bedeutung für die Implementierung	102
20.12	ID 11 – Output string – ISO 11783-6 – B.9.2	103
20.12.1	Funktionsweise und Besonderheiten	104
20.12.2	Statischer Text vs. Dynamische Variable	104
20.12.3	Ereignisse (Events - Tabelle B.21)	104
20.12.4	Bedeutung für die Implementierung	105
20.13	ID 12 – Output number – ISO 11783-6 – B.9.3	105
20.13.1	Die Skalierungslogik	106
20.13.2	Formatierungsoptionen	106
20.13.3	Ereignisse (Events)	107
20.13.4	Bedeutung für die Implementierung	107
20.14	ID 13 – Output line – ISO 11783-6 – B.10.2	107
20.14.1	Funktionsweise und Geometrie	108
20.14.2	Ereignisse (Events - Tabelle B.26)	108
20.14.3	Bedeutung für die Implementierung	108
20.15	ID 14 – Output rectangle – ISO 11783-6 – B.10.3	108
20.15.1	Darstellung und Eigenschaften	109
20.15.2	Geometrische Berechnung	109
20.15.3	Ereignisse (Events - Tabelle B.28)	109
20.15.4	Bedeutung für die Implementierung	110
20.16	ID 15 – Output ellipse – ISO 11783-6 – B.10.4	110
20.16.1	Ellipsentypen und Geometrie	111
20.16.2	Winkelberechnung (Wichtig!)	111
20.16.3	Ereignisse (Events - Tabelle B.30)	111
20.16.4	Bedeutung für die Implementierung	111
20.17	ID 16 – Output polygon – ISO 11783-6 – B.10.5	112
20.17.1	Polygontypen und Füllregeln	113
20.17.2	Geometrie und Punkte	113

20.17.3	Ereignisse (Events - Tabelle B.32)	113
20.17.4	Bedeutung für die Implementierung	113
20.18	ID 17 – Output meter – ISO 11783-6 – B.11.2	113
20.18.1	Funktionsweise und Darstellung	115
20.18.2	Deflektionsrichtung (AID 5, Bit 3)	115
20.18.3	Ereignisse (Events - Tabelle B.34)	115
20.18.4	Bedeutung für die Implementierung	115
20.19	ID 18 – Output linear bar graph – ISO 11783-6 – B.11.3	115
20.19.1	Funktionsweise und Optionen	117
20.19.2	Skalierung	117
20.19.3	Ereignisse (Events - Tabelle B.36)	117
20.19.4	Bedeutung für die Implementierung	118
20.19.5	 Podcast	118
20.20	ID 19 – Output arched bar graph – ISO 11783-6 – B.11.4	118
20.20.1	Funktionsweise und Geometrie	119
20.20.2	Ereignisse (Events - Tabelle B.38)	120
20.20.3	Bedeutung für die Implementierung	120
20.21	ID 20 – Picture graphic – ISO 11783-6 – B.12.2	120
20.21.1	Funktionsweise und Darstellung	121
20.21.2	Ereignisse (Events - Tabelle B.40)	121
20.21.3	Bedeutung für die Implementierung	122
20.22	ID 21 – Number variable – ISO 11783-6 – B.13.2	122
20.22.1	Funktionsweise und Referenzierung	122
20.22.2	Ereignisse (Events - Tabelle B.42)	122
20.22.3	Bedeutung für die Implementierung	122
20.23	ID 22 – String variable – ISO 11783-6 – B.13.3	123
20.23.1	Funktionsweise und Besonderheiten	123
20.23.2	Referenzierung und Aktualisierung	123
20.23.3	Ereignisse (Events - Tabelle B.42)	123
20.23.4	Bedeutung für die Implementierung	123
20.24	ID 23 – Font attributes – ISO 11783-6 – B.14.2	124
20.24.1	Schriftgrößen und Render-Modi	124
20.24.2	Ereignisse (Events - Tabelle B.45)	125
20.24.3	Bedeutung für die Implementierung	125
20.25	ID 24 – Line attributes – ISO 11783-6 – B.14.3	125
20.25.1	Linienstärken und Darstellung	126
20.25.2	Linienstil (Line Art - AID 3)	126
20.25.3	Ereignisse (Events - Tabelle B.47)	126
20.25.4	Bedeutung für die Implementierung	126
20.26	ID 25 – Fill attributes – ISO 11783-6 – B.14.4	126
20.26.1	Fülltypen und Logik	127
20.26.2	Verwendung von Füllmustern (Wichtig!)	127
20.26.3	Ereignisse (Events - Tabelle B.49)	127
20.26.4	Bedeutung für die Implementierung	127
20.27	ID 26 – Input attributes – ISO 11783-6 – B.14.5	128
20.27.1	Funktionsweise und Validierung	128
20.27.2	Ereignisse (Events - Tabelle B.51)	128
20.27.3	Bedeutung für die Implementierung	129
20.28	ID 27 – Object pointer – ISO 11783-6 – B.15	129
20.28.1	Funktionsweise und Anwendung	129
20.28.2	Ereignisse (Events - Tabelle B.54)	129
20.28.3	Bedeutung für die Implementierung	129

20.29 ID 28 – Macro – ISO 11783-6 – B.16	130
20.29.1 Funktionsweise und Struktur	130
20.29.2 Verfügbare Makro-Befehle (Auszug)	130
20.29.3 Ereignisse (Events)	131
20.29.4 Bedeutung für die Implementierung	131
20.30 ID 29 – Auxiliary Function Type 1 – ISO 11783-6 – J.4.2	131
20.30.1 Funktionsweise	132
20.31 ID 30 – Auxiliary Input Type 1 – ISO 11783-6 – J.4.4	132
20.31.1 Funktionsweise	133
20.32 ID 31 – Auxiliary Function Type 2 – ISO 11783-6 – J.4.3	133
20.32.1 Besonderheiten	134
20.33 ID 32 – Auxiliary Input Type 2 – ISO 11783-6 – J.4.5	134
20.33.1 Funktionsweise	135
20.34 ID 33 – Auxiliary Control Designator Type 2 Object Pointer – ISO 11783-6 – J.4.7	136
20.34.1 Ereignisse (Events - Tabelle J.7)	136
20.34.2 Nutzen für den Entwickler	136
20.35 ID 34 – Window Mask – ISO 11783-6 – B.19	137
20.35.1 Ereignisse (Events - Tabelle B.60)	139
20.36 ID 35 – Key Group – ISO 11783-6 – B.20	139
20.36.1 Ereignisse (Events - Tabelle B.62)	140
20.36.2 Bedeutung und Funktionsweise	140
20.37 ID 36 – Graphics Context Object – ISO 11783-6 – B.18	140
20.37.1 Funktionsweise und Struktur	142
20.37.2 Graphics Context Commands	142
20.37.3 Ereignisse (Events - Tabelle B.58)	142
20.37.4 Anwendungsbeispiel	143
20.38 ID 37 – Output List – ISO 11783-6 – B.9.4	143
20.38.1 Funktionsweise	144
20.38.2 Ereignisse (Events - Tabelle B.24)	144
20.39 ID 38 – Extended Input Attributes – ISO 11783-6 – B.14.6	144
20.39.1 Funktionsweise	145
20.40 ID 39 – Colour Map – ISO 11783-6 – B.17	145
20.40.1 Funktionsweise	146
20.40.2 Anmerkung zu VT Version 6	146
20.41 ID 40 – Object Label Reference List – ISO 11783-6 – B.21	146
20.41.1 Bedeutung und Funktionsweise	147
20.42 ID 41 – External Object Definition – ISO 11783-6 – B.22	147
20.42.1 Funktionsweise	148
20.42.2 Ereignisse (Events - Tabelle B.65)	148
20.43 ID 42 – External Reference NAME – ISO 11783-6 – B.23	148
20.43.1 Funktionsweise	148
20.43.2 Ereignisse (Events - Tabelle B.67)	148
20.44 ID 43 – External Object Pointer – ISO 11783-6 – B.24	149
20.44.1 Funktionsweise und Regeln	149
20.44.2 Ereignisse (Events - Tabelle B.69)	149
20.44.3 Anwendungsbeispiel	150
20.45 ID 44 – Animation – ISO 11783-6 – B.25	150
20.45.1 Funktionsweise	151
20.45.2 Ereignisse (Events - Tabelle B.71)	152

20.45.3 Empfehlung	152
20.46 ID 45 – Colour Palette – ISO 11783-6 – B.26	152
20.46.1 Funktionsweise	152
20.46.2 Unterschied zur Colour Map (ID 39)	153
20.46.3 Ereignisse (Events)	153
20.47 ID 46 – Graphic Data (PNG) – ISO 11783-6 – B.27	153
20.47.1 Besonderheiten	153
20.47.2 Verwendung	153
20.48 ID 47 – Working Set Special Controls – ISO 11783-6 – B.29	154
20.48.1 Bedeutung und Funktionsweise	154
20.48.2 Ereignisse (Events - Tabelle B.77)	155
20.49 ID 48 – Scaled Graphic – ISO 11783-6 – B.28	155
20.49.1 Ereignisse (Events - Tabelle B.75)	156
20.50 ISOBUS VT Objekte	156
20.50.1 Liste der Objekte	156

0.1 `myst: enable_extensions: [„colon_fence“] html_meta: „description lang=en“: „metadata description“ „description lang=de“: „metadata description“ „description lang=fr“: „description des métadonnées“ „keywords“: „Sphinx, MyST“ „property=og:locale“: „de“`



**Meisterschulen
am Ostbahnhof**




1. Wiki 2: ISOBUS Virtual Terminal

1.1 Meisterschulen am Ostbahnhof, München

Willkommen bei der Dokumentation zu ISOBUS Virtual Terminal.

Diese Dokumentation ist Teil der Wissensdatenbank der Meisterschulen am Ostbahnhof München.

Nützliche Links:

-  [Hauptmenü](#)
-  [Super-Suche \(alle Wikis\)](#)
-  [PDF-Handbuch herunterladen](#)

!!! note This project is under active development.

Contents

 [Hauptmenü](#) |  [Super-Suche \(alle Wikis\)](#) | [Schnelle Suche \(IEC 61499\)](#)

2. AuxAssignOfApp

AuxAssignOfApp.txt
findet sich auf dem logiBUS®-Webserver:

ESP32 File Server

Current Directory: /data/

Name	Type	Size	Version	Chip	Update	Delete
DefaultPool.iop	file	22 kB		N/A		Delete
test_FORTE_PC.fboot	file	353 kB		N/A		Delete
AuxAssignOfApp.txt	file	0 kB		N/A		Delete

384.0 kB of 8088.0 kB used (95 % free), 7704.0 kB remaining

Show [Version](#) of running application.

Dateiformat:

```
339 2 2 31002 28001 cfb6d2a008100a0
339 2 2 31003 28003 cfb6d2a008100a0
```

```
339 2 2 31002 28001 cfb6d2a008100a0
339 2 2 31003 28003 cfb6d2a008100a0
```

dabei bedeutet

	Beschreibung
339	Hersteller Code
2	ModelIdentCode
2	Aux Typ (in diesem Fall Boolean non Latching)
31002	AuxFunction Objekt ID am Implement
28001	AuxInput Objekt ID am Joystick
cfb6d2a008100a0	ISO-Name des Joystick

3. Bitmaps einfügen

!!! note in der ISO 11783-6 werden für VT3/UT2 keine PNG Grafiken unterstützt. siehe auch [ISOBUS-Objekte-Versionen](#)

3.1 PNG finden

<https://www.iso.org/obp/>

3.2 PNG herunterladen

Rechtsklick auf das Symbol, und Speichern, idealerweise im Ordner:

C:\git\ms\4diac_training1\Ventilsteuerung\ISO-DesignerProjects\Workspace\DefaultPool\img
andere Ordner sind möglich, aber der Ordnung halber sei dieser empfohlen.

3.3 ImageMagick®

<https://imagemagick.org/> kann das umwandeln erleichtern, es wandelt bei Bedarf 100te Bilder mit einem Mausklick.

<https://imagemagick.org/script/download.php#windows> Download für Windows

unter: https://github.com/Meisterschulen-am-Ostbahnhof-Munchen/4diac_training1/tree/main/Ventilsteuerung/ISO-DesignerProjects/Workspace_TECU/DefaultPool/img/script



finden Sie ein Script, das, (vorausgesetzt Image Magick ist installiert) per Doppelklick folgende Prozesse ausführt:

Original:

von https://www.iso.org/obp/graphics/grs/343178e8-8b69-4dd7-83c6-cee7c66b28a4_200.png

3.3.1 1. Alle Dateien unter `img_original` werden um 2 Pixel ringsum zugeschnitten:

zum Ordner: `img_original`

vorher		nachher
	->	
200x200		196x196


dadurch werden die 2px großen Schnittmarken von <https://www.iso.org/obp/> entfernt. In der Regel will man die aufgrund des Erscheinungsbildes nicht haben.

Danach liegen die Zugeschnittenen Bilder unter „img_cropped“

3.3.2 2. Alle Dateien unter `img_cropped` werden auf 64x64 skaliert

zum Ordner: `img_cropped`

vorher		nachher
--------	--	---------

	->	.
196x196		64x64

Danach liegen die zugeschnittenen Bilder unter „img_resized“

3.3.3 3. umwandlung auf 2 Farben (Monochrome)

zum Ordner: `img_resized`

vorher		nachher
.	->	.
64x64		64x64

Danach liegen die Bilder unter „img_monochrome“

3.3.4 4. Konvertierung ins Format BMP

zum Ordner: `img_monochrome`

vorher		nachher
.	->	.
64x64		64x64

Danach liegen die Bilder unter „img“

zum Ordner: `img`

3.4 Paint

mit Microsoft Paint das in jedem Windows enthalten ist geht das umwandeln PNG in BMP ganz leicht. will man jedoch viele Dateien wandeln, so ist der Aufwand nicht unerheblich.

3.4.1 PNG in BMP umwandeln

das Bitmap geeignet benennen, die Dateinamen von ISO sind oft wenig aussagekräftig. (keine Leerzeichen, keine Umlaute) siehe: <https://docs.ms-muc-docs.de/projects/visual-programming-languages-docs/de/latest/Allgemeines.html#namen>

3.4.2 BMP Skalieren

die Größe ist wichtig:

wo	Größe			
Softkey	72x72 pixel			
Datamask	frei			
CCI A3	64x64 pixel			

Gedanken: 80x80 -> als Button, macht Innenfläche 72x72

72x72 -> als Button, macht Innenfläche 64x64

!!! note Tipp: auch bei Buttons diese Größen anwenden, macht ein klares Erscheinungsbild.

für Größere Buttons vielfache davon, z.B. 80x160, 160x160 usw.

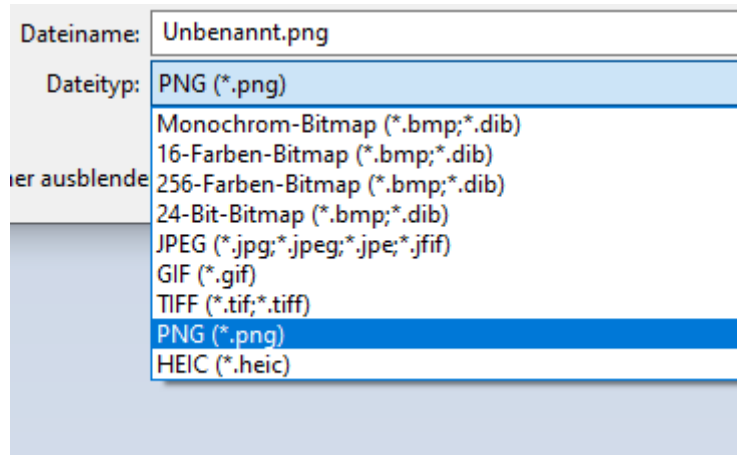
Empfehlung: wenn das Symbol als Softkey und als CCI A3 Symbol verwendet werden soll, dann 64x64 Pixel machen.

Man kann die Bilder vor dem Skalieren auch zuschneiden.

3.4.3 BMP Farbtiefe reduzieren

für ein schnelles laden die Farbtiefe so weit als möglich reduzieren.

Windows Paint bietet folgende Farbtiefen an:



Format	Farben	Bit	Hinweis
Monochrom-Bitmap	2 ¹ = 2	1	
16-Farben-Bitmap	2 ⁴ = 16	4	
256-Farben-Bitmap	2 ⁸ = 256	8	
24-Bit-Bitmap	2 ²⁴ = 16.777.216	24	in ISO 11783-6 nicht vorhanden

3.5 BMP in ISO-Designer einfügen

nun können die Bilder in ISO-Designer verwendet werden, im Falle von Monochrom kann Weiß als Transparenter Hintergrund gesetzt werden, im Falle einer anderen Farbtiefe wird oft Pink verwendet.

4. Einträge von Schülern

Schuelerbereich/tree/main/Eintraege

1. Raphael Gross [Data_Mask-Button_erstellen_u_bearbeiten.pdf](#)
2. Markus Winter [Softkey_und_Container_erstellen.pdf](#)

5. Welcome to the ISOBUS-VT-Objects wiki!

5.1 Podcast

- logiBUS®: Revolutioniert die Agrar-IT – So wird ISOBUS zum Smart Home für Landwirte

6. ISOBUS Objekt IDs

Siehe auch:

- [ISOBUS Wiki - Colours](#)
- [ISOBUS Wiki - Objectpool Objects Database](#)

Stand Jetter ISO-Designer 5.6.1

hier stimmen die Objektnummern mit den IDs in der Norm überein.

(die Objektnummern können jeden Wert annehmen; mit Ausnahme der Makros)

die Festlegung in dieser Tabelle kann man also als Good Practice bezeichnen.

TypeName	StartID	FmtStr	ID	Link	Wiki
WorkingSet	0	WorkingSet_%ld	0	ID 0 - Working-set - ISO-11783-6 - B.1	Wiki
DataMask	1000	DataMask_%ld	1	ID 1 - Data-mask - ISO-11783-6 - B.2.	Wiki
AlarmMask	2000	AlarmMask_%ld	2	ID 2 - Alarm Mask - ISO 11783-6 - B.3	Wiki
Container	3000	Container_%ld	3	ID 3 - Container - ISO 11783-6 - B.4	Wiki
SoftKeyMask	4000	SoftKeyMask_%ld	4	ID 4 - Soft key mask - ISO 11783-6 - B.5	Wiki
SoftKey	5000	SoftKey_%ld	5	ID 5 - Key (Soft Key) - ISO 11783-6 - B.6	Wiki
KeyGroup	35000	KeyGroup_%ld	35	ID 35 - Key Group - ISO 11783-6 - B.20	Wiki
Button	6000	Button_%ld	6	ID 6 - Button - ISO 11783-6 - B.7	Wiki
InputBoolean	7000	InputBoolean_%ld	7	ID 7 - Input boolean - ISO 11783-6 - B.8.2	Wiki
InputString	8000	InputString_%ld	8	ID 8 - Input string - ISO 11783-6 - B.8.3	Wiki
InputNumber	9000	InputNumber_%ld	9	ID 9 - Input number - ISO 11783-6 - B.8.4	Wiki
InputList	10000	InputList_%ld	10	ID 10 - Input list - ISO 11783-6 - B.8.5	Wiki
OutputString	11000	OutputString_%ld	11	ID 11 - Output string - ISO 11783-6 - B.9.2	Wiki
OutputNumber	12000	OutputNumber_%ld	12	ID 12 - Output number - ISO 11783-6 - B.9.3	Wiki
OutputList	37000	OutputList_%ld	37	ID 37 - Output List - ISO 11783-6 - B.9.4	Wiki
Line	13000	Line_%ld	13	ID 13 - Output line - ISO 11783-6 - B.10.2	Wiki

Rectangle	14000	Rectangle_%ld	14	ID 14 – Output rectangle – ISO 11783-6 – B.10.3	Wiki
Ellipse	15000	Ellipse_%ld	15	ID 15 – Output ellipse – ISO 11783-6 – B.10.4	Wiki
Polygon	16000	Polygon_%ld	16	ID 16 – Output polygon – ISO 11783-6 – B.10.5	Wiki
Meter	17000	Meter_%ld	17	ID 17 – Output meter – ISO 11783-6 – B.11.2	Wiki
LinearBargraph	18000	LinearBargraph_%ld	18	ID 18 – Output linear bar graph – ISO 11783-6 – B.11.3	Wiki
ArchedBargraph	19000	ArchedBargraph_%ld	19	ID 19 – Output arched bar graph – ISO 11783-6 – B.11.4	Wiki
PictureGraphic	20000	PictureGraphic_%ld	20	ID 20 – Picture graphic – ISO 11783-6 – B.12.2	Wiki
GraphicsContext	36000	GraphicsContext_%ld	36	ID 36 – Graphics Context Object – ISO 11783-6 – B.18	Wiki
NumberVariable	21000	NumberVariable_%ld	21	ID 21 – Number variable – ISO 11783-6 – B.13.2	Wiki
StringVariable	22000	StringVariable_%ld	22	ID 22 – String variable – ISO 11783-6 – B.13.3	Wiki
FontAttributes	23000	FontAttributes_%ld	23	ID 23 – Font attributes – ISO 11783-6 – B.14.2	Wiki
LineAttributes	24000	LineAttributes_%ld	24	ID 24 – Line attributes – ISO 11783-6 – B.14.3	Wiki
FillAttributes	25000	FillAttributes_%ld	25	ID 25 – Fill attributes – ISO 11783-6 – B.14.4	Wiki
InputAttributes	26000	InputAttributes_%ld	26	ID 26 – Input attributes – ISO 11783-6 – B.14.5	Wiki
ExtendedInputAttributes	38000	ExtendedInputAttributes_%ld	38	ID 38 – Extended Input Attributes – ISO 11783-6 – B.14.6	Wiki
ObjectPointer	27000	ObjectPointer_%ld	27	ID 27 – Object pointer – ISO 11783-6 – B.15	Wiki
Macro	0	Macro_%ld	0	ID 28 – Macro – ISO 11783-6 – B.16	Wiki
AuxFunction2	31000	AuxFunction2_%ld	31	ID 31 – Auxiliary Function Type 2 – ISO 11783-6 – J.4.3	Wiki
AuxInput2	32000	AuxInput2_%ld	32	ID 32 – Auxiliary Input Type 2 – ISO 11783-6 – J.4.5	Wiki
AuxObjectPointer	33000	AuxObjectPointer_%ld	33	ID 33 – Auxiliary Control Designator Type 2	Wiki

				Object Pointer – ISO 11783-6 – J.4.7	
ColorMap	39000	ColorMap_%ld	39	ID 39 – Colour Map – ISO 11783-6 – B.17	Wiki
WindowMask	34000	WindowMask_%ld	34	ID 34 – Window Mask – ISO 11783-6 – B.19	Wiki
ObjectLabelReferenceList	40000	ObjectLabelReferenceList_%ld	40	ID 40 – Object Label Reference List – ISO 11783-6 – B.21	Wiki
ExternalObjectDefinition	41000	ExternalObjectDefinition_%ld	41	ID 41 – External Object Definition – ISO 11783-6 – B.22	Wiki
ExternalReferenceName	42000	ExternalReferenceName_%ld	42	ID 42 – External reference NAME – ISO 11783-6 – B.23	Wiki
ExternalObjectPointer	43000	ExternalObjectPointer_%ld	43	ID 43 – External Object Pointer – ISO 11783-6 – B.24	Wiki
Animation	44000	Animation_%ld	44	ID 44 – Animation – ISO 11783-6 – B.25	Wiki
ScaledGraphic	48000	ScaledGraphic_%ld	48	ID 48 – Scaled Graphic – ISO 11783-6 – B.28	Wiki
GraphicData	46000	GraphicData_%ld	46	ID 46 – Graphic Data (PNG) – ISO 11783-6 – B.27	Wiki
ColorPalette	45000	ColorPalette_%ld	45	ID 45 – Colour Palette – ISO 11783-6 – B.26	Wiki
WorkingSetSpecialControls	47000	WorkingSetSpecialControls_%ld	47	ID 47 – Working Set Special Controls – ISO 11783-6 – B.29	Wiki
IDsForTemporaryUse	64000	IDsForTemporaryUse_%ld	64		
Proxy	4194304	Proxy_%ld			

In Versionen vor V. 5.6.0, also bis ISO-Designer 5.5.1

war folgende Benennung üblich:

hier stimmte die ID nicht mit der ObjectID zusammen.

z.B. bei AUX Function.

TypeName	StartID	FmtStr
WorkingSet	0	WorkingSet_%ld
DataMask	100	DataMask_%ld
AlarmMask	200	AlarmMask_%ld
Container	300	Container_%ld
SoftKeyMask	400	SoftKeyMask_%ld
SoftKey	500	SoftKey_%ld
KeyGroup	3600	KeyGroup_%ld

Button	600	Button_%ld
InputBoolean	700	InputBoolean_%ld
InputString	800	InputString_%ld
InputNumber	900	InputNumber_%ld
InputList	1000	InputList_%ld
OutputString	1100	OutputString_%ld
OutputNumber	1200	OutputNumber_%ld
OutputList	3000	OutputList_%ld
Line	1300	Line_%ld
Rectangle	1400	Rectangle_%ld
Ellipse	1500	Ellipse_%ld
Polygon	1600	Polygon_%ld
Meter	1700	Meter_%ld
LinearBargraph	1800	LinearBargraph_%ld
ArchedBargraph	1900	ArchedBargraph_%ld
PictureGraphic	2000	[File Name]_%ld
GraphicsContext	3100	GraphicsContext_%ld
NumberVariable	2100	NumberVariable_%ld
StringVariable	2200	StringVariable_%ld
FontAttributes	2300	FontAttributes_%ld
LineAttributes	2400	LineAttributes_%ld
FillAttributes	2500	FillAttributes_%ld
InputAttributes	2600	InputAttributes_%ld
ExtendedInputAttributes	3200	ExtendedInputAttributes_%ld
ObjectPointer	2700	ObjectPointer_%ld
Macro	0	Macro_%ld
AuxInput2	2800	AuxInput2_%ld
AuxFunction2	2900	AuxFunction2_%ld
AuxObjectPointer	3300	AuxObjectPointer_%ld
ColorMap	3400	ColorMap_%ld
WindowMask	3500	WindowMask_%ld
ObjectLabelReferenceList	3700	ObjectLabelReferenceList_%ld
ExternalObjectDefinition	3800	ExternalObjectDefinition_%ld
ExternalReferenceName	3900	ExternalReferenceName_%ld
ExternalObjectPointer	4000	ExternalObjectPointer_%ld
Animation	4100	Animation_%ld

IDsForTemporaryUse	64000	IDsForTemporaryUse_%ld
Proxy	4194304	Proxy_%ld

7. ISOBUS-Objekte-Versionen

Welches Objekt ist in welcher Version vorhanden ?

die Tabelle unter: https://extranet.epec.fi/Public/Manuals/EPEC_Programming_And_Libraries/projecttopics/topic000962.htm ist leider falsch und veraltet:

hier soll eine Aktuelle Tabelle entstehen, basierend auf der ISO 11783-6:2018(en) Tractors and machinery for agriculture and forestry – Serial control and communications data network – Part 6: Virtual terminal

ID	Object description	VT 2	VT 3	VT 4	VT 5	VT 6	Standard	Jetter support
0	Working set	X	X	X	X	X	11783-6 – B.1	L2 - L6
1	Data mask	X	X	X	X	X	11783-6 – B.2	L2 - L6
2	Alarm Mask	X	X	X	X	X	11783-6 – B.3	L2 - L6
3	Container	X	X	X	X	X	11783-6 – B.4	L2 - L6
4	Soft Key Mask	X	X	X	X	X	11783-6 – B.5	L2 - L6
5	Key	X	X	X	X	X	11783-6 – B.6	L2 - L6
6	Button	X	X	X	X	X	11783-6 – B.7	L2 - L6
7	Input Boolean	X	X	X	X	X	11783-6 – B.8.2	L2 - L6
8	Input String	X	X	X	X	X	11783-6 – B.8.3	L2 - L6
9	Input Number	X	X	X	X	X	11783-6 – B.8.4	L2 - L6
10	Input List	X	X	X	X	X	11783-6 – B.8.5	L2 - L6
11	Output String	X	X	X	X	X	11783-6 – B.9.2	L2 - L6
12	Output Number	X	X	X	X	X	11783-6 – B.9.3	L2 - L6
13	Output Line	X	X	X	X	X	11783-6 – B.10.2	L2 - L6
14	Output Rectangle	X	X	X	X	X	11783-6 – B.10.3	L2 - L6
15	Output Ellipse	X	X	X	X	X	11783-6 – B.10.4	L2 - L6
16	Output Polygon	X	X	X	X	X	11783-6 – B.10.5	L2 - L6
17	Output Meter	X	X	X	X	X	11783-6 – B.11.2	L2 - L6
18	Output Linear Bar Graph	X	X	X	X	X	11783-6 – B.11.3	L2 - L6
19	Output Arched Bar Graph	X	X	X	X	X	11783-6 – B.11.4	L2 - L6
20	Picture Graphic (BMP)	X	X	X	X	X	11783-6 – B.12.1 + B.12.2	L2 - L6
21	Number Variable	X	X	X	X	X	11783-6 – B.13.2	L2 - L6
22	String Variable	X	X	X	X	X	11783-6 – B.13.3	L2 - L6
23	Font Attributes	X	X	X	X	X	11783-6 – B.14.2	L2 - L6
24	Line Attributes	X	X	X	X	X	11783-6 – B.14.3	L2 - L6

25	Fill Attributes	X	X	X	X	X	11783-6 - B.14.4	L2 - L6
26	Input Attributes	X	X	X	X	X	11783-6 - B.14.5	L2 - L6
27	Object Pointer	X	X	X	X	X	11783-6 - B.15	L2 - L6
28	Macro	X	X	X	X	X	11783-6 - B.16	L2 - L6
29	Auxiliary Function Type 1	X					11783-6 - J.4.2	L6
30	Auxiliary Input Type 1	X					11783-6 - J.4.4	L6
31	Auxiliary Function Type 2		X	X	X	X	11783-6 - J.4.3	L3 - L6
32	Auxiliary Input Type 2		X	X	X	X	11783-6 - J.4.5	L3 - L6
33	Auxiliary Control DesignatorType 2 Object Pointer		X	X	X	X	11783-6 - J.4.7	L3 - L6
34	Windows Mask			X	X	X	11783-6 - B.19	L4 - L6
35	Key Group			X	X	X	11783-6 - B.20	L4 - L6
36	Graphics Context Object			X	X	X	11783-6 - B.18	L3 - L6
37	Output List			X	X	X	11783-6 - B.9.4	L4 - L6
38	Extended Input Attributes			X	X	X	11783-6 - B.14.6	L4 - L6
39	Colour Map			X	X	X	11783-6 - B.17	L5 - L6
40	Object Label Reference List			X	X	X	11783-6 - B.21	L5 - L6
41	External Object Definition				X	X	11783-6 - B.22	L5 - L6
42	External reference NAME				X	X	11783-6 - B.23	L5 - L6
43	External Object Pointer				X	X	11783-6 - B.24	L5 - L6
44	Animation				X	X	11783-6 - B.25	L5 - L6
45	Colour Palette				X	X	11783-6 - B.26	L6
46	Graphic Data (PNG)					X	11783-6 - B.27	L6
47	Working Set Special Controls					X	11783-6 - B.29	L6
48	Scaled Graphic					X	11783-6 - B.28	L6

8. ISOBUS Objekte

schauen Sie sich diesen Link genau an.

8.1 Programming

Programming_And_Libraries

künftig: ISOBUS-Objekte-Versionen

ISOBUS Wiki von Tobias Tenberg:

- <https://isobus-studio.com/en/isobus-wiki>
- ISOBUS Wiki - Colours
- ISOBUS Wiki - Objectpool Objects Database

Wie Sie sehen gibt es verschiedenen Objekte, aber nicht alle werden in jeder ISOBUS Version unterstützt.

Add Project

Please choose a Platform:

- VT Level 2
- VT Level 3
- VT Level 4
- VT Level 5
- VT Level 6**

Selected Platform: VT Level 6

Display: Device 200x200 Display Name: Display1

Project Name: MyProject

Location: C:\Users\mail\Documents\ISO-DesignerProjects\

Add to current Workspace Create new Workspace MyWorkspace1

Create folder for Workspace

Create Working Set and active Mask

Add Existing... OK Cancel

Wenn Sie mit ISO-Designer ein neues Projekt anlegen würden dann fragt er Sie nach dem „VT Level“
Wenn Sie in der AEF Datenbank nach Geräten suchen werden Sie sowas hier sehen:

Options	Description
UT 2.0 CLIENT	
Field	Value
Functionality	UT
Functionality Type	Client
Generation	2.0

Was ist also der Unterschied zwischen UT und VT ?
 oder ist es etwa dasselbe ?
 ist VT 2.0 dasselbe wie UT 2.0 ?
 hier kommt die Auflösung:

8.2 VT Levels

	ISO 11783-06	Jetter ISO-Designer	UT laut AEF Datenbank	AUX laut AEF Datenbank
1	Unbenannt	< nicht wählbar >	< nicht zertifizierbar >	kein AUX
2	VT Version 2	VT Level 2	UT 1.0	AUX-O 1.0
3	VT Version 3	VT Level 3	UT 2.0	AUX-N 1.0
4	VT Version 4	VT Level 4	UT 2.0	AUX-N 1.0
5	VT Version 5	VT Level 5	UT 2.0	AUX-N 1.0
6	VT Version 6	VT Level 6	UT 2.0	AUX-N 1.0
7	VT Version 7	< nicht wählbar >	UT 3.0	AUX-N 1.0

man begann also mit der ersten Version der Norm, und veröffentlichte diese.

in der 2. Version war von einem VT Version 2 die Rede ... und so fort. Schauen Sie das bitte in der Norm nach.

die AEF hält sich nicht an dieses Schema, sondern führt für Marketingzwecke eigene Versionen ein. an UT 3.0 wird gerade gearbeitet, und wird wohl VT 4 und 5 einschließen.

AUX ist in der ISOBUS Norm nur zum jeweiligen VT Standard erwähnt, die AEF hat daraus AUX-O und AUX-N gemacht.

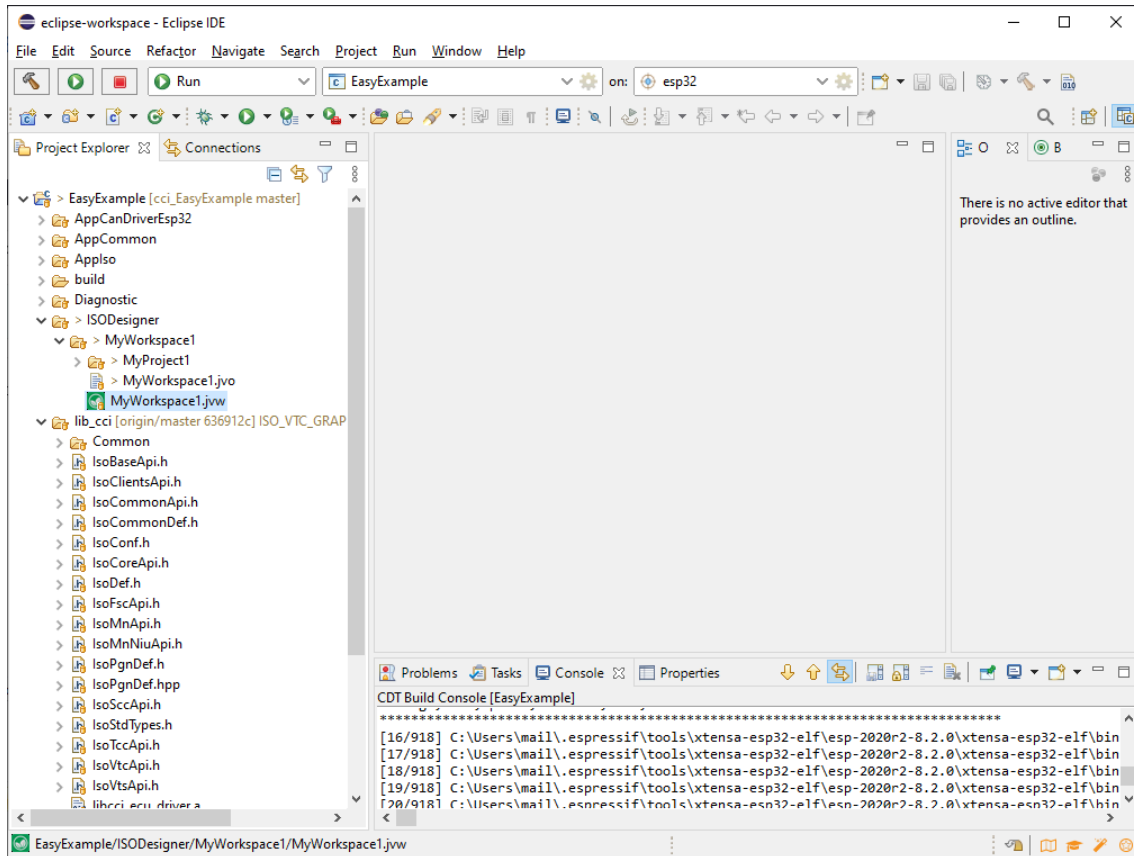
9. ISOBUS Wiki Links

Hier ist eine Sammlung von Links zum [ISOBUS Wiki](#) von Tobias Tenberg, die detaillierte Informationen zu verschiedenen ISOBUS-Objekten bieten.

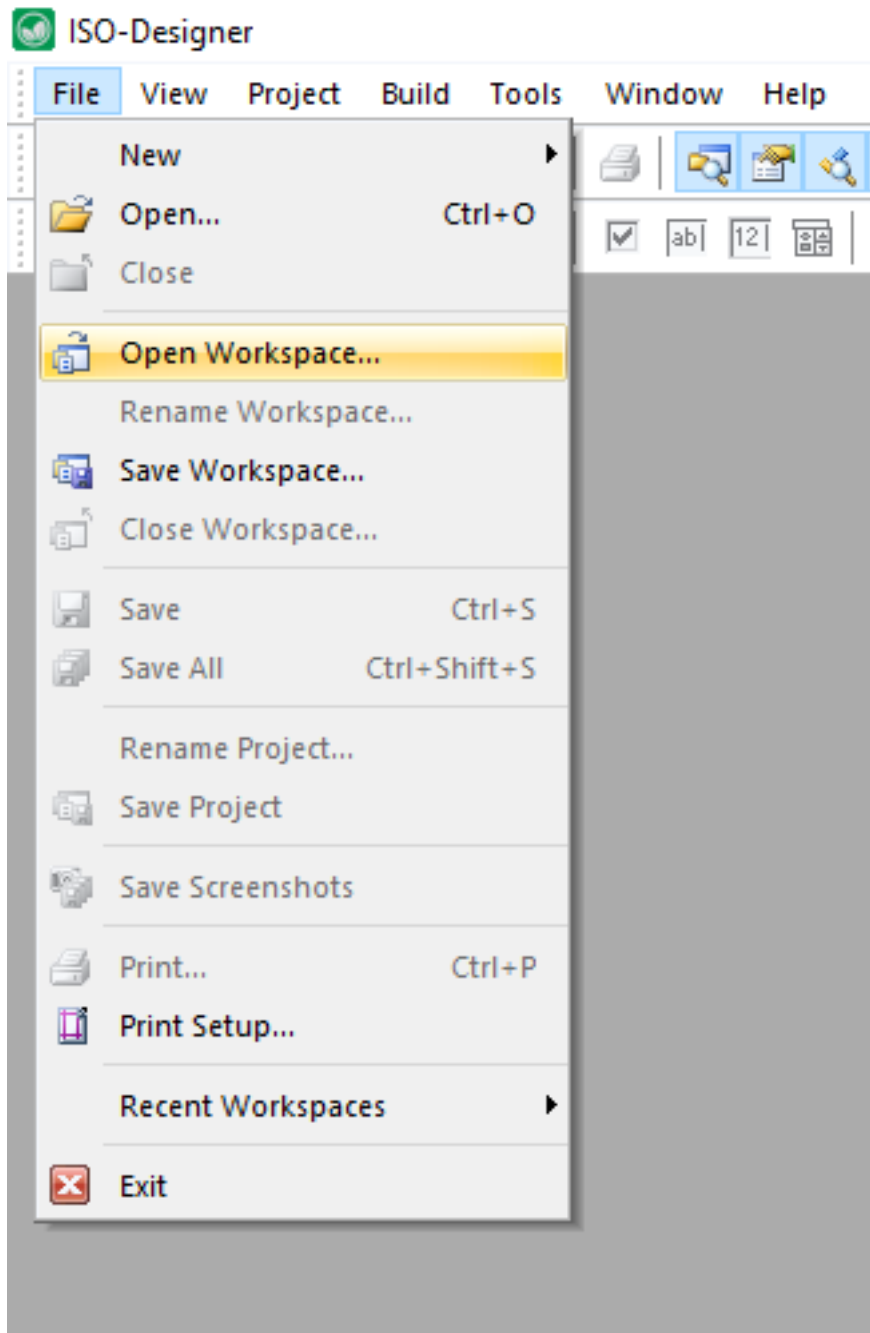
- [Alarm Mask](#)
- [Arched Bar Graph](#)
- [Boolean](#)
- [Button](#)
- [Container](#)
- [Data Mask](#)
- [Ellipse](#)
- [Fill Attribute](#)
- [Font Attribute](#)
- [Line \(Output\)](#)
- [Line Attribute](#)
- [List \(Input\)](#)
- [List \(Output\)](#)
- [Meter](#)
- [Number \(Input\)](#)
- [Number \(Output\)](#)
- [Number Variable](#)
- [Object Pointer](#)
- [Picture Graphic object](#)
- [Polygon](#)
- [Rectangle](#)
- [Softkey](#)
- [Softkey Mask](#)
- [String \(Input\)](#)
- [String \(Output\)](#)
- [String Variable](#)
- [Window Mask](#)
- [Working Set](#)

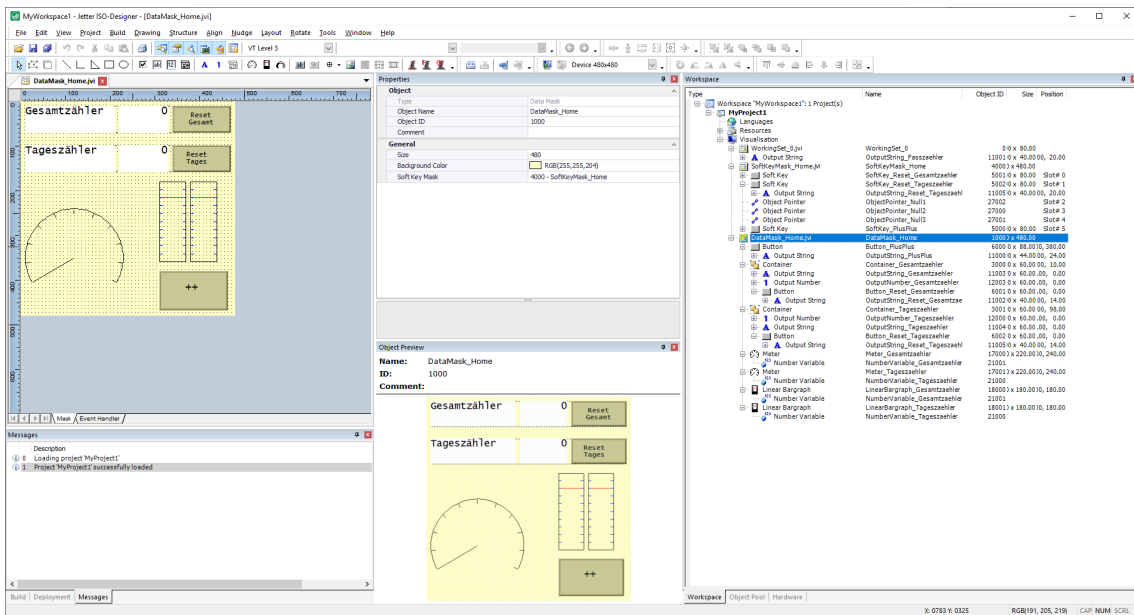
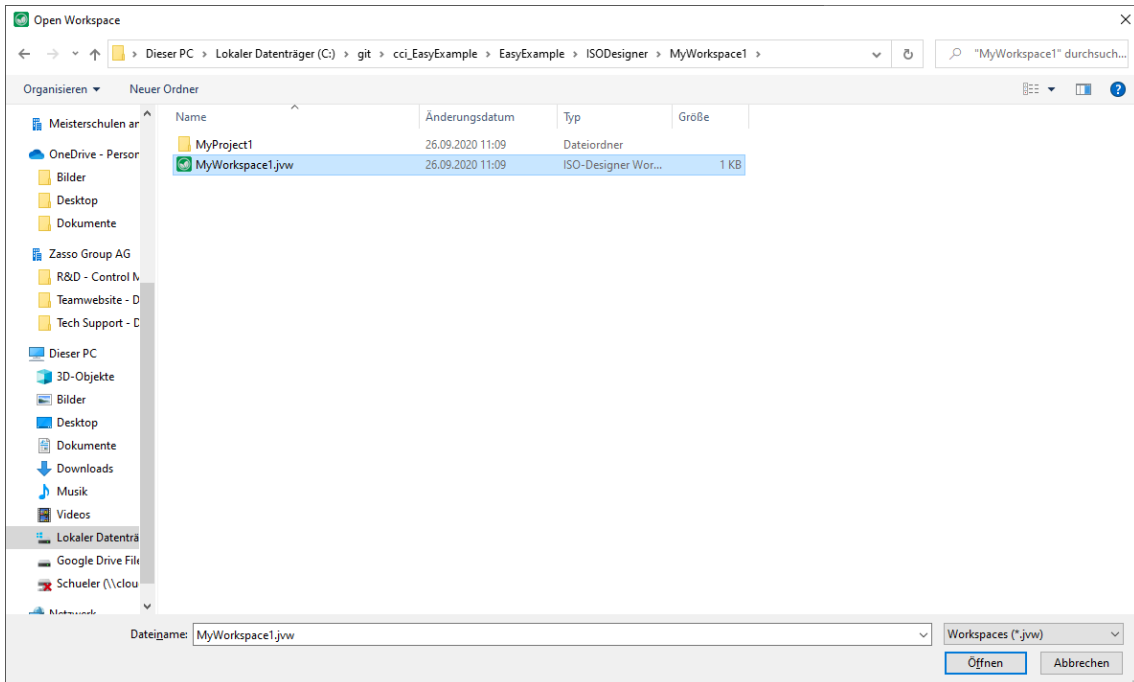
10. Jetter ISO Designer Workspace öffnen

Doppelklick hier:



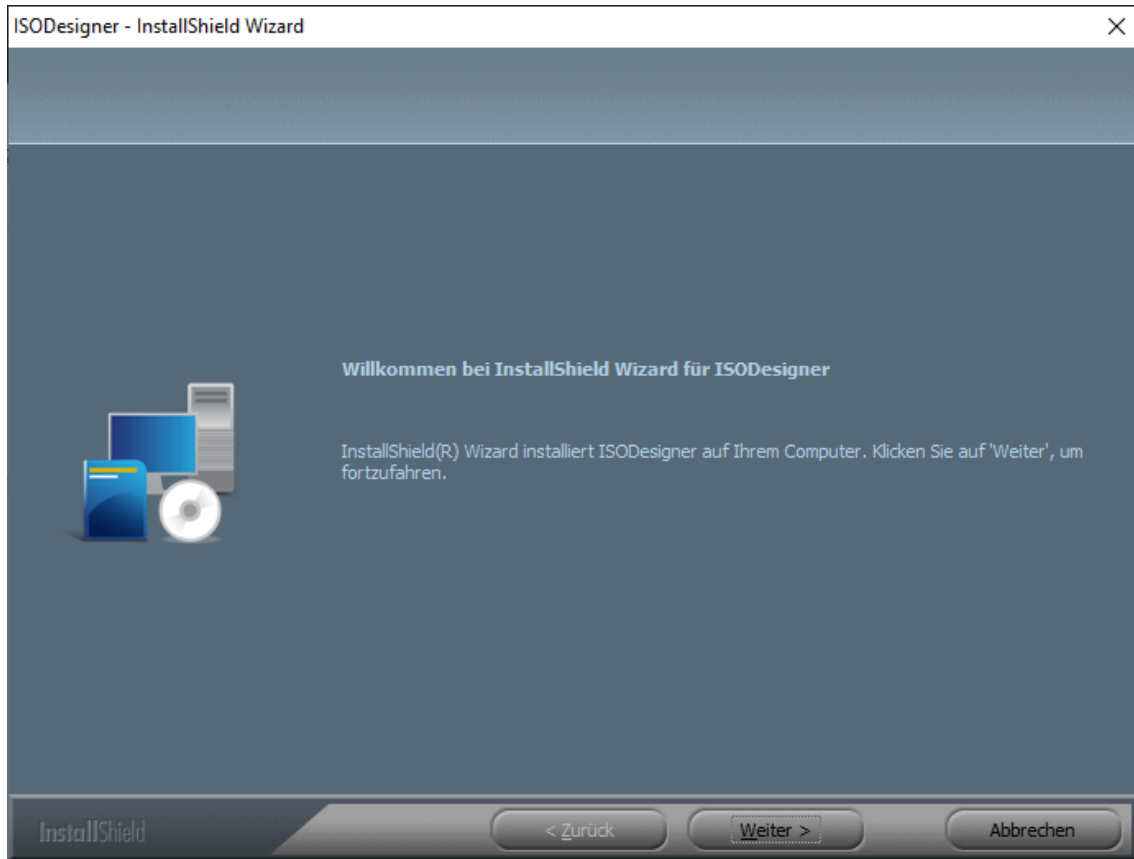
ALTERNATIV:

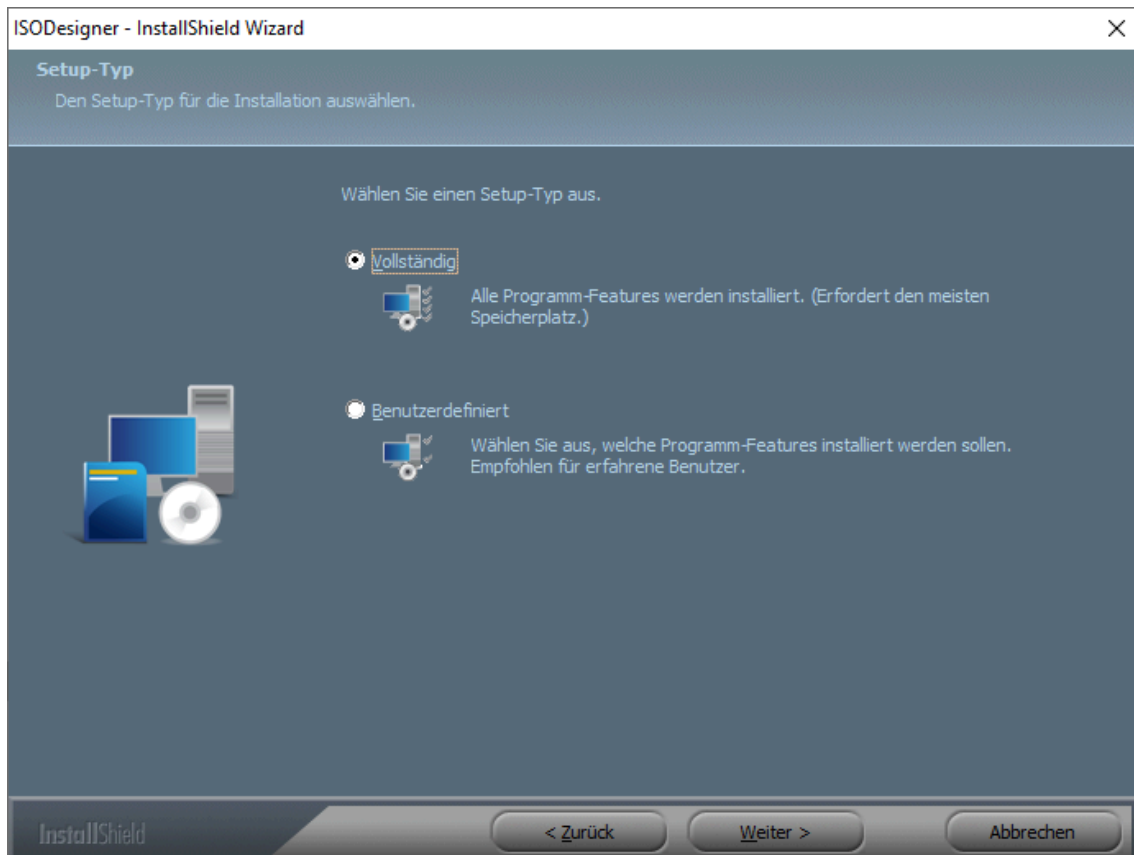
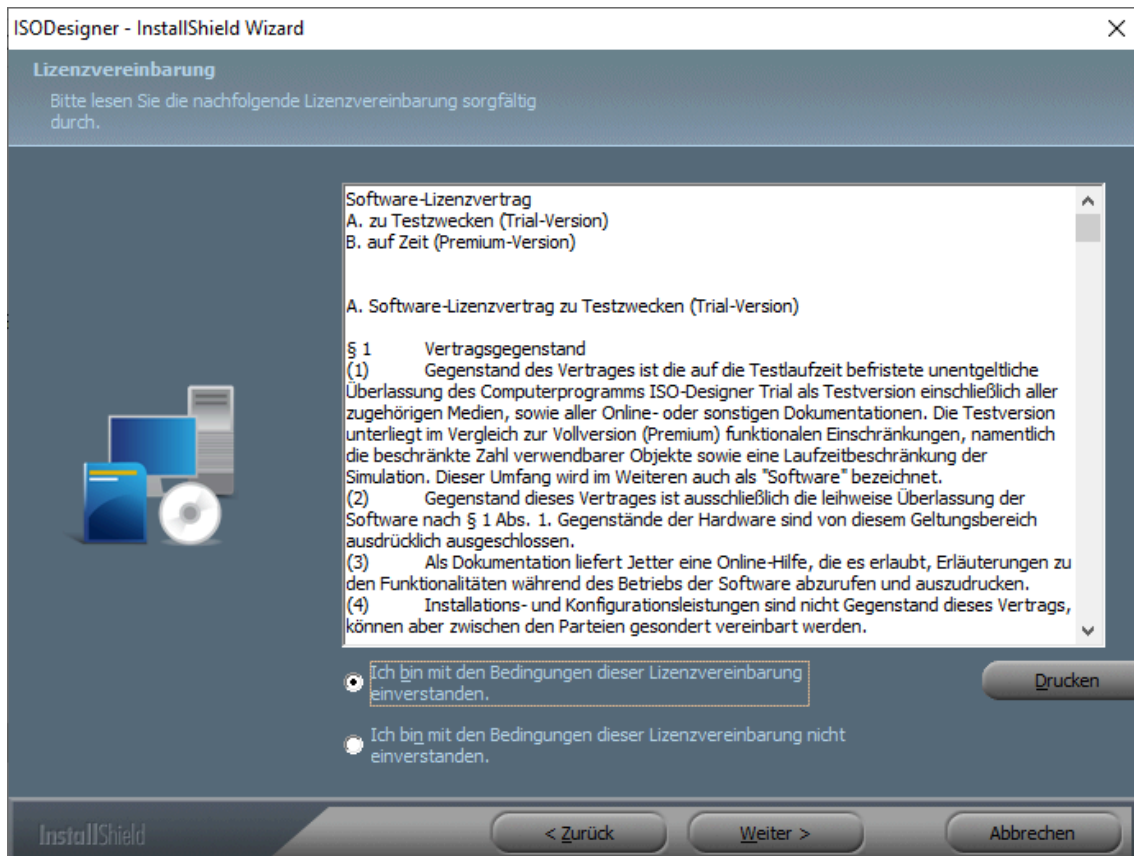


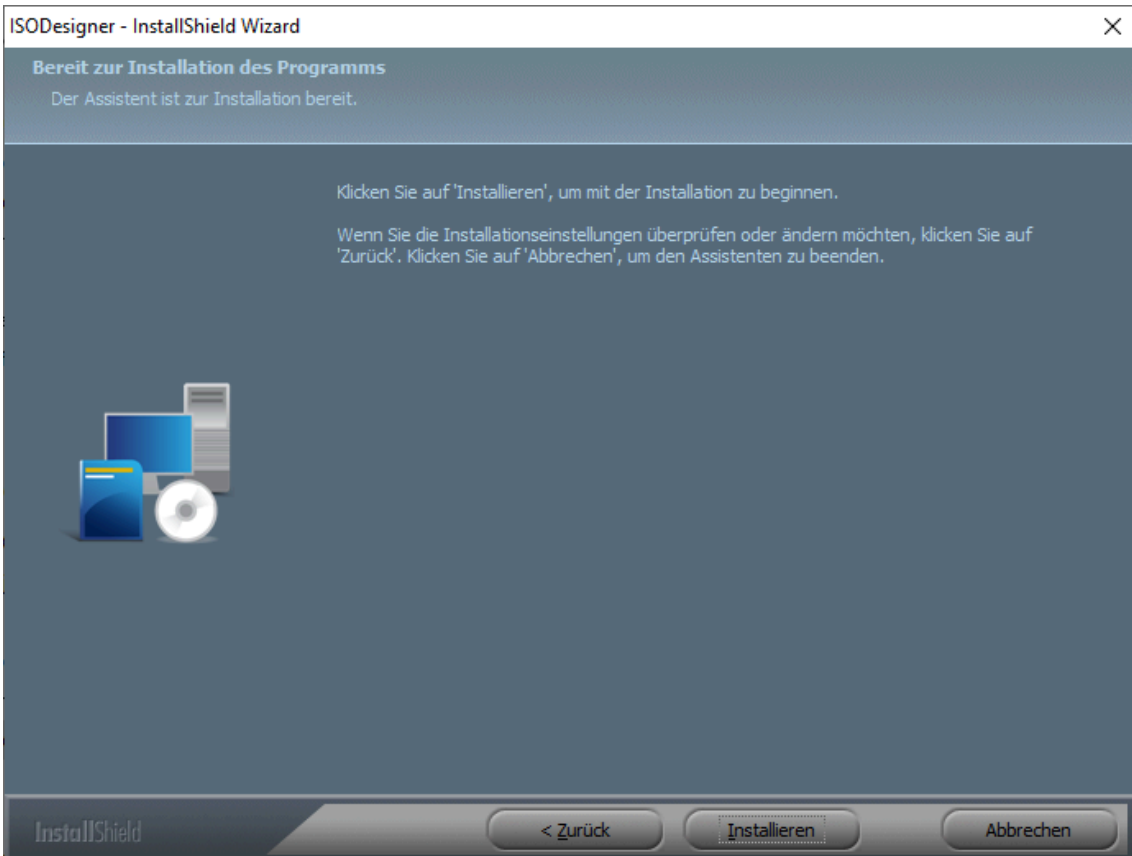
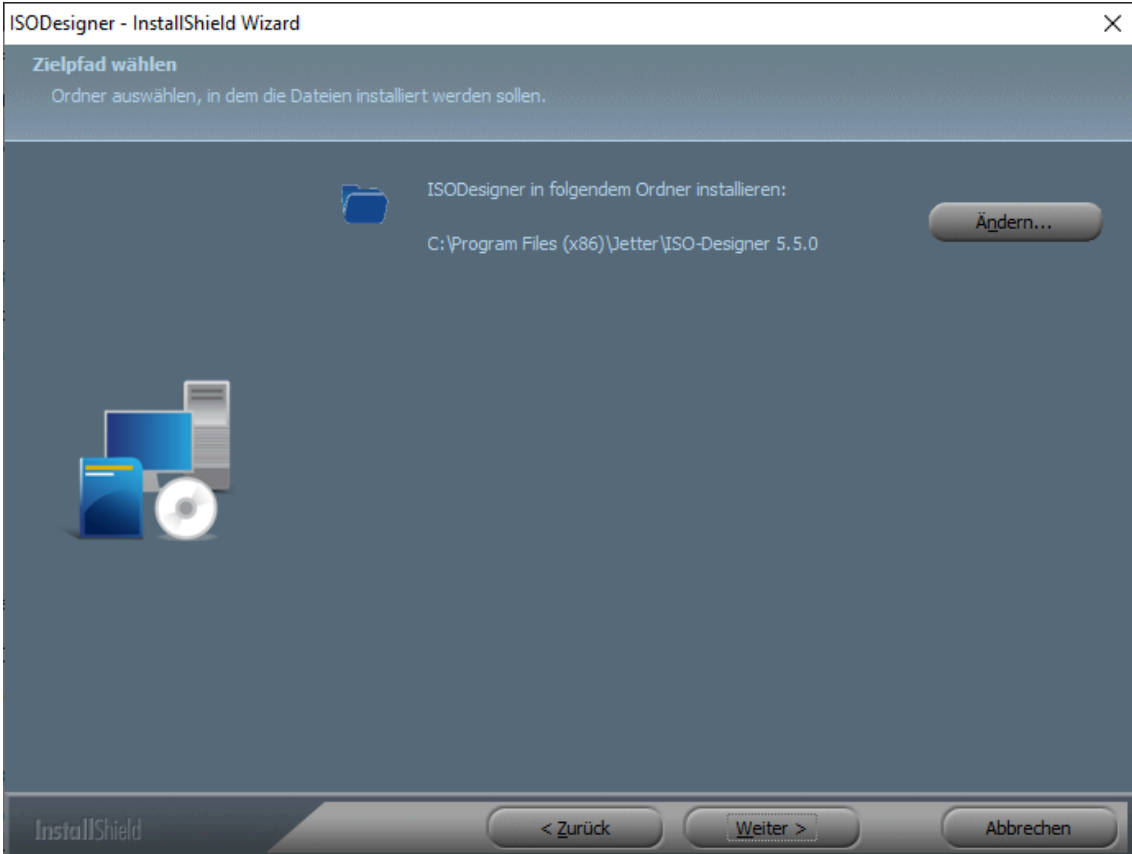


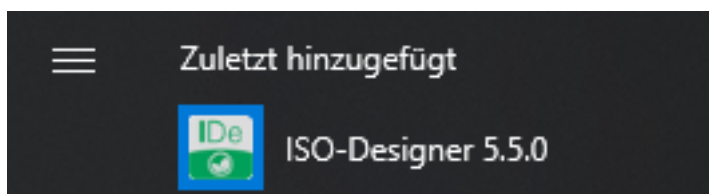
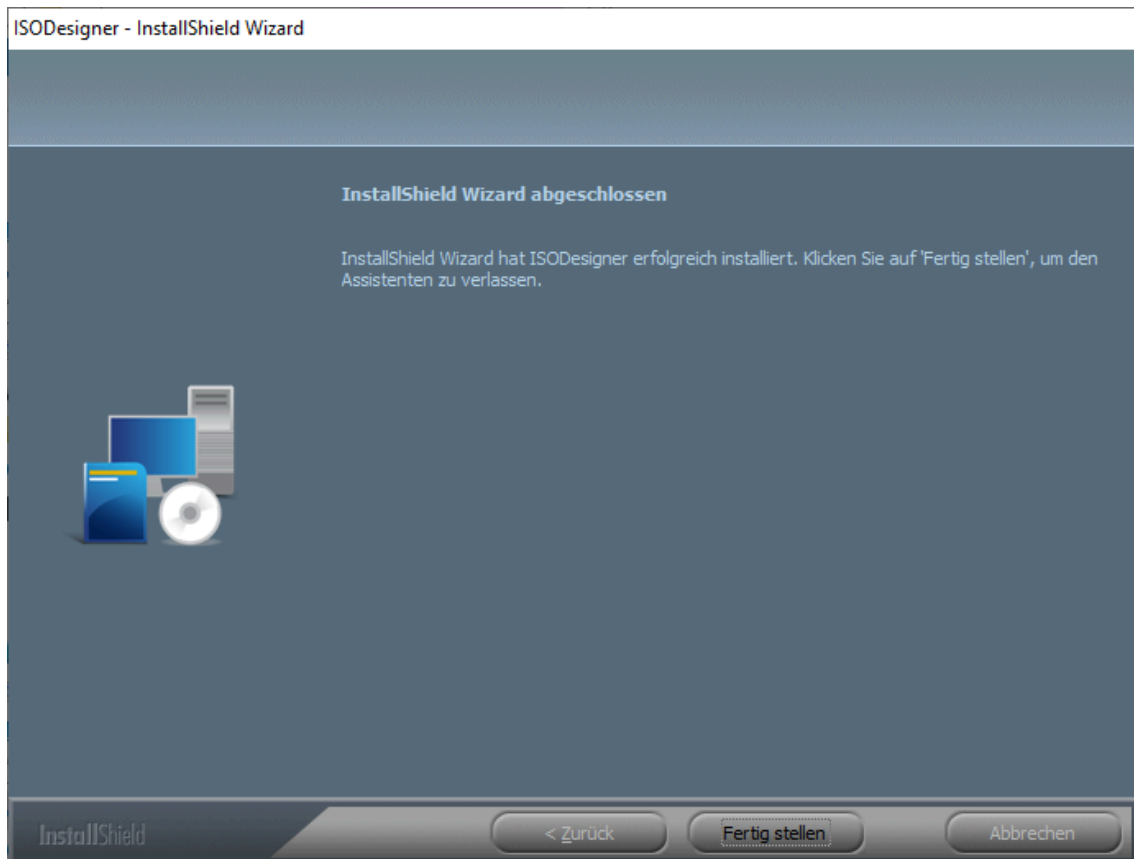
11. Jetter ISO-Designer

M:\Landmaschinenmechanik\Unterricht\SL\Jetter\ISODesignerSetup551.exe

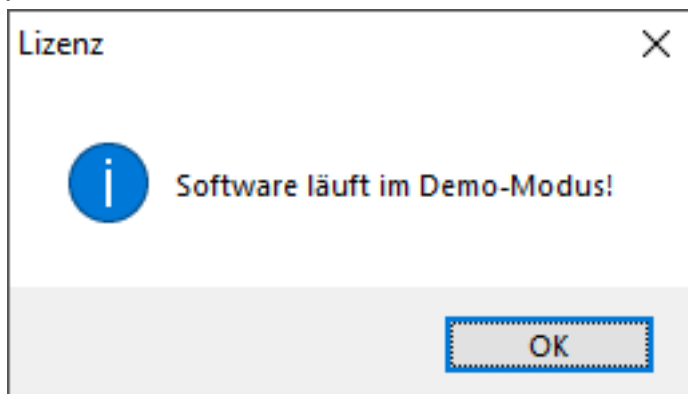








Ja



OK

12. Leere Objekte (Minimum Child Objects)

In der ISO 11783-6 Norm gibt es für Container-Objekte (Objekte, die andere Objekte enthalten) eine wichtige Unterscheidung: **Darf das Objekt leer sein oder nicht?**

Diese Regel wird durch den zulässigen Wertebereich des Attributs „**Number of objects to follow**“ definiert.

12.1 Objekte, die NICHT leer sein dürfen (Minimum 1 Child)

Diese Objekte müssen zwingend mindestens ein Kind-Objekt enthalten, da sie ohne Inhalt ihre Funktion im System nicht erfüllen können (meist weil sie als **Designator** / Icon für den Benutzer sichtbar sein müssen).

Objekt ID	Name	Grund	Norm-Referenz
0	Working Set	Muss im VT-Menü zur Auswahl angezeigt werden. Ohne grafischen Designator wäre das Gerät für den Bediener unsichtbar und nicht auswählbar.	Table B.2
29	Auxiliary Function Type 1	Muss auf dem Zuweisungs-Bildschirm (Mapping Screen) dargestellt werden, damit der Benutzer die Funktion (z.B. „Heben“) einem Schalter zuordnen kann.	Table J.1
30	Auxiliary Input Type 1	Muss auf dem Mapping Screen dargestellt werden, um den physischen Eingang (z.B. „Joystick Taste A“) zu repräsentieren.	Table J.3
31	Auxiliary Function Type 2	Siehe ID 29.	Table J.2
32	Auxiliary Input Type 2	Siehe ID 30.	Table J.4

Fehlerfall: Wenn ein solches Objekt mit 0 Kind-Objekten im Objektpool definiert wird, ist der Pool **ungültig** und wird vom VT (oder einem Prüftool) abgelehnt.

12.2 Objekte, die leer sein dürfen (Minimum 0 Children)

Für die meisten anderen Container ist ein Inhalt von 0 erlaubt. Dies kann durchaus sinnvoll sein.

Objekt ID	Name	Verhalten bei 0 Kindern
1	Data Mask	Zeigt nur die Hintergrundfarbe an.
2	Alarm Mask	Zeigt nur die Hintergrundfarbe an (Warnung ohne Inhalt).
3	Container	Ist effektiv unsichtbar und beansprucht keinen Platz (sofern nicht durch Width/Height eine Fläche definiert ist, die Hintergrundfarbe hat).
4	Soft Key Mask	Blendet alle Softkeys aus oder lässt sie unbeschriftet.
5	Key	Zeigt eine leere Taste in der definierten Hintergrundfarbe.
6	Button	Zeigt eine leere Schaltfläche in der definierten Hintergrundfarbe.

35	Key Group	Technisch erlaubt, aber funktional sinnlos (eine leere Gruppe).
44	Animation	Eine Animation ohne Frames zeigt nichts an.

12.2.1 Besonderheit: Input List (ID 10)

Auch eine Input List darf theoretisch leer sein (`Number of list items = 0`). In diesem Fall ist die Liste zwar vorhanden, enthält aber keine auswählbaren Optionen.

13. Objekthierarchie (Parent-Child Beziehungen)

Diese Seite basiert auf **Tabelle A.2 – Allowed hierarchical relationships of objects** der ISO 11783-6:2018 (Annex A). Sie definiert, welche Objekte (Child) in welchen anderen Objekten (Parent) enthalten sein dürfen.

Die angegebene Zahl steht für die minimale **VT-Version**, ab der diese Beziehung unterstützt wird.

13.1 Parent: Working Set object (ID 0)

Das Working Set Objekt ist der Wurzel-Container einer Objekt-Pool-Hierarchie.

Child Object	Min. VT Version
Container object (ID 3)	4
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output List object (ID 37)	4
Output Line object (ID 13)	2
Output Rectangle object (ID 14)	2
Output Ellipse object (ID 15)	2
Output Polygon object (ID 16)	2
Output Meter object (ID 17)	4
Output Linear Bar Graph object (ID 18)	4
Output Arched Bar Graph object (ID 19)	4
Graphics Context object (ID 36)	4
Picture Graphic object (ID 20)	2
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	4

13.2 Parent: Data Mask object (ID 1)

Die Datenmaske ist der Hauptanzeigebereich für die Bedienung.

Child Object	Min. VT Version
Working Set object (ID 0)	3
Container object (ID 3)	2
Button object (ID 6)	2
Input Boolean object (ID 7)	2
Input String object (ID 8)	2

Input Number object (ID 9)	2
Input List object (ID 10)	2
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output List object (ID 37)	4
Output Line object (ID 13)	2
Output Rectangle object (ID 14)	2
Output Ellipse object (ID 15)	2
Output Polygon object (ID 16)	2
Output Meter object (ID 17)	2
Output Linear Bar Graph object (ID 18)	2
Output Arched Bar Graph object (ID 19)	2
Graphics Context object (ID 36)	4
Animation object (ID 44)	5
Picture Graphic object (ID 20)	2
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	2
External Object Pointer object (ID 43)	5
Auxiliary Function Type 2 object (ID 31)	3
Auxiliary Input Type 2 object (ID 32)	3
Auxiliary Control Designator Type 2 Object Pointer (ID 33)	3

13.3 Parent: Alarm Mask object (ID 2)

Alarmmasken dienen zur Anzeige von Warnungen.

Child Object	Min. VT Version
Working Set object (ID 0)	3
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output List object (ID 37)	4
Output Line object (ID 13)	2
Output Rectangle object (ID 14)	2
Output Ellipse object (ID 15)	2
Output Polygon object (ID 16)	2
Output Meter object (ID 17)	2
Output Linear Bar Graph object (ID 18)	2
Output Arched Bar Graph object (ID 19)	2
Graphics Context object (ID 36)	4

Animation object (ID 44)	5
Picture Graphic object (ID 20)	2
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	2
External Object Pointer object (ID 43)	5
Auxiliary Function Type 2 object (ID 31)	3
Auxiliary Input Type 2 object (ID 32)	3
Auxiliary Control Designator Type 2 Object Pointer (ID 33)	3

13.4 Parent: Container object (ID 3)

Container dienen zur Gruppierung von Objekten.

Child Object	Min. VT Version
Working Set object (ID 0)	3
Container object (ID 3)	2
Button object (ID 6)	2
Input Boolean object (ID 7)	2
Input String object (ID 8)	2
Input Number object (ID 9)	2
Input List object (ID 10)	2
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output List object (ID 37)	4
Output Line object (ID 13)	2
Output Rectangle object (ID 14)	2
Output Ellipse object (ID 15)	2
Output Polygon object (ID 16)	2
Output Meter object (ID 17)	2
Output Linear Bar Graph object (ID 18)	2
Output Arched Bar Graph object (ID 19)	2
Graphics Context object (ID 36)	4
Animation object (ID 44)	5
Picture Graphic object (ID 20)	2
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	2
External Object Pointer object (ID 43)	5
Auxiliary Function Type 2 object (ID 31)	3
Auxiliary Input Type 2 object (ID 32)	3

Auxiliary Control Designator Type 2 Object Pointer (ID 33)	3
--	---

13.5 Parent: Window Mask object (ID 34)

Fenstermasken werden in User-Layouts verwendet.

Child Object	Min. VT Version
Working Set object (ID 0)	4
Container object (ID 3)	4
Button object (ID 6)	4
Input Boolean object (ID 7)	4
Input String object (ID 8)	4
Input Number object (ID 9)	4
Input List object (ID 10)	4
Output String object (ID 11)	4
Output Number object (ID 12)	4
Output List object (ID 37)	4
Output Line object (ID 13)	4
Output Rectangle object (ID 14)	4
Output Ellipse object (ID 15)	4
Output Polygon object (ID 16)	4
Output Meter object (ID 17)	4
Output Linear Bar Graph object (ID 18)	4
Output Arched Bar Graph object (ID 19)	4
Graphics Context object (ID 36)	4
Animation object (ID 44)	5
Picture Graphic object (ID 20)	4
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	4
External Object Pointer object (ID 43)	5
Auxiliary Function Type 2 object (ID 31)	6
Auxiliary Input Type 2 object (ID 32)	6
Auxiliary Control Designator Type 2 Object Pointer (ID 33)	4

13.6 Parent: Soft Key Mask object (ID 4)

Definiert die Belegung der Softkeys.

Child Object	Min. VT Version
Key object (ID 5)	2
Object Pointer object (ID 27)	2

External Object Pointer object (ID 43)	5
--	---

13.7 Parent: Key object (ID 5)

Inhalt einer Taste.

Child Object	Min. VT Version
Working Set object (ID 0)	4
Container object (ID 3)	2
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output List object (ID 37)	4
Output Line object (ID 13)	2
Output Rectangle object (ID 14)	2
Output Ellipse object (ID 15)	2
Output Polygon object (ID 16)	2
Output Meter object (ID 17)	4
Output Linear Bar Graph object (ID 18)	4
Output Arched Bar Graph object (ID 19)	4
Graphics Context object (ID 36)	4
Animation object (ID 44)	5
Picture Graphic object (ID 20)	2
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	2
External Object Pointer object (ID 43)	5

13.8 Parent: Button object (ID 6)

Inhalt einer Schaltfläche.

Child Object	Min. VT Version
Working Set object (ID 0)	4
Container object (ID 3)	2
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output List object (ID 37)	4
Output Line object (ID 13)	2
Output Rectangle object (ID 14)	2
Output Ellipse object (ID 15)	2
Output Polygon object (ID 16)	2
Output Meter object (ID 17)	4

Output Linear Bar Graph object (ID 18)	4
Output Arched Bar Graph object (ID 19)	4
Graphics Context object (ID 36)	4
Animation object (ID 44)	5
Picture Graphic object (ID 20)	2
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	2
External Object Pointer object (ID 43)	5

13.9 Parent: Key Group object (ID 35)

Gruppierung von Softkeys.

Child Object	Min. VT Version
Key object (ID 5)	4
Object Pointer object (ID 27)	4
External Object Pointer object (ID 43)	5

13.10 Parent: Input List object (ID 10)

Optionen in einer Auswahlliste.

Child Object	Min. VT Version
Working Set object (ID 0)	4
Container object (ID 3)	4
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output List object (ID 37)	4
Output Line object (ID 13)	4
Output Rectangle object (ID 14)	4
Output Ellipse object (ID 15)	4
Output Polygon object (ID 16)	4
Output Meter object (ID 17)	4
Output Linear Bar Graph object (ID 18)	4
Output Arched Bar Graph object (ID 19)	4
Graphics Context object (ID 36)	4
Picture Graphic object (ID 20)	2
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	4
External Object Pointer object (ID 43)	5

13.11 Parent: Output List object (ID 37)

Optionen in einer Ausgabeliste.

Child Object	Min. VT Version
Working Set object (ID 0)	4
Container object (ID 3)	4
Button object (ID 6)	4
Input Boolean object (ID 7)	4
Input String object (ID 8)	4
Input Number object (ID 9)	4
Input List object (ID 10)	4
Output String object (ID 11)	4
Output Number object (ID 12)	4
Output List object (ID 37)	4
Output Line object (ID 13)	4
Output Rectangle object (ID 14)	4
Output Ellipse object (ID 15)	4
Output Polygon object (ID 16)	4
Output Meter object (ID 17)	4
Output Linear Bar Graph object (ID 18)	4
Output Arched Bar Graph object (ID 19)	4
Graphics Context object (ID 36)	4
Animation object (ID 44)	5
Picture Graphic object (ID 20)	4
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	4
External Object Pointer object (ID 43)	5
Auxiliary Function Type 2 object (ID 31)	6
Auxiliary Input Type 2 object (ID 32)	6
Auxiliary Control Designator Type 2 Object Pointer (ID 33)	4

13.12 Parent: Auxiliary Function Type 1 object (ID 29)

Designator für Hilfsfunktionen (veraltet).

Child Object	Min. VT Version
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output Line object (ID 13)	2
Output Rectangle object (ID 14)	2

Output Ellipse object (ID 15)	2
Output Polygon object (ID 16)	2
Picture Graphic object (ID 20)	2

13.13 Parent: Auxiliary Input Type 1 object (ID 30)

Designator für Hilfseingänge (veraltet).

Child Object	Min. VT Version
Output String object (ID 11)	2
Output Number object (ID 12)	2
Output Line object (ID 13)	2
Output Rectangle object (ID 14)	2
Output Ellipse object (ID 15)	2
Output Polygon object (ID 16)	2
Picture Graphic object (ID 20)	2

13.14 Parent: Auxiliary Function Type 2 object (ID 31)

Designator für Hilfsfunktionen.

Child Object	Min. VT Version
Container object (ID 3)	3
Output String object (ID 11)	3
Output Number object (ID 12)	3
Output List object (ID 37)	4
Output Line object (ID 13)	3
Output Rectangle object (ID 14)	3
Output Ellipse object (ID 15)	3
Output Polygon object (ID 16)	3
Output Meter object (ID 17)	3
Output Linear Bar Graph object (ID 18)	3
Output Arched Bar Graph object (ID 19)	3
Graphics Context object (ID 36)	4
Picture Graphic object (ID 20)	3
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	3

13.15 Parent: Auxiliary Input Type 2 object (ID 32)

Designator für Hilfseingänge.

Child Object	Min. VT Version
--------------	-----------------

Container object (ID 3)	3
Output String object (ID 11)	3
Output Number object (ID 12)	3
Output List object (ID 37)	4
Output Line object (ID 13)	3
Output Rectangle object (ID 14)	3
Output Ellipse object (ID 15)	3
Output Polygon object (ID 16)	3
Output Meter object (ID 17)	3
Output Linear Bar Graph object (ID 18)	3
Output Arched Bar Graph object (ID 19)	3
Graphics Context object (ID 36)	4
Picture Graphic object (ID 20)	3
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	3

13.16 Parent: Object Label graphic representation

Objekte, die als Icon/Grafik in einem Label verwendet werden dürfen.

Child Object	Min. VT Version
Container object (ID 3)	4
Output String object (ID 11)	4
Output Number object (ID 12)	4
Output List object (ID 37)	4
Output Line object (ID 13)	4
Output Rectangle object (ID 14)	4
Output Ellipse object (ID 15)	4
Output Polygon object (ID 16)	4
Output Meter object (ID 17)	4
Output Linear Bar Graph object (ID 18)	4
Output Arched Bar Graph object (ID 19)	4
Graphics Context object (ID 36)	4
Picture Graphic object (ID 20)	4
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	4

13.17 Parent: Object Label Reference List object (ID 40)

Definiert Zuweisungen von Labels zu Objekten.

Child Object	Min. VT Version
Working Set object (ID 0)	4
Data Mask object (ID 1)	4
Alarm Mask object (ID 2)	4
Container object (ID 3)	4
Window Mask object (ID 34)	4
Soft Key Mask object (ID 4)	4
Key object (ID 5)	4
Button object (ID 6)	4
Input Boolean object (ID 7)	4
Input String object (ID 8)	4
Input Number object (ID 9)	4
Input List object (ID 10)	4
Output String object (ID 11)	4
Output Number object (ID 12)	4
Output List object (ID 37)	4
Output Line object (ID 13)	4
Output Rectangle object (ID 14)	4
Output Ellipse object (ID 15)	4
Output Polygon object (ID 16)	4
Output Meter object (ID 17)	4
Output Linear Bar Graph object (ID 18)	4
Output Arched Bar Graph object (ID 19)	4
Graphics Context object (ID 36)	4
Animation object (ID 44)	5
Picture Graphic object (ID 20)	4
Scaled Graphic object (ID 48)	6
Number Variable object (ID 21)	5
String Variable object (ID 22)	5
Font Attributes object (ID 23)	5
Line Attributes object (ID 24)	5
Fill Attributes object (ID 25)	5
Input Attributes object (ID 26)	5
Extended Input Attributes object (ID 41)	5
Colour Map object (ID 42)	5
Object Label Reference List object (ID 40)	5

Object Pointer object (ID 27)	5
External Object Definition object (ID 45)	5
External Reference NAME object (ID 46)	5
External Object Pointer object (ID 43)	5
Macro object (ID 47)	5
Auxiliary Function Type 1 object (ID 29)	5
Auxiliary Input Type 1 object (ID 30)	5
Auxiliary Function Type 2 object (ID 31)	4
Auxiliary Input Type 2 object (ID 32)	4
Auxiliary Control Designator Type 2 Object Pointer (ID 33)	5
Colour Palette object (ID 49)	6
Graphic Data object (ID 50)	6
Working Set Special Controls object (ID 51)	6

13.18 Parent: Animation object (ID 44)

Sequenz von Objekten.

Child Object	Min. VT Version
Container object (ID 3)	5
Output String object (ID 11)	5
Output Number object (ID 12)	5
Output List object (ID 37)	5
Output Line object (ID 13)	5
Output Rectangle object (ID 14)	5
Output Ellipse object (ID 15)	5
Output Polygon object (ID 16)	5
Output Meter object (ID 17)	5
Output Linear Bar Graph object (ID 18)	5
Output Arched Bar Graph object (ID 19)	5
Graphics Context object (ID 36)	5
Animation object (ID 44)	5
Picture Graphic object (ID 20)	5
Scaled Graphic object (ID 48)	6
Object Pointer object (ID 27)	5

14. Ungültige Objektpools (Invalid Object Pools)

Damit ein ISOBUS-Objektpool (IOP) vom Virtual Terminal (VT) akzeptiert und geladen wird, muss er strikten strukturellen Regeln folgen. Verletzt der Pool eine dieser Regeln, sendet das VT am Ende der Übertragung (End of Object Pool) einen Fehlercode und verwirft meist den gesamten Pool.

Hier sind die wichtigsten Regeln, die jeder Programmierer kennen muss:

14.1 1. Strukturelle Integrität (Parsing-Fehler)

Dies sind die häufigsten Fehler, die dazu führen, dass der Parser des VTs „aus dem Tritt“ gerät.

- **Falsche Längenangaben:** Wenn ein Objekt sagt „Es folgen 5 Kind-Objekte“, aber im Byte-Stream folgen nur 4 (oder 6), liest das VT das nächste Objekt-Byte fälschlicherweise als Attribut des vorherigen. Der Pool ist dann „kaputt“.
- **Unbekannte Objekttypen:** Eine Objekt-Typ-ID, die in der Norm (und der unterstützten VT-Version) nicht definiert ist.
- **Fehlendes ID 0 Objekt:** Jede Arbeitsgruppe **muss** genau ein Objekt mit der **ID 0** (Working Set) enthalten. Fehlt dieses, weiß das VT nicht, wer der Besitzer des Pools ist.

14.2 2. Referenzierungs-Fehler (Dangling Pointers)

Das VT prüft beim Laden (oder spätestens beim Aktivieren), ob alle Verknüpfungen gültig sind.

- **Tote Verweise:** Ein Objekt (z. B. ein Container) verweist in seiner Child-Liste auf eine Objekt-ID (z. B. 100), aber im gesamten Pool gibt es kein Objekt mit der ID 100.
 - *Ausnahme:* Referenzen auf NULL (65535 bzw. FFFFh) sind erlaubt und erwünscht (z. B. leere Softkeys).
- **Falsche Objekttypen:** Ein Attribut verlangt einen spezifischen Typ, aber die referenzierte ID gehört zu einem anderen Typ.
 - *Beispiel:* Ein Input Number Objekt verweist bei Variable Reference auf ein String Variable Objekt. -> **Ungültig!**
 - *Beispiel:* Ein Output Rectangle verweist bei Line Attributes auf ein Fill Attributes Objekt. -> **Ungültig!**

14.3 3. Logische Fehler & Rekursion

- **Zirkuläre Referenzen:** Ein Container A enthält Container B, und Container B enthält wieder Container A. Dies führt zu einer Endlosschleife beim Zeichnen. Das VT erkennt dies und lehnt den Pool ab oder stürzt (im schlimmsten Fall) ab.
- **Mehrfache Definition:** Eine Objekt-ID (z. B. ID 50) darf im Pool nur ein einziges Mal definiert werden. Doppelte IDs sind verboten.

14.4 4. Spezifische Objekt-Regeln (Hard Constraints)

Einige Objekte haben sehr spezifische Regeln, die oft übersehen werden:

- **Designator-Pflicht:** Objekte wie Working Set, Auxiliary Function und Auxiliary Input **müssen** mindestens ein Kind-Objekt haben (siehe Seite [Leere Objekte](#)).
- **Input Lists:** Der Value einer Input List darf nicht größer sein als Anzahl der Listeneinträge - 1.
- **Soft Key Masks:** Dürfen nur Key Objekte (ID 5) oder Object Pointer (ID 27) enthalten, die wiederum auf Keys zeigen. Ein Button (ID 6) in einer Soft Key Mask ist ungültig.

- **Variablen-Größe:** Wenn ein String-Objekt auf eine String-Variablen verweist, sollte die definierte Länge im String-Objekt idealerweise zur Länge der Variablen passen (obwohl VTs hier oft tolerant sind und clippen/padden).

14.5 5. Ressourcen-Limits

Auch wenn der Pool syntaktisch korrekt ist, kann er abgelehnt werden, wenn er die Hardware-Limits des VTs sprengt:

- **Speicher voll:** Der Pool ist größer als der verfügbare Flash-Speicher des VTs (Memory Out of range).
- **Zu viele Objekte:** Einige ältere VTs haben Limits für die absolute Anzahl an Objekten (z. B. max. 65534, was durch das 16-Bit ID-Feld vorgegeben ist, aber reale Limits liegen oft niedriger).
- **Softkey-Limit:** Eine Softkey-Maske, die mehr Keys definiert, als das VT verwalten kann (obwohl Paging vorgeschrieben ist, gibt es oft harte Limits).

14.6 Tipps zur Fehlersuche

Wenn das VT beim Laden „Object Pool Error“ meldet:

1. Prüfen, ob **ID 0** existiert.
2. Prüfen, ob alle referenzierten IDs tatsächlich im Pool enthalten sind.
3. Prüfen, ob alle `Number of objects/macros/bytes` Felder exakt mit den folgenden Daten übereinstimmen.
4. Prüfen, ob Typ-Referenzen (Number Var vs. String Var) stimmen.

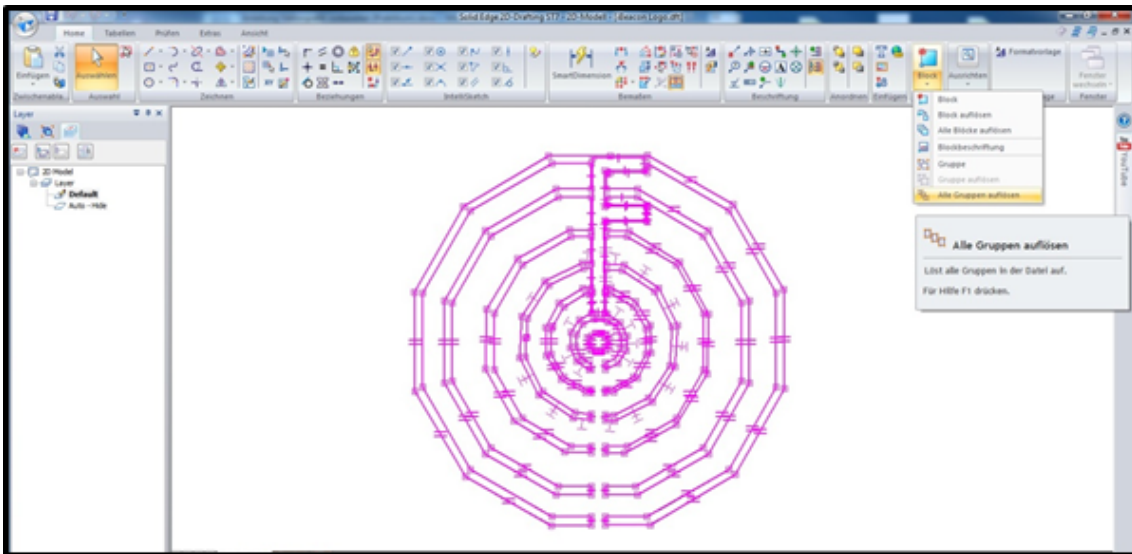
15. Vektorgrafiken erstellen

15.1 Coral Draw X5

Für den Fall, dass die Datei aus einer z.B. PDF Datei gewonnen werden soll, kann man sich mit Coral Draw X5 behelfen. Die Größe kann auch über Anordnen->Änderungen->Größe geändert werden. Achtung: Sehr ungenau und wird ein paar Versuche benötigen, um die gewünschte Größe zu bekommen. Falls es eine andere Möglichkeit geben sollte, bitte hier ausbessern (Lächeln)

15.2 Solid Edge

- Falls etwas nicht klappt, hilft es einfach sich die Arbeit zu machen und jeden Strich noch ein Mal neu zu setzen,
- um die klaren Bezüge zwischen den Linien sicherzustellen. Zeitraubend, aber dann sollte es gehen. (Zwinkern)
- Wenn man Symbole für die Displays zeichnet müssen diese die Maße 72x72 haben. Kurz ein Rechteck zeichnen und kontrollieren, dann ist man auf der sicheren Seite.
- Gesamtes Objekt markieren → Block → Alle Gruppen auflösen



Fertig skalierte Vektordatei speichern

Danach Speichern unter → Als übersetzt speichern (im .dxf Format)

Achtung ! Im Objekt darf es keine Rundungen geben. Alles muss in „Polygon über Mittelpunkt“ sein. Im Idealfall neun Seitenteile.

Ein Weiterarbeiten mit Rundungen würde sonst in Libre CAD bzw. ISO Designer zu Problemen führen !

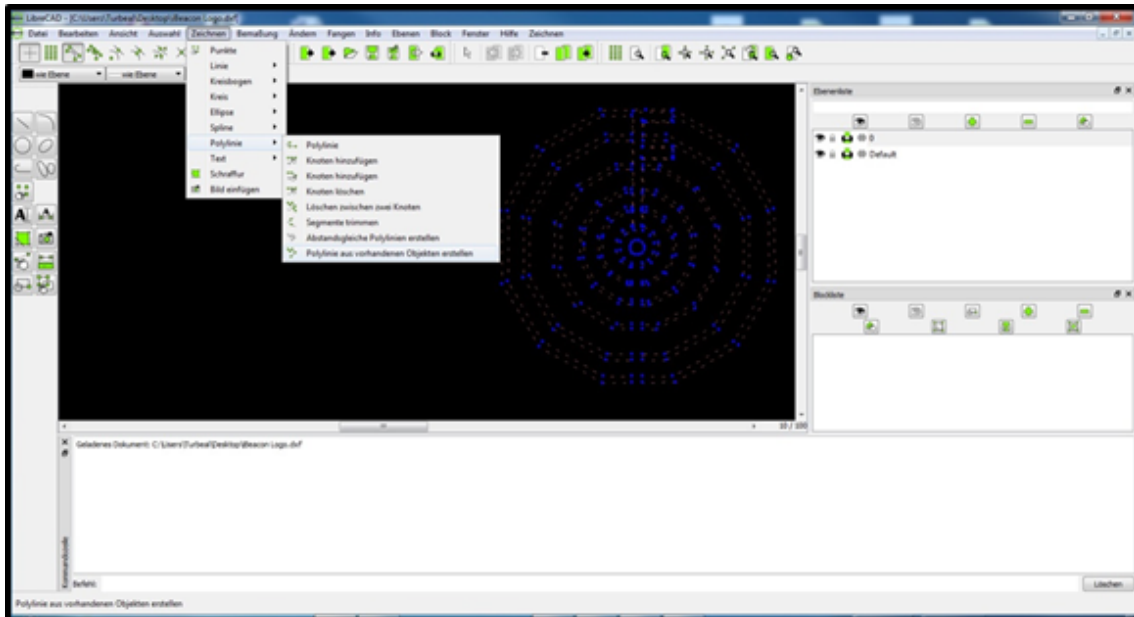
15.3 Libre CAD

Gespeicherte .dxf Datei in Libre CAD öffnen Alle Linien müssen verbunden werden: gesamtes Objekt markieren → Zeichnen (Werkzeuge) → Polylinie → Polylinie aus vorhandenen Objekten erstellen
Achtung: Es kann sein, dass nur Bruchstücke von dem Objekt als Polygone verbunden werden und es dann im ISO-Designer (nächster Schritt) zu Problemen kommt. Das Objekt wird nur an diesen Teilen sichtbar sein!!!

Es hilft dann, in Solid Edge das Objekt noch ein Mal Schritt für Schritt durch zu gehen.

Falls man beim Klick auf die Linie keine lila/pinke Linie hat, sondern eine gestrichelte „lila“ Linie hat, hat man den Fehler vielleicht schon gefunden.

Dieses gestrichelte Linie mit z.B. dem Trimmwerkzeug aus Solid Edge entfernen und mit einer neuen Linie die Punkte verbinden.



Vorgang mit allen einzelnen Objekten wiederholen Speichern Schließen

15.4 ISO-Designer

- Die Unterstützung von DXFs für den Import von Vektorgrafiken wurde von Jetter entfernt.
- ISO-Designer 5.3.1 ist die letzte die noch DXF importieren kann.

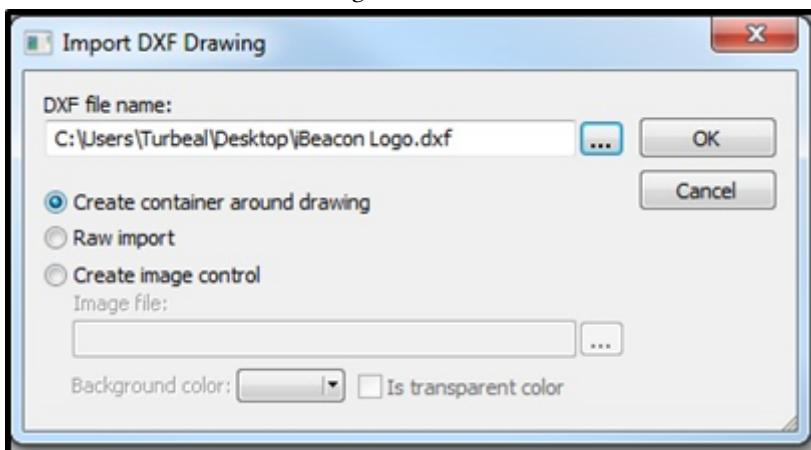
File → New → Workspace: Name der .dxf Datei) File → New → Project : neuste Plattform wählen (2010) höchste Auflösung (480x480) Display name und Project name mit Name der .dxf Datei benennen

Linie oder irgendwas in WorkingSet.jvi zeichnen

Rechtsklick in DataMask.jvi → DXF importieren → Haken bei create container around drawing

WICHTIG: im Working Set bleibt bis zum Schluss der Strich/Linie drin.

Das Symbol wird auch in DataMask eingefügt. Zwei verschiedene Registerkarten !!! Sonst kommt es am Ende zu einer Fehlermeldung und es funktioniert nicht !!!



Speichern

Close Workspace (wegen BUG)

Wieder öffnen über Recent Workspaces

Füllattribute ausfüllen

Build → Build All

Build → Deploy

Über „Deployment“ sollte man jetzt „0 Error(s)“ lesen können.

Falls das nicht so ist, den Strich unter WorkingSet überprüfen. (siehe oben)

15.5 Alternative mit Autodesk Fusion 360 und Plugin

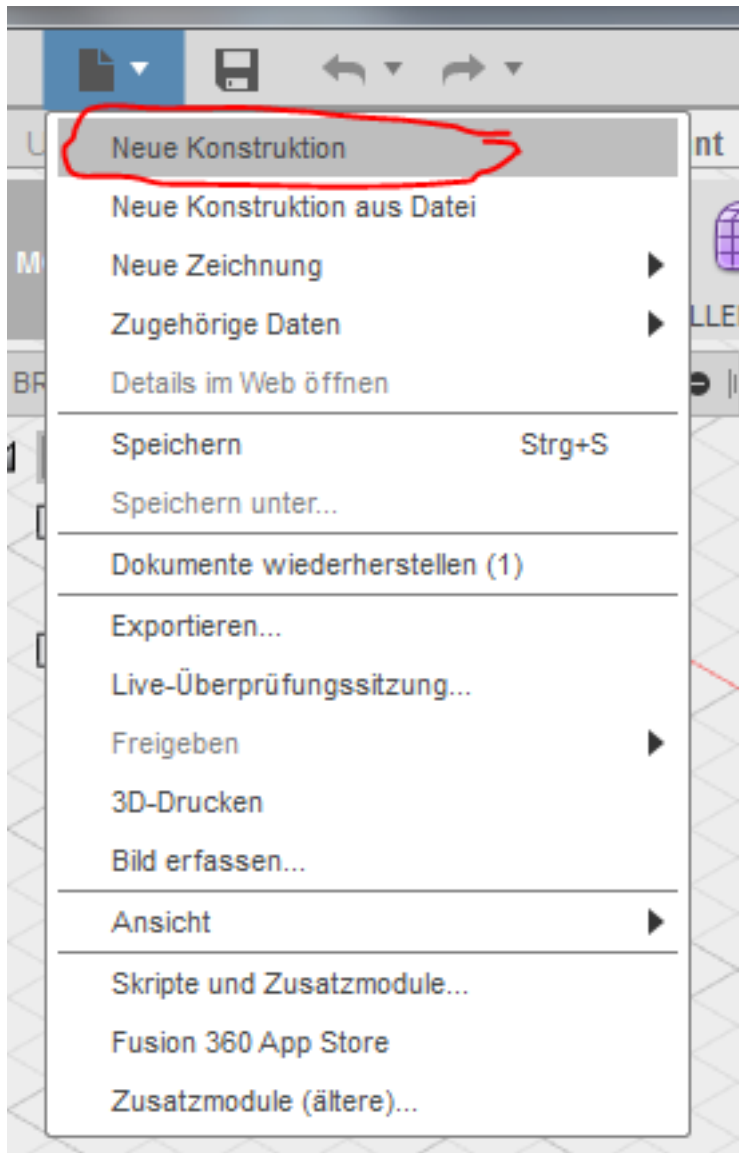
Software und Plugin installieren Fusion 360 herunterladen und installieren

Studenten / Schüler kostenlos.

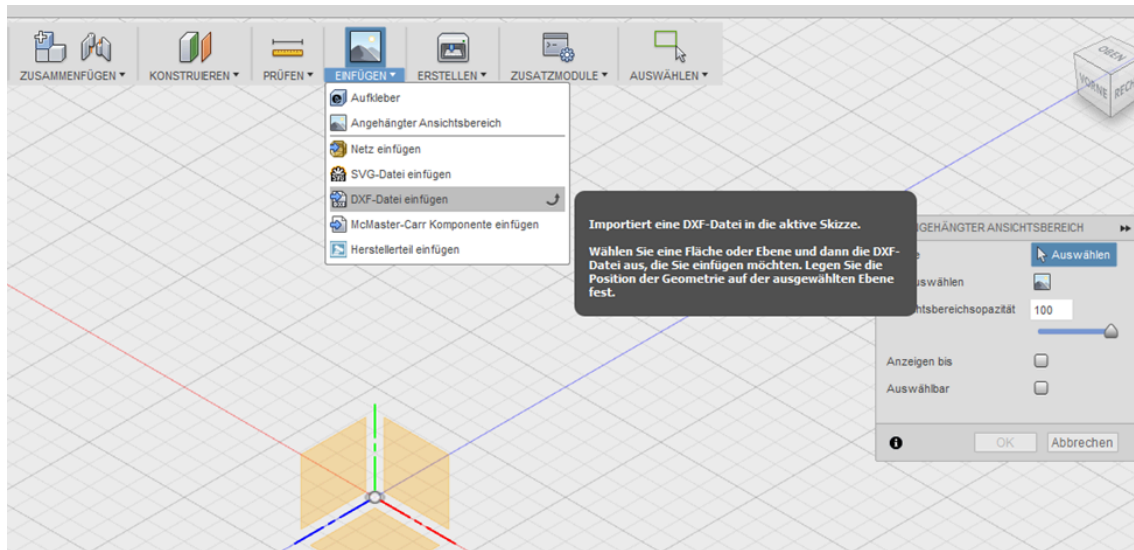
Das Plugin „DXFSplineToPolyline-win64“ installieren.

15.5.1 Fusion 360 anwenden.

Neue Konstruktion erstellen

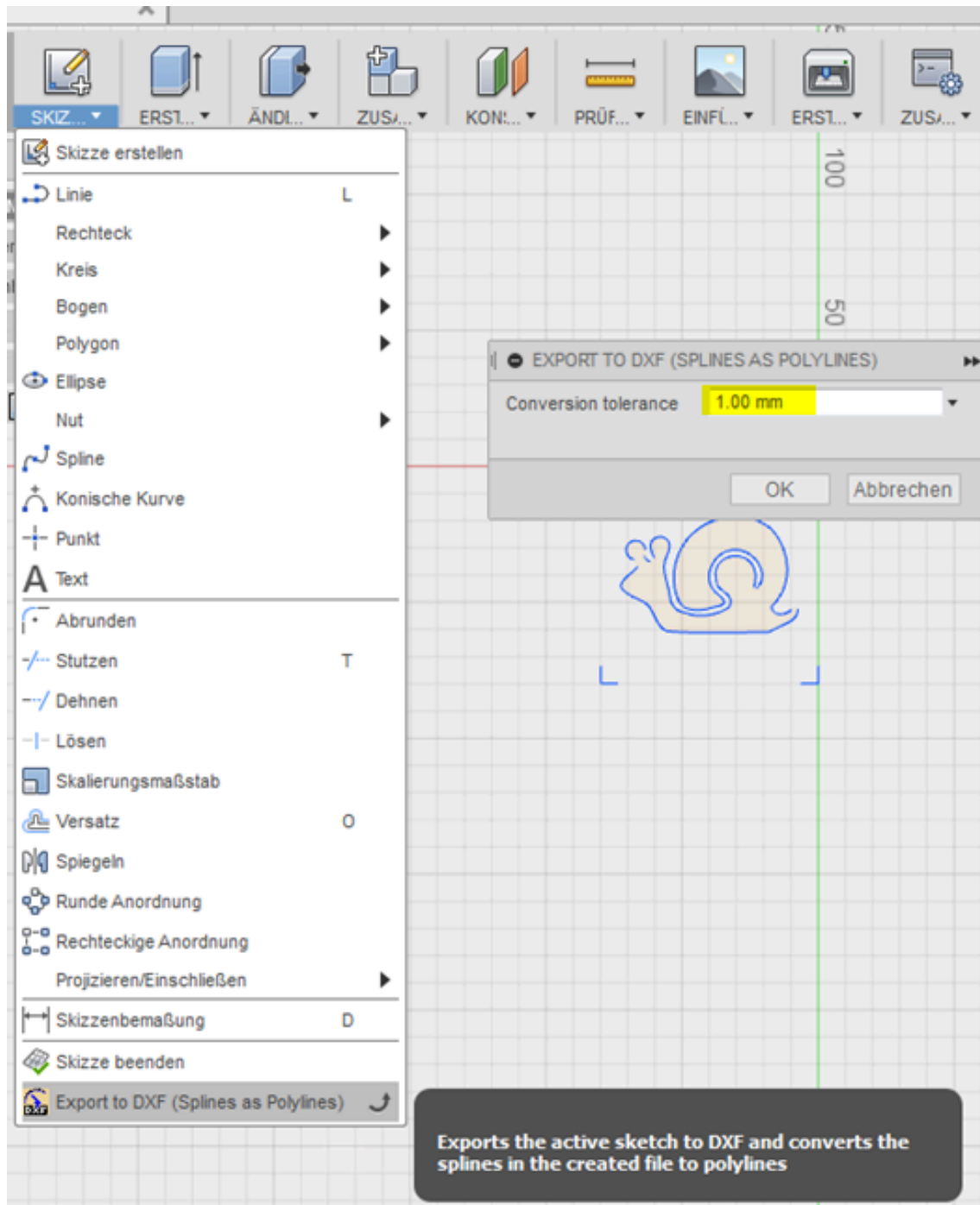


DXF importieren (Einfügen/DXF-Datei einfügen) untere Ebene und Datei auswählen



Optional: Nach Bedarf bearbeiten, skalieren (Skizze/Skalierungsmaßstab (Maßstab so wählen das Zielgröße z.B. 80x80) und in den Ursprung verschieben (Ändern/Verschieben)

Über Menüpunkt Skizze kann das Plugin „Export to DXF (Splines as Polyline)“ ausgeführt werden Als Toleranz wird 1.00 mm angegeben.



Weiter mit Libre CAD

16. Videos

16.1 ISO-Designer: Deutsch

- ISO-Designer: Deutsch
- 1. ISO-Designer: Objekte kopieren in Projekt
- 2. ISO-Designer: Objekte ausrichten
- 3. ISO-Designer: Importieren und Exportieren von Texten von und zu einer Excel CSV-Datei
- 4. ISO-Designer: Sprachumschaltung von Texten und Bildern
- 5. ISO-Designer: Komfortable Suchfunktionen
- 6. ISO-Designer: Verwenden der Objekt Bibliothek
- 7. ISO-Designer: Objekteigenschaften uebernehmen

16.2 ISO-Designer: englisch

- ISO-Designer: englisch
- 1. ISO-Designer: Copy Objects
- 2. ISO-Designer: Arranging Objects
- 3. ISO-Designer: Using Object-Library
- 4. ISO-Designer: Search Functions
- 5. ISO-Designer: Inheriting Object-Properties
- 6. ISO Designer: Importing/exporting texts into/from an Excel CSV file
- 7. ISO-Designer: Language selection within texts and images

17. logiBUS®

17.1 Podcast

- Digitale Gedächtnisse: Das SR-Flip-Flop im logiBUS®-System erklärt
- ESP32 als Industrie-SPS: Revolution mit Eclipse 4diac und logiBUS®
- From Cable Chaos to Custom Control: How Logibus is Revolutionizing Agricultural Tech with Accessible ISOBUS
- Logibus Revolution: Unpacking the „No Mapping“ Future and Streamlined Development
- Logibus Unleashed: How Eliminating „Mapping“ Simplifies Complex Systems and Boosts Usability

17.2 Video

- LogiBUS Projekt Update

18. Objekte

Die **ISO 11783-6** spezifiziert ein seriell betriebenes Datenkommunikationsnetzwerk für die Steuerung und Kommunikation bei landwirtschaftlichen oder forstwirtschaftlichen Traktoren sowie montierten, halbmontierten, gezogenen oder selbstfahrenden Geräten¹. Genauer gesagt handelt es sich um ein **virtuelles Terminal**, das als Schnittstelle für die Steuerung und Überwachung verschiedener Anbaugeräte dient. Mit ISOBUS können Landwirte Anbaugeräte unterschiedlicher Art über dasselbe Terminal steuern, unabhängig vom Hersteller¹. Dies ermöglicht eine bessere Übersichtlichkeit und Effizienz auf dem Traktor.



1. ISOBUS – Wikipedia. <https://de.wikipedia.org/wiki/ISOBUS>.
2. ISO 11783-6:2018. <https://www.iso.org/standard/71173.html>.
3. ISO 11783-6 - 2018-06 - DIN Media - Beuth. <https://www.dinmedia.de/de/norm/iso-11783-6/291680983>.
4. Rothmund-Wodok. ISOBUS - Norm ISO 11783 - <gil-net.de>. https://gil-net.de/Publikationen/22_163.pdf.
5. en.wikipedia.org. https://en.wikipedia.org/wiki/ISO_11783.

18.1 Attribute

18.1.1 Empfohlene Lektüre:

- Hilfe Jetter ISO-Designer.
 - ISO-Objekte
 - Die einzelnen ISO-Objekte
 - Font-, Line-, Fill- und Input-Attributes

The screenshot shows the ISO-Designer application window. The left sidebar contains a table of contents with 'Font-, Line-, Fill- und Input-Attributes' selected. The main content area displays the help page for these attributes, including an introduction and a table of font attributes.

Font-, Line-, Fill- und Input-Attributes

Einleitung

Die Objekte Font-, Line-, Fill- und Input-Attributes dienen dazu, bestimmte Eigenschaften von Objekten zu gruppieren. Die Objekte Font-, Line-, Fill- und Input-Attributes gruppieren Eigenschaften, die z. B. das Aussehen von Texten, Werten, Linien und Füllungen definieren. Sie können die Objekte als Unterobjekte wiederverwenden, wenn mehrere Objekte den gleichen Satz von Eigenschaften haben sollen. Die Objekte sind im **Object Pool** vorhanden. Sie können im [Fenster Properties](#) die einzelnen Eigenschaften anzeigen und anpassen.

Font Attributes

Die **Font Attributes** gruppieren Eigenschaften, die das Aussehen von Texten definieren.

Font Attributes/Font	Beschreibung
Size	Schriftgröße
Color	Farbe des Objekts
Bold	Fettschrift
Crossed Out	Durchgestrichene Schrift
Underline	Unterstrichene Schrift
Italic	Kursive Schrift
Inverted	Invertierte Schrift
Flash Inverted	Hintergrund- und Schriftfarben wechseln
Flash Hidden	Schrift blinkt

18.1.2 Podcast

- [ISOBUS-Oberflächen: Hintergrundfarbe vs. Füll-Attribute – So optimieren Sie Ihre UI und Performance](#)

18.2 Ausgabe

18.3 Auxiliary Funktionen

18.4 Bedienelemente

In der **ISO 11783-6** werden verschiedene Bedienelemente spezifiziert, die in landwirtschaftlichen und forstwirtschaftlichen Traktoren sowie Anbaugeräten verwendet werden. Hier sind die relevanten Bedienelemente gemäß ISO 11783-6:

1. **Key (Soft Key)** (ID 5): Softkeys sind programmierbare Tasten auf dem virtuellen Terminal. Die Softkey-Maske ermöglicht die Anpassung dieser Tasten je nach Kontext und Anbaugerät. Der Fahrer kann sie verwenden, um bestimmte Funktionen auszuführen oder Einstellungen vorzunehmen¹.
2. **Button** (ID 6): Buttons sind physische Schaltflächen oder Tasten auf dem Bedienfeld. Sie dienen dazu, bestimmte Aktionen auszulösen, z. B. das Starten oder Stoppen eines Anbaugeräts oder das Ändern von Einstellungen¹.

Diese Bedienelemente sind entscheidend für die effiziente Steuerung und Überwachung von Anbaugeräten über das virtuelle Terminal. Wenn du noch weitere Fragen hast, stehe ich gerne zur Verfügung! 🚗

Quelle: Unterhaltung mit Copilot, 18.7.2024

1. ISOBUS in Funktionalitäten - AEF Online. https://www.aef-online.org/fileadmin/user_upload/Content/pdfs/AEF_handfan_DE.pdf.
2. ISO 11783-6:2018. <https://www.iso.org/standard/71173.html>.
3. ISO 11783-6 - 2014-07 - DIN Media. <https://www.dinmedia.de/de/norm/iso-11783-6/215929001>.
4. ISO CONTROL von Reichhardt | PRILLINGER Ersatzteile mit Serviceplus. <https://www.prillinger.at/de/p/ISO-CONTROL-Joystick~00570058>.
5. ISOBUS-Nachrüstung - Reichhardt. <https://www.reichhardt.com/alle-produkte/isobus-kompetenz/>.
6. ISO 11783-6:2018(en), Tractors and machinery for agriculture and <https://www.iso.org/obp/ui/en/>.
7. ISO 11783 – STANDARD AND ITS IMPLEMENTATION - NTNU. <https://folk.ntnu.no/skoge/prost/proceedings/ifac2005/Fullpapers/02392.pdf>.
8. INTERNATIONAL ISO This is a preview of ISO 11783-6:2018. Click here to https://webstore.ansi.org/preview-pages/ISO/preview_ISO+11783-6-2018.pdf.
9. ISOBUS Maskmaker - ANEDO. <https://anedo.eu/produkt/isobus-mask-maker/>.
10. Atemwiderstände von Masken - DGUV Publikationen. <https://publikationen.dguv.de/widgets/pdf/download/article/4577>.
11. Übersicht: Typen von Atemschutzmasken | Eigenschaften Schutzmasken <https://www.heller-medizintechnik.de/informationen/patienten/schutzmasken/>.
12. Atemwiderstände von Masken - DGUV Informationsportal. https://www.dguv.de/medien/ipa/publikationen/ipa-journale/ipa-journale2022/ipa_journal_1_2022_atemwiderstaende_masken.pdf.

18.5 Bilder

18.6 Container

18.6.1 Podcast

- [ISOBUS-Container: Dynamische Bedienfelder für klare Sicht und mehr Effizienz](#)

18.7 Eingabe

18.8 Externe Objekte

18.9 Funktionen

18.10 Masken

18.10.1 Podcast

- [ISOBUS-Masken im VT: Das Herzstück der Land- und Baumaschinen-Interfaces \(ISO 11783-6\)](#)

Die **ISO 11783-6** beschreibt ein seriell betriebenes Datenkommunikationsnetzwerk für landwirtschaftliche und forstwirtschaftliche Traktoren sowie montierte, halbmontierte, gezogene oder selbstfahrende Geräte. Ein zentrales Element ist das **virtuelle Terminal**, das als Schnittstelle für die Steuerung und Überwachung verschiedener Anbaugeräte dient. Hier sind die Maskentypen gemäß ISO 11783-6 aufgelistet:

1. **Working Set (Arbeitsset)** (ID 0): Dieses Objekt definiert die Anzeige von Daten aus verschiedenen Arbeitssets auf einer Maske. Es ermöglicht die gleichzeitige Darstellung von Informationen von mehreren Anbaugeräten auf demselben virtuellen Terminal.
2. **Data Mask (Datenmaske)** (ID 1): Die Datenmaske ist ein rechteckiger Bereich auf dem Display mit einer Mindestauflösung von 200 x 200 Pixeln. Sie zeigt verschiedene Objekte wie Schaltflächen, Zahlen- und Zeichenfelder, geometrische Formen, Messgeräte und Balkendiagramme an².
3. **Alarm Mask (Alarmmaske)** (ID 2): Diese Maske dient zur Anzeige von Alarmen und Warnungen. Sie informiert den Fahrer über kritische Zustände oder Ereignisse, z. B. Motorüberhitzung oder niedriger Kraftstoffstand.
4. **Soft Key Mask (Softkey-Maske)** (ID 4): Softkeys sind programmierbare Tasten auf dem virtuellen Terminal. Die Softkey-Maske ermöglicht die Anpassung dieser Tasten je nach Kontext und Anbaugerät. Der Fahrer kann sie verwenden, um bestimmte Funktionen auszuführen oder Einstellungen vorzunehmen¹.

Ich hoffe, das hilft! Wenn du noch weitere Fragen hast, stehe ich gerne zur Verfügung. 🚜

Quelle: Unterhaltung mit Copilot, 18.7.2024

1. ISO 11783 – STANDARD AND ITS IMPLEMENTATION - NTNU. <https://folk.ntnu.no/skoge/prost/proceedings/ifac2005/Fullpapers/02392.pdf>.
2. ISO 11783-6:2018(en), Tractors and machinery for agriculture and <https://www.iso.org/obp/ui/en/>.
3. INTERNATIONAL ISO This is a preview of ISO 11783-6:2018. Click here to https://webstore.ansi.org/preview-pages/ISO/preview_ISO+11783-6-2018.pdf.
4. ISOBUS Maskmaker - ANEDO. <https://anedo.eu/produkt/isobus-mask-maker/>.
5. Atemwiderstände von Masken - DGUV Publikationen. <https://publikationen.dguv.de/widgets/pdf/download/article/4577>.
6. Übersicht: Typen von Atemschutzmasken | Eigenschaften Schutzmasken <https://www.heller-medizintechnik.de/informationen/patienten/schutzmasken/>.
7. Atemwiderstände von Masken - DGUV Informationsportal. https://www.dguv.de/medien/ipa/publikationen/ipa-journale/ipa-journale2022/ipa_journal_1_2022_atemwiderstaende_masken.pdf.

18.11 Spezial & Sonstiges

18.12 Variablen

18.12.0.1 Empfohlene Lektüre:

- ISO 11783-6
 - Kapitel B.13.2
 - Kapitel B.13.3
- Hilfe Jetter ISO-Designer.

- ISO-Objekte
 - Die einzelnen ISO-Objekte
 - Number- und Stringvariablen

ISO-Designer
— □ ×

Ausblenden
Zurück
Drucken
Optionen

Inhalt Index Suchen

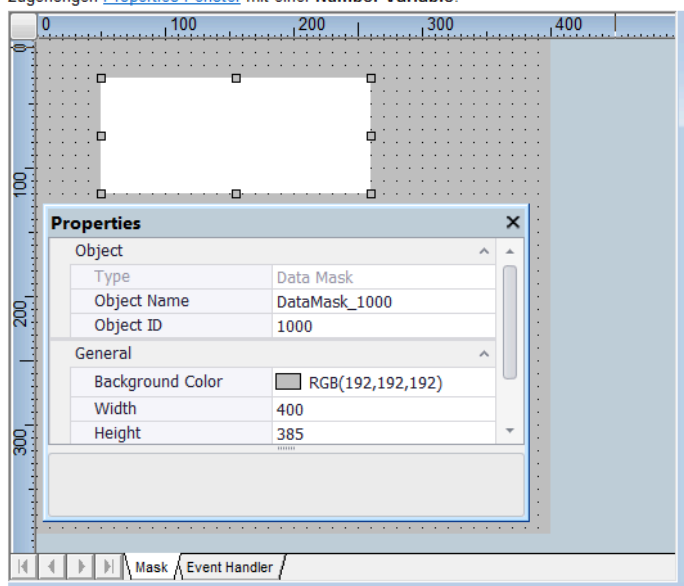
- [-] ? Vorspann
- [-] ? So können Sie uns erreichen
- [-] ? Versionsupdates
- [-] ? Produktbeschreibung
- [-] ? Erstes ISO-Projekt anlegen
- [-] ? ISO-Designer konfigurieren
- [-] ? ISO-Objekte
 - [-] ? Objekteigenschaften (Properties)
 - [-] ? Die einzelnen ISO-Objekte
 - [-] ? Das Objekt Line
 - [-] ? Das Objekt Polyline
 - [-] ? Das Objekt Polygon
 - [-] ? Das Objekt Ellipse
 - [-] ? Das Objekt Rectangle
 - [-] ? Das Objekt Button
 - [-] ? Das Objekt Object Pointer
 - [-] ? Das Objekt Input Boolean
 - [-] ? Das Objekt Input Number
 - [-] ? Das Objekt Input String
 - [-] ? Das Objekt Input List
 - [-] ? Das Objekt Output String
 - [-] ? Das Objekt Output Number
 - [-] ? Das Objekt Output List
 - [-] ? Das Objekt Soft Key
 - [-] ? Das Object Linear Bargraph
 - [-] ? Das Objekt Arched Bargraph
 - [-] ? Das Objekt Meter
 - [-] ? Das Objekt Container
 - [-] ? Das Objekt Macro
 - [-] ? Das Objekt Graphics Context
 - [-] ? Das Objekt Color Map
 - [-] ? Das Objekt Picture Graphic
 - [-] ? Animation object
 - [-] ? External Object Pointer
 - [-] ? External Object Definition
 - [-] ? External Reference NAME
 - [-] ? Aux Input
 - [-] ? Aux Function
 - [-] ? Aux Object Pointer
 - [-] ? **Number- und Stringvariablen**
 - [-] ? Font-, Line-, Fill- und Input-Att
- [-] ? Objekte erstellen
- [-] ? Die verschiedenen Maskentypen
- [-] ? ISO-konforme Masken erstellen
- [-] ? Handling von Objekten (Object Pool u
- [-] ? Der Event Handler
- [-] ? Mehrsprachigkeit und Ressourcenver
- [-] ? IOP-Datei

Number- und Stringvariablen

← →

Einleitung

Mit den Objekten **Number Variable** und **String Variable** speichern Sie Integer-Werte und Strings. Folgendes Bild zeigt ein Objekt **Input Number** auf einer Maske und einen Ausschnitt des zugehörigen **Properties-Fenster** mit einer **Number Variable**:



Eine Number Variable anlegen

Eine **Number Variable** dient zum Speichern eines Integer-Werts. Eine **Number Variable** legen Sie wie folgt an:

Schritt	Vorgehen
1	Wählen Sie das Objekt aus, dem Sie die Number Variable zuweisen wollen.
2	Wählen Sie im Fenster Properties unter Number Variable die Zeile Object ID aus.
3	Klappen Sie die Dropdown-Liste auf.
4	Wählen Sie aus der Liste eine Number Variable aus oder legen Sie über Add new

19. Referenztabellen (Annex A)

Diese Sektion enthält wichtige Nachschlagetabellen aus dem Anhang A der ISO 11783-6 Norm. Diese sind besonders hilfreich für die Softwareentwicklung (C/C++ Code) und das Erstellen von Makros.

19.1 Kommando-Übersicht (ISO 11783-6 Table A.5)

Diese Tabelle listet alle Befehle (Messages), die zwischen ECU (Working Set) und VT ausgetauscht werden können.

- **Function (Decimal/Hex):** Der Code im ersten Byte der Nachricht (bei Destination Specific PGN).
- **Allowed in Macro:** Ob der Befehl in einem Makro-Objekt gespeichert werden darf.

Clause	Message Name	PGN	Function (Hex)	Macro?	VT Vers.
C.2.3	Object Pool Transfer	ECU->VT	11	No	2
C.2.4	End of Object Pool	ECU->VT	12	No	2
D.4	Get Number of Soft Keys	ECU->VT	C2	No	2
D.6	Get Text Font Data	ECU->VT	C3	No	2
D.8	Get Hardware Message	ECU->VT	C7	No	2
D.12	Get Window Mask Data	ECU->VT	C4	No	4
D.14	Get Supported Objects	ECU->VT	C5	No	4
D.16	Screen Capture	ECU->VT	C6	No	6
D.18	Identify VT	ECU->VT	BB	No	4
E.2	Get Versions	ECU->VT	DF	No	2
E.4	Store Version	ECU->VT	D0	No	2
E.6	Load Version	ECU->VT	D1	No	2
E.8	Delete Version	ECU->VT	D2	No	2
E.10	Extended Get Versions	ECU->VT	D3	No	5
E.12	Extended Store Version	ECU->VT	D4	No	5
E.14	Extended Load Version	ECU->VT	D5	No	5
E.16	Extended Delete Version	ECU->VT	D6	No	5
F.2	Hide/Show Object	ECU->VT	A0	Yes	2
F.4	Enable/Disable Object	ECU->VT	A1	Yes	2
F.6	Select Input Object	ECU->VT	A2	Yes	2
F.8	ESC Command	ECU->VT	92	No	2
F.10	Control Audio Signal	ECU->VT	A3	Yes	2
F.12	Set Audio Volume	ECU->VT	A4	Yes	2
F.14	Change Child Location	ECU->VT	A5	Yes	2

F.16	Change Child Position	ECU->VT	B4	Yes	2
F.18	Change Size	ECU->VT	A6	Yes	2
F.20	Change Background Colour	ECU->VT	A7	Yes	2
F.22	Change Numeric Value	ECU->VT	A8	Yes	2
F.24	Change String Value	ECU->VT	B3	Yes	2
F.26	Change End Point	ECU->VT	A9	Yes	2
F.28	Change Font Attributes	ECU->VT	AA	Yes	2
F.30	Change Line Attributes	ECU->VT	AB	Yes	2
F.32	Change Fill Attributes	ECU->VT	AC	Yes	2
F.34	Change Active Mask	ECU->VT	AD	Yes	2
F.36	Change Soft Key Mask	ECU->VT	AE	Yes	2
F.38	Change Attribute	ECU->VT	AF	Yes	2
F.40	Change Priority	ECU->VT	B0	Yes	2
F.42	Change List Item	ECU->VT	B1	Yes	2
F.44	Delete Object Pool	ECU->VT	B2	No	2
F.46	Lock/Unlock Mask	ECU->VT	BD	Yes	4
F.48	Execute Macro	ECU->VT	BE	Yes	4
F.50	Change Object Label	ECU->VT	B5	Yes	4
F.52	Change Polygon Point	ECU->VT	B6	Yes	4
F.54	Change Polygon Scale	ECU->VT	B7	Yes	4
F.56	Graphics Context Command	ECU->VT	B8	No	4
F.58	Get Attribute Value	ECU->VT	B9	No	4
F.60	Select Colour Map/Palette	ECU->VT	BA	Yes	4
F.62	Execute Extended Macro	ECU->VT	BC	Yes	5
F.64	Select Active Working Set	ECU->VT	90	Yes	6

19.2 Event IDs (ISO 11783-6 Table A.3)

Diese Tabelle listet alle Ereignis-IDs (Event IDs), die in Makros (Macro object, ID 28) oder als Trigger für Attribute verwendet werden können.

Hinweis: Ab VT Version 5 werden 16-Bit Makro-Referenzen unterstützt.

Event ID	Event Name	Auslöser (Occurs when)
0	Reserved	-
1	On Activate	Working Set wird aktiv.
2	On Deactivate	Working Set wird inaktiv.
3	On Show	Container/Maske wird sichtbar.
4	On Hide	Container/Maske wird unsichtbar.
5	On Enable	Objekt wird aktiviert (Input enabled).

6	On Disable	Objekt wird deaktiviert (Input disabled).
7	On Change Active Mask	Die aktive Datenmaske wurde gewechselt.
8	On Change Soft Key Mask	Die Softkey-Maske wurde gewechselt.
9	On Change Attribute	Ein Attribut wurde geändert.
10	On Change Background Colour	Hintergrundfarbe wurde geändert.
11	On Change Font Attributes	Schriftattribute wurden geändert.
12	On Change Line Attributes	Linienattribute wurden geändert.
13	On Change Fill Attributes	Füllattribute wurden geändert.
14	On Change Child Location	Die Position eines Kind-Objekts wurde geändert.
15	On Change Size	Die Größe eines Objekts wurde geändert.
16	On Change Value	Der Wert (Zahl/String) hat sich geändert.
17	On Change Priority	Die Alarm-Priorität hat sich geändert.
18	On Change End Point	Endpunkt einer Linie wurde geändert.
19	On Input Field Selection	Fokus erhalten (Navigation).
20	On Input Field Deselection	Fokus verloren.
21	On ESC	Eingabe abgebrochen (ESC).
22	On entry of a value	Wert bestätigt (ENTER), keine Änderung nötig.
23	On entry of a new value	Wert bestätigt (ENTER) UND geändert.
24	On Key Press	Taste/Button gedrückt.
25	On Key Release	Taste/Button losgelassen.
26	On Change Child Position	Relative Position im Container geändert (Scrolling).
27	On Pointing Event Press	Touch-Display gedrückt.
28	On Pointing Event Release	Touch-Display losgelassen.
29-239	Reserved	-
240-254	Proprietary Events	Herstellerspezifische Events.
255	Reserved	-
255	Use Extended Macro Ref.	(Nur in Byte-Stream) Kennzeichnet 16-Bit Makro ID.

19.3 Standard Farbpalette (ISO 11783-6 Table A.4)

Die VT-Standardfarbpalette definiert 256 Farben. Die ersten 16 Farben sind identisch mit der Standard-VGA-Palette.

Index	Name / Beschreibung	R, G, B Wert	Hex Code (RGB)
0	Black	0, 0, 0	#000000
1	White	255, 255, 255	#FFFFFF
2	Green	0, 153, 0	#009900
3	Teal	0, 153, 153	#009999
4	Maroon	153, 0, 0	#990000

5	Purple	153, 0, 153	#990099
6	Olive	153, 153, 0	#999900
7	Silver	204, 204, 204	#CCCCCC
8	Grey	153, 153, 153	#999999
9	Blue	0, 0, 255	#0000FF
10	Lime	0, 255, 0	#00FF00
11	Cyan	0, 255, 255	#00FFFF
12	Red	255, 0, 0	#FF0000
13	Magenta	255, 0, 255	#FF00FF
14	Yellow	255, 255, 0	#FFFF00
15	Navy	0, 0, 153	#000099
...
232-255	Proprietary	-	VT-Hersteller spezifisch

Hinweis: Für die Indizes 16 bis 231 ist eine algorithmische Verteilung im Farbwürfel (6x6x6) definiert. Siehe ISO 11783-6 Table A.4 für die vollständige Liste.

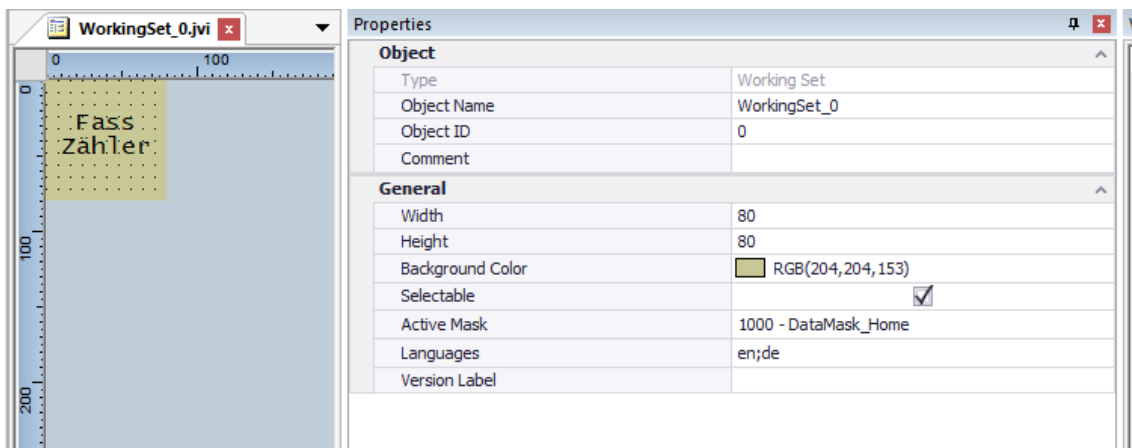
Weitere Informationen und Visualisierungen finden sich im [ISOBUS Wiki - Colours](#) von Tobias Tenberg.

19.3.0.1 Transparenz

Der Farbindex, der im Attribut „Transparency Colour“ (z. B. bei Picture Graphics) angegeben wird, wird nicht gezeichnet. Oft wird hierfür eine „Signalfarbe“ wie Magenta (Index 13) verwendet, wenn diese im Bild nicht vorkommt.

20. Isobus objects

20.1 ID 0 – Working set – ISO 11783-6 – B.1



Das **Working Set** (Arbeitsset) Objekt mit der **ID 0** ist das zentrale Verwaltungselement einer Arbeitsgruppe (Working Set) im ISOBUS. Es definiert, wie sich die Arbeitsgruppe gegenüber dem Virtuellen Terminal (VT) präsentiert und welche Maske initial angezeigt wird.

20.1.1 Wichtige Eigenschaften (gemäß ISO 11783-6, Anhang B.1)

Jede Arbeitsgruppe muss **genau ein** Working Set Objekt in ihrem Objektpool definieren. Nur das VT kann dieses Objekt aktivieren.

20.1.1.1 Attribute und Record Format (Tabelle B.2)

Die folgende Tabelle beschreibt den Aufbau des Working Set Objekts im Objektpool (Byte-Stream).

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	0	3	Objekttyp = Working Set.
[1]	Background colour	Integer	1	0 – 255	4	Hintergrundfarbe.
[2]	Selectable	Boolean	1	0 oder 1	5	0 = FALSE, 1 = TRUE. Gibt an, ob das Working Set vom Bediener ausgewählt werden kann.

[3]	Active mask	Integer	2	0 – 65534	6 – 7	Objekt-ID der Data oder Alarm Mask, die angezeigt wird, wenn das Working Set aktiv ist.
-	Number of objects to follow	Integer	1	1 – 255	8	Anzahl der folgenden Kind-Objekte (Designator). Muss mindestens 1 sein.
-	Number of macros to follow	Integer	1	0 – 255	9	Anzahl der folgenden Makro-Referenzen.
-	Number of languages to follow	Integer	1	0 – 255	10	Anzahl der folgenden Sprachcodes.
-	Repeat: {Object ID}	Integer	2	0 – 65534	11 + ...	Objekt-ID eines Kind-Objekts (Teil des Designators). Liste aller Objekte erfolgt vor den Makros.
-	{X Location}	Signed Integer	2	-32768 bis +32767	13 + ...	Relative X-Position des Kind-Objekts (Pixel).
-	{Y Location}	Signed Integer	2	-32768 bis +32767	15 + ...	Relative Y-Position des Kind-Objekts (Pixel).
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach den Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros (oder Referenz auf 16-Bit Makro ID).
-	Repeat: {Language Code}	String	2	-	var.	(Nach den Makros) Zweistelliger Sprachcode

						gemäß ISO 639 (z.B. „de“, „en“).
--	--	--	--	--	--	----------------------------------

20.1.1.2 Designator (Child-Objekte)

Das Working Set Objekt dient als Container für eine grafische Kennung (Designator), die das Gerät repräsentiert (z. B. Icon und Name).

- **Struktur:** Die unter Number of objects to follow angegebene Anzahl von Objekten folgt direkt im Record. Jedes Kind-Objekt wird durch ID, X-Pos und Y-Pos definiert.
- **Platzbeschränkung:** Die Gesamtheit dieser Objekte muss in den Bereich eines **Softkeys** passen. Das VT schneidet alles ab, was über diesen Bereich hinausragt.
- **Koordinaten:** Relativ zur oberen linken Ecke des Working Set Objekts.

20.1.1.3 Makros (Events)

Das Working Set definiert Ereignisse, die Makros auslösen können. Siehe Tabelle B.1 für die Event-Definitionen.

- **On Activate:** Ausgelöst bei Auswahl des Working Sets durch den Bediener.
- **On Deactivate:** Ausgelöst beim Verlassen des Working Sets.
- **On Change Active Mask:** Ausgelöst durch das Kommando Change Active Mask.

20.1.1.4 Sprachenunterstützung

Die Liste der Sprachcodes (Language Code) informiert das VT, welche Sprachen das Working Set unterstützt. Jeder Code besteht aus 2 ASCII-Zeichen (z.B. „de“, „en“, „fr“). Dies dient der Information; die eigentliche Sprachumschaltung erfolgt durch das Laden entsprechender Sprach-Objekte oder Texte.

20.1.2 Bedeutung für die Implementierung

Das Working Set Objekt ist der „Anker“ einer Applikation auf dem VT. Ohne ein korrekt definiertes Objekt ID 0 kann das VT die Arbeitsgruppe nicht identifizieren oder die erste Maske laden. Entwickler müssen sicherstellen, dass die Active mask AID auf eine gültige Datenmaske im Pool verweist.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Working Set](#) von Tobias Tenberg.

Hinweis: Für detaillierte Implementierungsinformationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.2 ID 1 – Data mask – ISO 11783-6 – B.2

The screenshot displays a software development environment with two main windows:

- DataMask_Home.jvi:** A design canvas showing a data mask layout. It includes:
 - A digital display for "Gesamtzähler" (Total Counter) with a value of 0 and a "Reset Gesamt" button.
 - A digital display for "Tageszähler" (Daily Counter) with a value of 0 and a "Reset Tages" button.
 - A semi-circular gauge on the left.
 - A vertical digital display on the right.
 - A "++" button at the bottom right.
- Properties:** A panel showing the configuration for the "Data Mask" object:
 - Object:** Type: Data Mask; Object Name: DataMask_Home; Object ID: 1000; Comment: (empty).
 - General:** Size: 480; Background Color: RGB(255, 255, 204); Soft Key Mask: 4000 - SoftKeyMask_Home.

Die **Data Mask** (Datenmaske) mit der **ID 1** ist das primäre Anzeigeelement für die Benutzeroberfläche einer Arbeitsgruppe. Sie dient als Hauptcontainer für alle visuellen Objekte (Buttons, Zahlenfelder, Grafiken), die dem Bediener auf dem Virtuellen Terminal (VT) angezeigt werden.

20.2.0.1 Attribute und Record Format (Tabelle B.4)

Die folgende Tabelle beschreibt den Aufbau des Data Mask Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	1	3	Objekttyp = Data mask.
[1]	Back-ground colour	Integer	1	0 – 255	4	Hintergrund-farbe der Maske.
[2]	Soft Key Mask	Integer	2	0 – 65534, 65535	5 – 6	Objekt-ID der zugehörigen Soft Key Mask (65535 = NULL).
-	Number of objects to follow	Integer	1	0 – 255	7	Anzahl der direkt enthaltenen Objekte.
-	Number of macros to follow	Integer	1	0 – 255	8	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	9 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	11 + ...	X-Position relativ zur Maske (Pixel).
-	{Y Location}	Signed Integer	2	-32768 bis +32767	13 + ...	Y-Position relativ zur Maske (Pixel).
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.2.0.2 Struktur und Child-Objekte

Die Datenmaske dient als Container für alle sichtbaren Elemente.

- **Objekt-Liste:** Jedes Kind-Objekt wird mit seiner ID und seiner Position (X/Y) definiert. Jedes Set aus ID und Position belegt 6 Bytes.
- **Reihenfolge:** Objekte werden in der gelisteten Reihenfolge gezeichnet (Z-Order). Höhere Indizes liegen über niedrigeren.
- **Koordinaten:** Die Positionierung erfolgt absolut in VT-Pixeln, bezogen auf die obere linke Ecke der Maske (0,0).

20.2.1 Ereignisse (Events - Tabelle B.3)

Die Datenmaske reagiert auf folgende Ereignisse:

- **On Show:** Ausgelöst, wenn die Maske sichtbar wird. Das VT zeichnet den Hintergrund, die Kinder und die Soft Key Mask.
- **On Hide:** Ausgelöst, wenn die Maske vom Display entfernt wird.
- **On Refresh:** Ausgelöst bei Änderungen an Kind-Objekten, die ein Neuzeichnen erfordern.
- **On Change Background Colour:** Reaktion auf eine Änderung der Hintergrundfarbe.
- **On Change Soft Key Mask:** Reaktion auf den Wechsel der zugewiesenen Soft Key Mask.
- **Pointing Events:** press und release bei Touch-Bedienung auf der Maskenfläche.

20.2.2 Verhalten und Einschränkungen

- **Zusammenhang mit Softkeys:** Jede Datenmaske „besitzt“ eine Softkey-Maske. Wenn die Datenmaske gewechselt wird, wechselt das VT in der Regel auch das Softkey-Layout.
- **Refresh:** Wenn ein untergeordnetes Objekt (Child) geändert wird, sorgt das VT für ein Redraw der betroffenen Bereiche.
- **Sichtbarkeit:** Es kann immer nur eine Datenmaske (oder Alarmmaske) pro Arbeitsgruppe aktiv und im Fokus des VT sein.

20.2.3 Bedeutung für die Implementierung

Die Datenmaske ist das Herzstück des HMI-Designs. Entwickler müssen darauf achten, dass die Auflösung der Maske zu den Fähigkeiten des VTs passt (Standard-Mindestauflösung oft 200x200 Pixel, moderne VTs bieten deutlich mehr). Eine effiziente Nutzung von Makros auf Masken-Events (z. B. On Show) kann helfen, Initialisierungen direkt im VT auszuführen.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Data Mask](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.3 ID 2 – Alarm Mask – ISO 11783-6 – B.3

Die **Alarm Mask** (Alarmmaske) mit der **ID 2** dient zur Anzeige kritischer Informationen oder Warnungen. Sie hat Vorrang vor normalen Datenmasken und kann je nach Priorität das gesamte Display oder Teile davon überlagern.

20.3.0.1 Attribute und Record Format (Tabelle B.6)

Die folgende Tabelle beschreibt den Aufbau des Alarm Mask Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	2	3	Objekttyp = Alarm Mask.
[1]	Back-ground colour	Integer	1	0 – 255	4	Hintergrund-farbe der Maske.

[2]	Soft Key Mask	Integer	2	0 – 65534, 65535	5 – 6	Objekt-ID der zugehörigen Soft Key Mask (65535 = NULL).
[3]	Priority	Integer	1	0 – 2	7	0 = Hoch, 1 = Mittel, 2 = Niedrig.
[4]	Acoustic signal	Integer	1	0 – 3	8	0 = Höchste Prio, 1 = Mittel, 2 = Niedrig, 3 = Aus.
-	Number of objects to follow	Integer	1	0 – 255	9	Anzahl der direkt enthaltenen Objekte.
-	Number of macros to follow	Integer	1	0 – 255	10	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	11 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	13 + ...	X-Position relativ zur Maske (Pixel).
-	{Y Location}	Signed Integer	2	-32768 bis +32767	15 + ...	Y-Position relativ zur Maske (Pixel).
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.3.0.2 Prioritätsstufen und Darstellung

Die Priorität steuert nicht nur die Reihenfolge der Alarme, sondern oft auch deren visuelle Darstellung auf dem VT:

- **High Priority (0):** Der Bediener ist in Gefahr oder es liegt eine schwere Fehlfunktion vor. Diese Alarme müssen sofort quittiert werden oder erzwingen den Fokus des Terminals.
- **Medium Priority (1):** Normale Fehlfunktion der Maschine.
- **Low Priority (2):** Nur zur Information (Statusmeldungen).

20.3.1 Ereignisse (Events - Tabelle B.5)

Die Alarmmaske reagiert auf folgende Ereignisse:

- **On Show:** Ausgelöst, wenn die Maske sichtbar wird. Das VT sendet eine VT Status Nachricht.

- **On Hide:** Ausgelöst, wenn die Maske vom Display entfernt wird.
- **On Refresh:** Ausgelöst bei Änderungen an Objekten innerhalb der Maske.
- **On Change Priority:** Bei Änderung der Priorität bewertet das VT alle aktiven Alarme neu.
- **On Change Soft Key Mask:** Wechselt die Softkey-Belegung bei aktiver Alarmmaske.
- **On Change Child Location / Position:** Aktualisierung der Kind-Objekte.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.3.2 Verhalten bei mehreren Alarmen

Wenn mehrere Alarmmasken von verschiedenen Arbeitsgruppen gleichzeitig aktiv sind, bestimmt die Priorität (AID 3), welche Maske im Vordergrund steht. Bei gleicher Priorität liegt die Entscheidung meist beim VT (oft zeitliche Abfolge).

20.3.3 Bedeutung für die Implementierung

Alarmmasken sollten sparsam und nur für echte Warnungen eingesetzt werden, da sie den Arbeitsfluss des Bedieners unterbrechen. Es ist wichtig, die passende akustische Unterstützung (AID 4) zu wählen, um die Aufmerksamkeit zu steuern, ohne den Bediener zu stressen.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Alarm Mask](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.4 ID 3 – Container – ISO 11783-6 – B.4

Das **Container** Objekt mit der **ID 3** dient dazu, mehrere Objekte logisch zu gruppieren. Ein Container selbst ist nicht sichtbar, ermöglicht aber das gemeinsame Verschieben, Ein-/Ausblenden oder Teilen einer gesamten Gruppe von Objekten.

20.4.0.1 Attribute und Record Format (Tabelle B.8)

Die folgende Tabelle beschreibt den Aufbau des Container Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	3	3	Objekttyp = Container.
[1]	Width	Integer	2	0 – 65535	4 – 5	Maximale Breite des Containers in Pixeln. Clipping erfolgt außerhalb dieses Bereichs.
[2]	Height	Integer	2	0 – 65535	6 – 7	Maximale Höhe des Containers in Pixeln. Clipping erfolgt außerhalb dieses Bereichs.

[3]	Hidden	Boolean	1	0 oder 1	8	0 = FALSE (Sichtbar), 1 = TRUE (Versteckt). Gibt an, ob der Container initial ausgeblendet ist.
-	Number of objects to follow	Integer	1	0 – 255	9	Anzahl der direkt enthaltenen Objekte.
-	Number of macros to follow	Integer	1	0 – 255	10	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	11 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	13 + ...	X-Position relativ zum Container (Pixel).
-	{Y Location}	Signed Integer	2	-32768 bis +32767	15 + ...	Y-Position relativ zum Container (Pixel).
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.4.0.2 Relative Positionierung und Clipping

Innerhalb eines Containers beginnt ein **eigenes Koordinatensystem**:

- **Relative Koordinaten:** Die X- und Y-Positionen der Child-Objekte beziehen sich auf die obere linke Ecke des Containers.
- **Clipping:** Alle Objekte oder Teile von Objekten, die außerhalb der durch Width und Height definierten Fläche liegen, werden vom VT abgeschnitten und nicht angezeigt.
- **Gruppen-Verschiebung:** Wenn der Container verschoben wird (z. B. per Change Child Position), verschieben sich alle darin enthaltenen Objekte automatisch mit.

20.4.1 Ereignisse (Events - Tabelle B.7)

Der Container reagiert auf folgende Ereignisse:

- **On Show:** Ausgelöst durch das Kommando Show Object (auch wenn der Container bereits sichtbar ist). Das VT zeichnet die enthaltenen Objekte neu.
- **On Hide:** Ausgelöst durch das Kommando Hide Object. Der Container wird mit der Hintergrundfarbe der Eltern-Maske überzeichnet.
- **On Refresh:** Ausgelöst bei Änderungen an untergeordneten Objekten, die ein Neuzeichnen erfordern.
- **On Change Child Location / Position:** Aktualisierung der Position von Kind-Objekten.

- **On Change Size:** Reaktion auf Größenänderung des Containers.

20.4.2 Nutzung in der Praxis

Container sind essenziell für dynamische Benutzeroberflächen:

- **Ein-/Ausblenden:** Mit dem Kommando `IsoVtcCmd_ObjHideShow` können komplexe Bedienfelder oder Statusanzeigen auf Knopfdruck erscheinen oder verschwinden.
- **Platzersparnis:** Mehrere Container können an der gleichen Stelle liegen; durch geschicktes Umschalten der Sichtbarkeit lassen sich verschiedene „Registerkarten“ oder Modi realisieren.

20.4.2.1 Beispiele aus dem ISO-Designer

The top screenshot shows the 'ISO-Designer' window. On the left is a tree view of objects, with 'Das Objekt Container' selected. On the right, a window titled 'Das Objekt Container' contains an introduction text and a diagram showing two gauges (one with a scale from 0 to 200, the other from 0 to 5) grouped within a dashed rectangular container box. Below the diagram are 'Voraussetzungen zum Zeichnen'.

The bottom screenshot shows the 'ISOBUS driver' window. On the left is a list of commands, with 'IsoVtcCmd_ObjHideShow()' selected. On the right, the details for this command are shown, including its signature, a description, and its parameters and return values.

```

IsoVtcCmd_ObjHideShow()

ISO_CLIENT_API iso_s16 IsoVtcCmd_ObjHideShow ( iso_u8 u8Instance,
                                                iso_u16 u16ObjId,
                                                iso_bool qVisible
                                                )

Command Hide/Show Object (Container) ( Part 6 - F.2 )

The Hide/Show Object command is used to hide or show a container object.

Parameters

[in] u8Instance iso_u8 :
    Identifier of the client instance

[in] u16ObjId iso_u16 :
    Object ID of Container

[in] qVisible iso_bool :
    0 = Hide, 1 = Show

Return values

iso_s16
    • E_NO_ERR - OK
    • E_OVERFLOW - buffer overflow
  
```

20.4.3 Bedeutung für die Implementierung

Da der Container ein logisches Element ist, verbraucht er selbst kaum Rechenleistung, ist aber mächtig in der Steuerung der Z-Order und Gruppierung. Entwickler sollten darauf achten, dass die `Width` und `Height` korrekt gesetzt sind, um ungewolltes Clipping zu vermeiden.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Container](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.5 ID 4 – Soft Key Mask – ISO 11783-6 – B.5

Die **Soft Key Mask** (Softkey-Maske) mit der **ID 4** ist ein spezieller Container, der die Belegung der physischen oder virtuellen Softkeys am Rand des Terminals definiert. Sie wird in der Regel einer Datenmaske oder Alarmmaske fest zugeordnet.

20.5.0.1 Attribute und Record Format (Tabelle B.10)

Die folgende Tabelle beschreibt den Aufbau des Soft Key Mask Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	4	3	Objekttyp = Soft Key Mask.
[1]	Background colour	Integer	1	0 – 255	4	Hintergrundfarbe. Das Key-Objekt hat ein eigenes Hintergrundattribut, das dieses überschreiben kann.
-	Number of objects to follow	Integer	1	0 – 255	5	Anzahl der enthaltenen Objekte (Keys oder Pointer).
-	Number of macros to follow	Integer	1	0 – 255	6	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	7 + ...	Objekt-ID eines enthaltenen Key-Objekts oder Pointers.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.5.0.2 Funktionsweise und Zuweisung

Eine Softkey-Maske enthält eine Liste von **Key Objekten** (ID 5), **Object Pointern** (ID 27) oder **External Object Pointern** (ID 43).

- **Reihenfolge:** Die Zuweisung zu den physischen Tasten am Terminal erfolgt strikt in der Reihenfolge der Liste.

- **NULL-Pointer:** Pointers auf die NULL Object ID reservieren eine Position (die folgenden Keys rücken nicht nach). Pointers auf NULL am Ende der Liste werden nicht angezeigt und nicht für das Paging berücksichtigt.
- **Paging:** Übersteigen die definierten Keys die Kapazität des VT, erstellt dieses automatisch Navigationshilfen (Pfeiltasten) zum Umblättern.

20.5.1 Ereignisse (Events - Tabelle B.9)

Die Softkey-Maske reagiert auf folgende Ereignisse:

- **On Show:** Ausgelöst, wenn die Maske sichtbar wird. Das VT zeichnet alle Kind-Objekte in der definierten Reihenfolge.
- **On Hide:** Ausgelöst beim Entfernen der Maske.
- **On Change Background Colour:** Reaktion auf eine Änderung der Hintergrundfarbe der Leiste.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.5.2 Zusammenspiel mit Datenmasken

Jede Datenmaske (ID 1) verweist auf eine Soft Key Mask (ID 4).

- Wird die Datenmaske gewechselt, wechselt das VT automatisch auch die Softkey-Belegung.
- Über das Kommando Change Soft Key Mask kann die Belegung der Tasten auch zur Laufzeit geändert werden, ohne die Hauptmaske zu wechseln.

20.5.3 Bedeutung für die Implementierung

Das Design der Softkey-Masken ist entscheidend für die Ergonomie. Entwickler sollten darauf achten, dass wichtige Funktionen (z. B. „Zurück“ oder „Home“) immer an der gleichen Position liegen. Durch den Einsatz von NULL-Pointern kann ein Springen der Tasten beim Wechsel zwischen verschiedenen Masken verhindert werden.

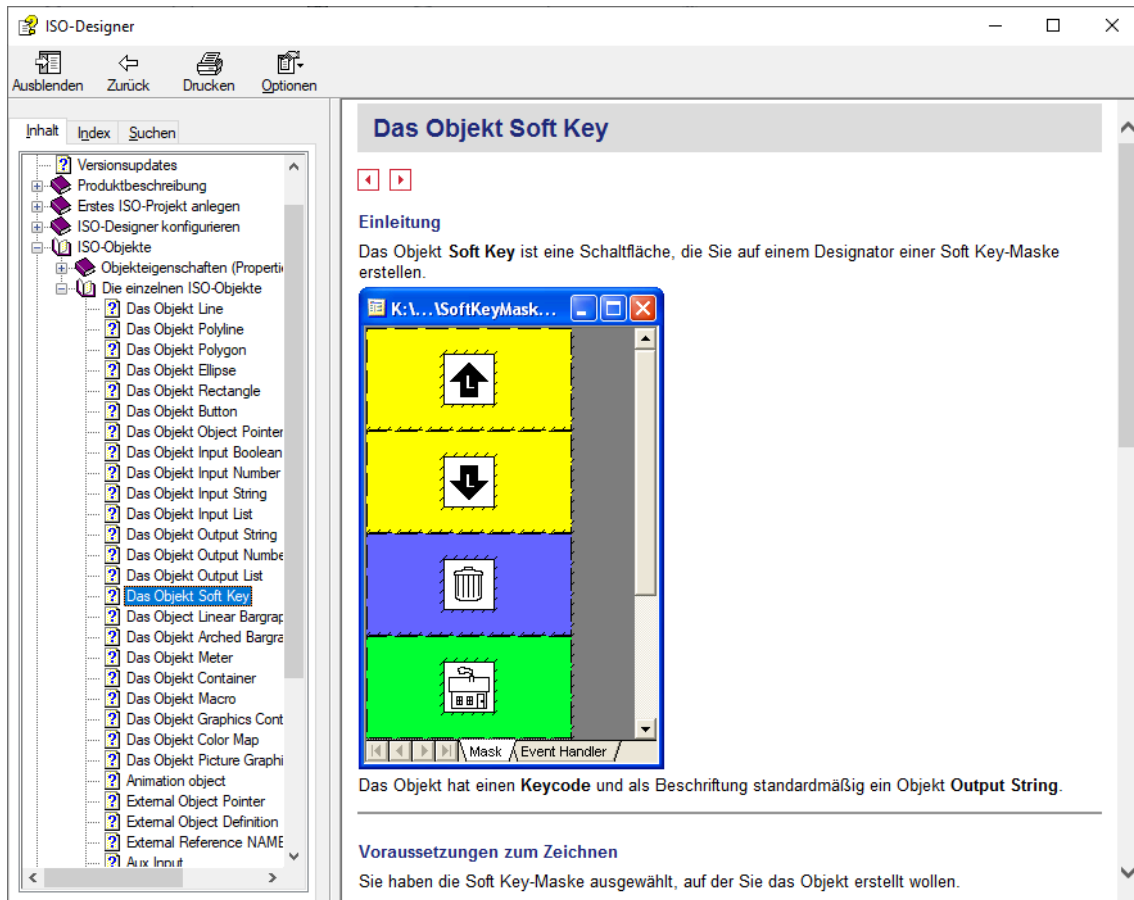
Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Softkey Mask](#) von Tobias Tenberg.

20.5.4 Podcast

- [ISO 11783-6: Softkeys und das Virtual Terminal verstehen – Dein Schlüssel zur Landmaschinen-Mechatronik](#)

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.6 ID 5 – Key (Soft Key) – ISO 11783-6 – B.6



20.6.1 Podcast

- [ISO 11783-6: Softkeys und das Virtual Terminal verstehen – Dein Schlüssel zur Landmaschinen-Mechatronik](#)

Das **Key Objekt** (ID 5) definiert das Aussehen und den funktionalen Code eines Softkeys. Es ist das interaktive Element innerhalb einer Softkey-Maske.

20.6.1.1 Attribute und Record Format (Tabelle B.12)

Die folgende Tabelle beschreibt den Aufbau des Key Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	5	3	Objekttyp = Key.
[1]	Back-ground colour	Integer	1	0 – 255	4	Hintergrund-farbe der Tas-te.
[2]	Key code	Integer	1	1 – 255	5	Code, der in der Soft Key Activation

						Nachricht gesendet wird. (0 ist für ACK reserviert).
-	Number of objects to follow	Integer	1	0 – 255	6	Anzahl der direkt enthaltenen Objekte (Symbole, Texte).
-	Number of macros to follow	Integer	1	0 – 255	7	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	8 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	10 + ...	X-Position relativ zur Taste (Pixel).
-	{Y Location}	Signed Integer	2	-32768 bis +32767	12 + ...	Y-Position relativ zur Taste (Pixel).
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.6.1.2 Designator und Child-Objekte

Ein Key-Objekt fungiert als Container für die grafischen Inhalte der Taste (z. B. Symbole oder Texte).

- **Koordinaten:** X- und Y-Positionen der Child-Objekte beziehen sich auf die obere linke Ecke des Softkey-Designators.
- **Clipping:** Alle Objekte, die außerhalb des physischen Bereichs des Softkeys liegen, werden vom VT abgeschnitten. Da die Softkey-Größen variieren, sollten Inhalte zentral platziert werden.

20.6.2 Ereignisse (Events - Tabelle B.11)

Das Key-Objekt reagiert auf folgende Ereignisse:

- **On Key Press:** Ausgelöst beim Drücken der Taste. Sendet Soft Key Activation.
- **On Key Release:** Ausgelöst beim Loslassen der Taste. Sendet Soft Key Activation.
- **On Change Background Colour:** Reaktion auf Änderung der Tastenfarbe.
- **On Change Child Location / Position:** Aktualisierung der grafischen Inhalte.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.
- **On Input Field Selection / De-selection:** Ausgelöst, wenn die Taste den Fokus erhält oder verliert (bei Navigation via Encoder/Cursor).

20.6.3 Der Key Code

Der **Key Code** (AID 2) ist das entscheidende Bindeglied zur Softwarelogik (C-Code). Während die Object ID die Taste im Grafik-Pool identifiziert, nutzt die Applikation den Key Code, um die Funktion zuzuordnen.

Dies erlaubt es, verschiedene grafische Tasten (unterschiedliche IDs) mit demselben funktionalen Code zu belegen.

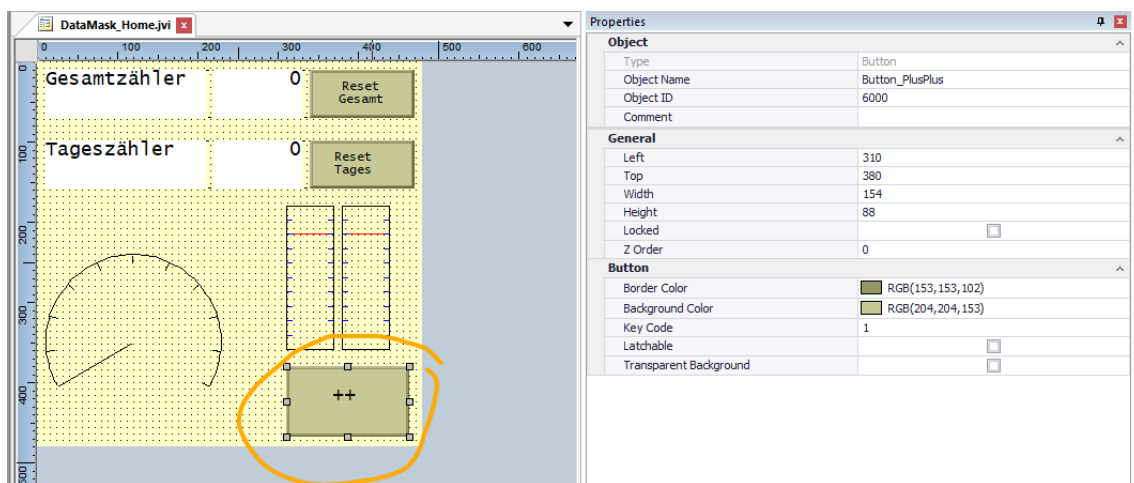
20.6.4 Bedeutung für die Implementierung

Da die tatsächliche Größe und Form der Softkeys von VT zu VT variieren kann (z. B. Hochformat vs. Querformat, Touch vs. Hardwaretasten), ist es Best Practice, grafische Inhalte (Bitmaps) innerhalb des Keys klein genug zu wählen und mittig zu positionieren. Ein Key-Objekt ohne Child-Objekte erscheint als leere Fläche in der gewählten Hintergrundfarbe.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Softkey](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.7 ID 6 – Button – ISO 11783-6 – B.7



Anhang B.7 – Button (Schaltfläche)

Anhang B.7 der ISO 11783-6:2018 widmet sich der detaillierten Definition des „Button“ (Schaltfläche) Objekts im Kontext des Virtuellen Terminals (VT). Schaltflächen sind grundlegende interaktive Elemente der Benutzeroberfläche, die es dem Bediener ermöglichen, Aktionen auszulösen oder Befehle an die landwirtschaftliche Maschine oder das Anbaugerät zu senden.

Überblick über das Button Objekt

Das Button Objekt, wie in B.7 definiert, ist ein grafisches Element, das auf der Datenmaske des VT angezeigt wird. Es ist so konzipiert, dass der Bediener durch Berührung (auf Touchscreens) oder Auswahl mit einem Navigationsmittel (wie z.B. Cursor-Tasten) damit interagieren kann. Jede Schaltfläche ist mit einer bestimmten Funktion oder einem Befehl verknüpft, der ausgeführt wird, wenn die Schaltfläche betätigt wird.

20.7.0.1 Attribute und Record Format (Tabelle B.14)

Die folgende Tabelle beschreibt den Aufbau des Button Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	6	3	Objekttyp = Button.
[1]	Width	Integer	2	0 – 65535	4 – 5	Maximale Breite des

						Button Area in Pixeln.
[2]	Height	Integer	2	0 – 65535	6 – 7	Maximale Höhe des Button Area in Pixeln.
[3]	Back-ground colour	Integer	1	0 – 255	8	Hintergrundfarbe.
[4]	Border colour	Integer	1	0 – 255	9	Farbe des Rahmens (wenn nicht unterdrückt).
[5]	Key Code	Integer	1	0 – 255	10	Code, der in der Button Activation Nachricht gesendet wird.
[6]	Options	Bitmask	1	0 – 63	11	Bit 0: Latchable (0=Tastend, 1=Rastend)Bit 1: State (nur bei Latchable: 0=Gelöst, 1=Gedrückt)Bit 2: Suppress Border (1=Kein Rahmen gezeichnet)Bit 3: Transparent (1=Hintergrund transparent)Bit 4: Disabled (1=Deaktiviert/Ausgegraut)Bit 5: No Border (1=Rahmenfläche entfällt, Button Face = Button Area).
-	Number of objects to follow	Integer	1	0 – 255	12	Anzahl der direkt enthaltenen Objekte (Symbole, Texte).

-	Number of macros to follow	Integer	1	0 – 255	13	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	14 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	16 + ...	X-Position relativ zur oberen linken Ecke der Button Face.
-	{Y Location}	Signed Integer	2	-32768 bis +32767	18 + ...	Y-Position relativ zur oberen linken Ecke der Button Face.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.7.0.2 Aufbau und Darstellung

Der Button besteht aus drei Bereichen:

1. **Button Area:** Die gesamte durch Width/Height definierte Fläche.
2. **Button Border:** Ein proprietärer 8-Pixel-Rahmen (wenn nicht durch Option Bit 5 entfernt).
3. **Button Face:** Die innere Fläche für Inhalte (Button Area minus Border).

20.7.0.3 Container-Struktur

Der Button ist ein Container. Er kann andere Objekte enthalten, die im **Button Face** dargestellt werden. Objekte, die über diesen Bereich hinausragen, werden abgeschnitten (Clipping).

20.7.1 Ereignisse (Events - Tabelle B.13)

Der Button reagiert auf folgende Ereignisse:

- **On Key Press:** Ausgelöst beim Betätigen des Buttons. Sendet Button Activation.
- **On Key Release:** Ausgelöst beim Loslassen. Sendet Button Activation.
- **On Enable:** Wenn der Button per Kommando aktiviert wird.
- **On Disable:** Wenn der Button deaktiviert wird.
- **On Input Field Selection:** Wenn der Button fokussiert wird (Navigation).
- **On Input Field De-selection:** Wenn der Fokus verloren geht.
- **On Change Background Colour:** Reaktion auf Farbänderung.
- **On Change Size:** Reaktion auf Größenänderung (löscht alten Bereich, zeichnet neu).
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

Rolle des Button Objekts im Virtuellen Terminal

Schaltflächen sind entscheidend für die Interaktion des Bedieners mit dem Virtuellen Terminal und den angeschlossenen landwirtschaftlichen Geräten. Sie ermöglichen:

- **Auslösen von Funktionen:** Starten oder Stoppen von Prozessen, Aktivieren von Anbaugeräten, Umschalten von Betriebsmodi.
- **Senden von Befehlen:** Steuern von Aktuatoren, Anpassen von Einstellungen, Navigieren in Menüs.

- **Bestätigung von Eingaben:** Quittieren von Alarmen, Bestätigen von Einstellungen.

Durch die standardisierte Definition des Button Objekts in Anhang B.7 wird sichergestellt, dass Schaltflächen auf verschiedenen Virtuellen Terminals und von verschiedenen Herstellern konsistent funktionieren und dargestellt werden. Dies trägt zur Interoperabilität und Benutzerfreundlichkeit des ISO 11783 Systems bei.

Bedeutung für die Implementierung

Für Entwickler von Virtuellen Terminals, Anbaugeräten oder Softwarekomponenten, die mit dem ISO 11783-Standard arbeiten, ist die genaue Kenntnis der Spezifikationen in Anhang B.7 unerlässlich. Sie müssen sicherstellen, dass ihre Implementierung des Button Objekts den normativen Anforderungen entspricht, um eine korrekte Funktion und Kompatibilität im ISO 11783 Netzwerk zu gewährleisten.

Hinweis

Für die vollständigen und detaillierten Spezifikationen aller Attribute und Eigenschaften des Button Objekts, einschließlich der genauen Datentypen, Wertebereiche und Verknüpfungsmechanismen, wird auf das offizielle Dokument „ISO 11783-6:2018“ verwiesen. Anhang B.7 liefert die normative Grundlage für die Implementierung von Schaltflächen im Virtuellen Terminal.

Auswertung des Buttons:

wird der Button gedrückt so werden folgende Nachrichten am ISOBUS abgesetzt:

- BUTTON_STATE_PRESSED
 - in dem Moment wo der Knopf gedrückt wurde
- BUTTON_STATE_HELD
 - falls der Knopf länger gehalten wurde
 - TODO Verweis auf ISO
- BUTTON_STATE_RELEASED
 - wenn der Knopf losgelassen wurde
- BUTTON_STATE_ABORTED
 - wenn der Knopf gedrückt, aber dann abgebrochen wurde
 - TODO besser beschreiben.

daraus ergibt sich:

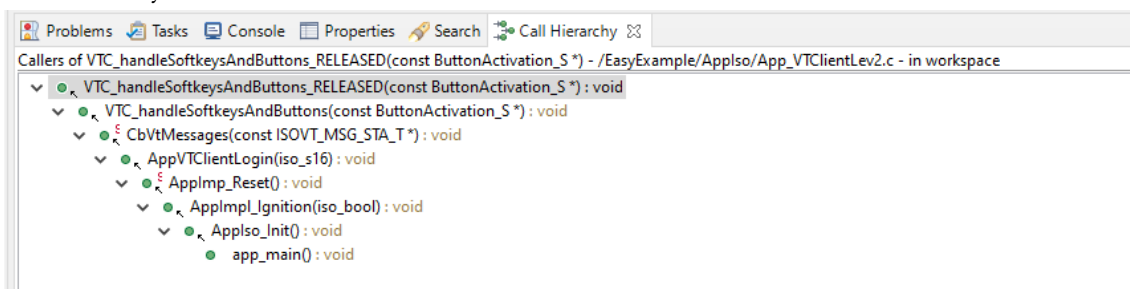
normale „Knopfdrücke“ dürfen nur auf „released“ oder „held“ ausgewertet werden, weil ein auswerten von „pressed“ die Abort Möglichkeit nehmen würde.

TODO: verweis auf internationale Standards, Safety etc..

es gibt 2 Wege der Auswertung:

1. Makro (ISO-Designer)
2. über einen Callback im C-Code (Eclipse)

Call Hierarchy:



```
iso_u32 Tageszaehler = 0;
iso_u32 Gesamtzaehler = 0;
iso_u32 Hugo = 0;
```

```
void VTC_handleSoftkeysAndButtons_RELEASED(const struct ButtonActivation_S *pButtonData)
{
    // what button was released
    switch (pButtonData->objectIdOfButtonObject) {

        case SoftKey_PlusPlus:
        case Button_PlusPlus:
            Tageszaehler++;
            Gesamtzaehler++;
    }
}
```

```

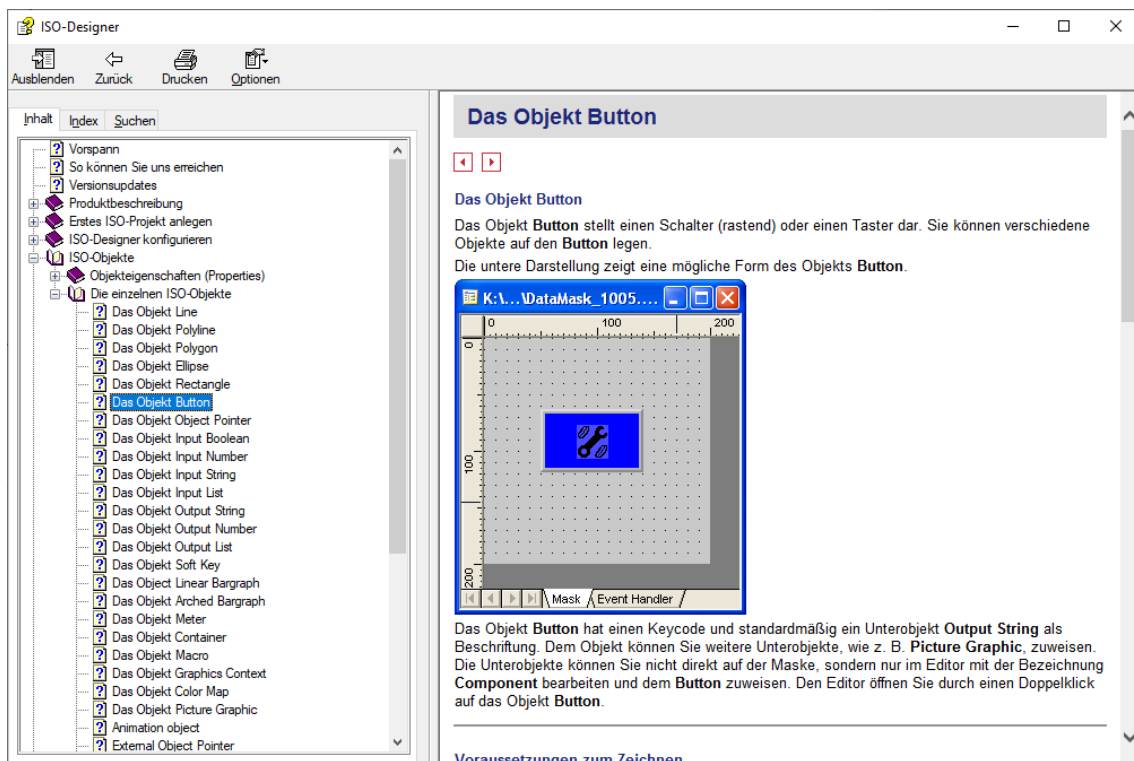
        break;

    case SoftKey_Reset_Gesamtzaehler:
    case Button_Reset_Gesamtzaehler:
        Gesamtzaehler = 0;
        break;

    case SoftKey_Reset_Tageszaehler:
    case Button_Reset_Tageszaehler:
        Tageszaehler = 0;
        break;

    default:
        break;
}
IsoVtcCmd_NumericValue(pButtonData->u8Instance, NumberVariable_Tageszaehler,
Tageszaehler);
IsoVtcCmd_NumericValue(pButtonData->u8Instance, NumberVariable_Gesamtzaehler,
Gesamtzaehler);
setU32("CF-A", "Tageszaehler", Tageszaehler);
setU32("CF-A", "Gesamtzaehler", Gesamtzaehler);
}

```



Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Button](#) von Tobias Tenberg.

20.8 ID 7 – Input boolean – ISO 11783-6 – B.8.2

Das **Input Boolean** Objekt mit der **ID 7** ermöglicht dem Bediener die Eingabe eines TRUE/FALSE-Wertes (z. B. in Form eines Kontrollkästchens).

20.8.0.1 Attribute und Record Format (Tabelle B.16)

Die folgende Tabelle beschreibt den Aufbau des Input Boolean Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.

[0]	Type	Integer	1	7	3	Objekttyp = Input Boolean.
[1]	Background colour	Integer	1	0 – 255	4	Hintergrundfarbe.
[2]	Width	Integer	2	0 – 65535	5 – 6	Breite und Höhe des quadratischen Feldes in Pixeln.
[3]	Foreground colour	Integer	2	0 – 65534	7 – 8	Objekt-ID eines Font Attributes Objekts für die Farbe des Indikators (nur Schriftfarbe relevant).
[4]	Variable reference	Integer	2	0 – 65534, 65535	9 – 10	Objekt-ID einer Number Variable zur Speicherung des Wertes. (65535 = Wert direkt in Attribut 5).
[5]	Value	Integer	1	0, 1 – 255	11	Wert: 0 = FALSE, >0 = TRUE. (Nur genutzt, wenn Variable reference NULL ist).
[6]	Enabled	Integer	1	0 oder 1	12	0 = Deaktiviert, 1 = Aktiviert.
-	Number of macros to follow	Integer	1	0 – 255	13	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.8.0.2 Darstellung

Das VT visualisiert den Booleschen Wert (z. B. als Checkbox).

- **Wert 0:** Hintergrundfarbe wird gezeichnet.
- **Wert > 0:** Indikator wird in Vordergrundfarbe auf Hintergrund gezeichnet.

20.8.1 Ereignisse (Events - Tabelle B.15)

Das Input Boolean Objekt reagiert auf folgende Ereignisse:

- **On Enable:** Wenn das Objekt aktiviert wird.
- **On Disable:** Wenn das Objekt deaktiviert wird.

- **On Input Field Selection:** Bei Fokus/Auswahl durch den Bediener.
- **On Input Field De-selection:** Bei Fokusverlust.
- **On Entry of Value:** Wenn der Bediener einen neuen Wert bestätigt (ENTER). Sendet Change Numeric Value.
- **On Change Value:** Wenn der Wert (z.B. durch Variable) geändert wird.
- **On Change Background Colour:** Reaktion auf Farbänderung.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.8.2 Bedeutung für die Implementierung

Das Input Boolean ist ideal für einfache Ja/Nein-Optionen oder das Aktivieren/Deaktivieren von Maschinenfunktionen. Da die grafische Ausprägung (Häkchen-Stil) vom VT-Hersteller abhängt, sorgt dieses Objekt für ein konsistentes Erscheinungsbild innerhalb der Terminal-Oberfläche.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Boolean](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.9 ID 8 – Input string – ISO 11783-6 – B.8.3

Das **Input String** Objekt mit der **ID 8** dient zur Eingabe und Anzeige von Textzeichenfolgen durch den Bediener.

20.9.0.1 Attribute und Record Format (Tabelle B.17)

Die folgende Tabelle beschreibt den Aufbau des Input String Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	8	3	Objekttyp = Input String.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des Eingabefeldes in Pixeln. Clipping erfolgt außerhalb dieses Bereichs.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des Eingabefeldes in Pixeln. Clipping erfolgt außerhalb dieses Bereichs.
[3]	Background colour	Integer	1	0 – 255	8	Hintergrundfarbe (nur bei deaktivierter Transparenz).
[4]	Font attributes	Integer	2	0 – 65534	9 – 10	Objekt-ID eines Font Attributes Objekts (Farbe, Größe, Font).
[5]	Input attributes	Integer	2	0 – 65534, 65535	11 – 12	Objekt-ID eines Input Attributes Ob-

						jekts zur Validierung oder NULL.
[6]	Options	Bitmask	1	0 – 7	13	Bit 0: TransparentBit 1: Auto-Wrap (Automatischer Zeilenumbruch)Bit 2: Wrap on Hyphen (Umbruch bei Bindestrich).
[7]	Variable reference	Integer	2	0 – 65534, 65535	14 – 15	Verweis auf ein String Variable Objekt. Wenn NULL, wird der Wert direkt im Attribut „Value“ gespeichert.
[8]	Justification	Integer	1	0 – 15	16	Textausrichtung: Bits 0-1 (Horiz.): 0=Links, 1=Mitte, 2=Rechts.Bits 2-3 (Vert.): 0=Oben, 1=Mitte, 2=Unten.
-	Length	Integer	1	0 – 255	17	Max. Länge in Bytes. Wenn Variable Reference != NULL, kann dies 0 sein.
-	Value	String	Length	-	18 ...	Initialer Wert des Strings (nur wenn Variable Reference == NULL).
[9]	Enabled	Integer	1	0 oder 1	var.	0 = Deaktiviert, 1 = Aktiviert. Position im Record ist abhängig von der Länge des Value-Feldes.
-	Number of macros to follow	Integer	1	0 – 255	var.	Anzahl der folgenden Makro-Referenzen.

-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.9.1 Funktionsweise und Optionen

Das Input String Objekt bietet flexible Möglichkeiten zur Textdarstellung:

- **Auto-Wrap:** Wenn aktiviert (Bit 1), bricht das VT den Text automatisch um, wenn die Breite des Feldes überschritten wird.
- **Justierung:** Über AID 8 wird sowohl die horizontale als auch die vertikale Ausrichtung gesteuert.
- **Validierung:** Durch die Verknüpfung mit einem Input Attributes Objekt kann die Eingabe auf bestimmte Zeichensätze begrenzt werden.

20.9.2 Ereignisse (Events - Tabelle B.15)

Das Input String Objekt reagiert auf folgende Ereignisse:

- **On Enable:** Wenn das Objekt aktiviert wird.
- **On Disable:** Wenn das Objekt deaktiviert wird.
- **On Input Field Selection:** Bei Fokus/Auswahl durch den Bediener.
- **On Input Field De-selection:** Bei Fokusverlust.
- **On Entry of Value:** Wenn der Bediener die Texteingabe bestätigt (ENTER). Sendet Change String Value.
- **On Change Value:** Wenn der Wert (z.B. durch Variable) geändert wird.
- **On ESC:** Wenn der Bediener die Eingabe abbricht.
- **On Change Background Colour:** Reaktion auf Farbänderung.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.9.3 Bedeutung für die Implementierung

Input Strings werden häufig für Namen (z. B. Feldnamen, Kundendaten) oder Passwörter verwendet. Da die Texteingabe auf Terminals ohne Tastatur (nur Touch oder Dreh-Drück-Steller) mühsam sein kann, sollten Standardwerte oder Auswahllisten (Input List) bevorzugt werden, wenn der Wertevorrat begrenzt ist.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - String \(Input\)](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.10 ID 9 – Input number – ISO 11783-6 – B.8.4

Das **Input Number** Objekt mit der **ID 9** ist eines der komplexesten und wichtigsten Eingabeobjekte. Es dient zur Eingabe und Anzeige von numerischen Werten und unterstützt automatische Skalierung, Formatierung und Grenzwertprüfung direkt im Terminal.

20.10.0.1 Attribute und Record Format (Tabelle B.18)

Die folgende Tabelle beschreibt den Aufbau des Input Number Objekts im Objekt-pool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.

[0]	Type	Integer	1	9	3	Objekttyp = Input Number.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des Eingabefeldes in Pixeln.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des Eingabefeldes in Pixeln.
[3]	Background colour	Integer	1	0 – 255	8	Hintergrundfarbe.
[4]	Font attributes	Integer	2	0 – 65534	9 – 10	Objekt-ID eines Font Attributes Objekts (Farbe, Größe, Font).
[5]	Options	Bitmask	1	0 – 15	11	Bit 0: Transparent Bit 1: Display leading zeros (Führende Nullen) Bit 2: Display zero as blank (0 als leer) Bit 3: Truncate (1=Abschneiden, 0=Runden).
[6]	Variable reference	Integer	2	0 – 65534, 65535	12 – 13	Verweis auf ein Number Variable Objekt für den Rohwert. Wenn NULL, wird Attribut „Value“ genutzt.
[14]	Value	Integer	4	$0 - 2^{32}-1$	14 – 17	Rohwert (unsigned 32-bit). Nur wenn Variable Reference == NULL.
[7]	Min value	Integer	4	$0 - 2^{32}-1$	18 – 21	Minimaler Rohwert (unsigned).
[8]	Max value	Integer	4	$0 - 2^{32}-1$	22 – 25	Maximaler Rohwert (unsigned).

[9]	Offset	Signed Integer	4	$-2^{31} - 2^{31} - 1$	26 - 29	Offset für die Skalierung.
[10]	Scale	Float	4	-	30 - 33	Skalierungsfaktor.
[11]	Number of decimals	Integer	1	0 - 7	34	Anzahl der Nachkommastellen.
[12]	Format	Boolean	1	0 oder 1	35	0 = Festkomma, 1 = Exponential.
[13]	Justification	Integer	1	0 - 15	36	Textausrichtung: Bits 0-1 (Horiz.), Bits 2-3 (Vert.).
[15]	Options 2	Bitmask	1	0 - 3	37	Bit 0: Enabled (0=Deaktiviert, 1=Aktiviert) Bit 1: Real time editing (1=Wert sofort senden).
-	Number of macros to follow	Integer	1	0 - 255	38	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 - 255	39...	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 - 255	40...	Makro ID des auszuführenden Makros.

20.10.1 Die Skalierungslogik

Das VT berechnet den angezeigten Wert automatisch nach folgender Formel:

$$\text{Angezeigter Wert} = (\text{Rohwert} + \text{Offset}) \times \text{Skalierungsfaktor}$$

Dies erlaubt es, physikalische Werte (z. B. 12,5 bar) als einfache Ganzzahlen im Speicher (z. B. 125) zu verarbeiten, während das VT die Umrechnung und Kommandarstellung übernimmt.

20.10.2 Validierung

Grenzwerte werden ebenfalls auf Basis der skalierten Werte geprüft. Das VT lässt das Schließen des Eingabefeldes (ENTER) nur zu, wenn der neue Wert innerhalb der skalierten Min/Max-Grenzen liegt:
Scaled Min <= Neuer Wert <= Scaled Max

20.10.3 Ereignisse (Events - Tabelle B.15)

Das Input Number Objekt reagiert auf folgende Ereignisse:

- **On Enable / On Disable:** Zustandsänderung des Objekts.
- **On Input Field Selection / De-selection:** Fokus-Ereignisse.
- **On Entry of Value:** Wenn der Bediener einen neuen Wert bestätigt. Sendet Change Numeric Value.
- **On Change Value:** Wenn der Wert durch das Programm geändert wird.

- **On ESC:** Abbruch der Eingabe.
- **On Change Background Colour:** Farbänderung.
- **On Change Attribute:** Allgemeine Attributänderung.

20.10.4 Real Time Editing (AID 15, Bit 1)

Wenn dieses Bit gesetzt ist, sendet das VT bei jeder Änderung (z. B. bei jedem Tastendruck am Inkrementgeber) den aktuellen Zwischenwert an die Arbeitsgruppe. Dies ermöglicht es der Maschine, sofort auf Änderungen zu reagieren (z. B. Drehzahlregelung in Echtzeit), noch bevor der Bediener die Eingabe final bestätigt.

20.10.5 Bedeutung für die Implementierung

Das Input Number Objekt nimmt der Maschinensteuerung (ECU) viel Arbeit bei der Formatierung und Validierung ab. Entwickler sollten darauf achten, den Skalierungsfaktor und die Nachkommastellen so zu wählen, dass keine Rundungsfehler die Anzeige verfälschen.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Number \(Input\)](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.11 ID 10 – Input list – ISO 11783-6 – B.8.5

Das **Input List** Objekt mit der **ID 10** ermöglicht dem Bediener die Auswahl eines Elements aus einer vordefinierten Liste von Objekten. Es wird häufig für Dropdown-Menüs oder Auswahllisten verwendet.

20.11.0.1 Attribute und Record Format (Tabelle B.20)

Die folgende Tabelle beschreibt den Aufbau des Input List Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	10	3	Objekttyp = Input List.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des Feldes (im geschlossenen Zustand).
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des Feldes (im geschlossenen Zustand).
[3]	Variable reference	Integer	2	0 – 65534, 65535	8 – 9	Verweis auf ein Number Variable Objekt für den Index.
[4]	Value	Integer	1	0 – 255	10	Aktueller gewählter Index (0-254). 255 = keine Auswahl. (Nur wenn Variable Reference == NULL).

-	Number of list items	Integer	1	0 – 255	11	Anzahl der Objekte in der Liste.
[5]	Options	Bitmask	1	0 – 3	12	Bit 0: Enabled (0=Deaktiviert, 1=Aktiviert)Bit 1: Real time editing (1=Wert bei Änderung senden).
-	Number of macros to follow	Integer	1	0 – 255	13	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534, 65535	14 + ...	Objekt-ID eines Listeneintrags (angezeigt als Option).
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.11.1 Funktionsweise und Darstellung

Die Input List zeigt im normalen Zustand nur das aktuell ausgewählte Element an.

- **Auswahl:** Wenn der Bediener das Objekt öffnet, zeigt das VT die Liste der verfügbaren Einträge an.
- **Index:** Der übertragene Wert ist der **nullbasierte Index** des gewählten Elements in der Liste.
- **Spezialwert 255:** Signalisiert „keine Auswahl“.

20.11.2 Ereignisse (Events - Tabelle B.19)

Das Input List Objekt reagiert auf folgende Ereignisse:

- **On Enable / On Disable:** Zustandsänderung des Objekts.
- **On Input Field Selection / De-selection:** Fokus-Ereignisse.
- **On Entry of Value:** Wenn der Bediener eine Auswahl bestätigt (ENTER). Sendet Change Numeric Value.
- **On Change Value:** Wenn der Index durch das Programm geändert wird.
- **On Entry of New Value:** Ausgelöst, wenn sich der Wert ändert (oft redundant zu „On Entry of Value“).
- **On ESC:** Abbruch der Auswahl.
- **On Change Attribute:** Allgemeine Attributänderung.
- **On Change Size:** Reaktion auf Größenänderung.

20.11.3 Bedeutung für die Implementierung

Input Lists sind hervorragend geeignet, um Fehleingaben zu vermeiden, da der Bediener nur aus gültigen Optionen wählen kann. Da die Darstellung (z. B. Schriftgröße in der aufgeklappten Liste) vom VT gesteuert wird, ist eine gute Lesbarkeit auf verschiedenen Endgeräten gewährleistet.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - List \(Input\)](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.12 ID 11 – Output string – ISO 11783-6 – B.9.2

Das **Output String** Objekt mit der **ID 11** dient zur rein visuellen Anzeige von Textzeichenfolgen auf dem Virtuellen Terminal. Im Gegensatz zum *Input String* erlaubt dieses Objekt keine direkte Bearbeitung durch den Bediener.

20.12.0.1 Attribute und Record Format (Tabelle B.22)

Die folgende Tabelle beschreibt den Aufbau des Output String Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	11	3	Objekttyp = Output String.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des Textfeldes in Pixeln. Clipping erfolgt außerhalb dieses Bereichs.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des Textfeldes in Pixeln. Clipping erfolgt außerhalb dieses Bereichs.
[3]	Background colour	Integer	1	0 – 255	8	Hintergrundfarbe (nur bei deaktivierter Transparenz).
[4]	Font attributes	Integer	2	0 – 65534	9 – 10	Objekt-ID eines Font Attributes Objekts (Farbe, Größe, Font).
[5]	Options	Bitmask	1	0 – 7	11	Bit 0: TransparentBit 1: Auto-Wrap (Automatischer Zeilenumbruch)Bit 2: Wrap on Hyphen (Umbruch bei Bindestrich).
[6]	Variable reference	Integer	2	0 – 65534, 65535	12 – 13	Verweis auf ein String Variable Objekt. Wenn NULL, wird der Wert direkt im Attri-

						but „Value“ gespeichert.
[7]	Justification	Integer	1	0 – 15	14	Textausrichtung: Bits 0-1 (Horiz.): 0=Links, 1=Mitte, 2=Rechts. Bits 2-3 (Vert.): 0=Oben, 1=Mitte, 2=Unten.
-	Length	Integer	2	0 – 65535	15 – 16	Länge des festen Textwerts in Bytes. Wenn Variable Ref != NULL, kann dies 0 sein.
-	Value	String	Length	-	17...	Statischer Textinhalt (nur wenn Variable Reference == NULL).
-	Number of macros to follow	Integer	1	0 – 255	var.	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.12.1 Funktionsweise und Besonderheiten

- **Transparenz:** Wenn Bit 0 gesetzt ist, ist der Hintergrund des Textfeldes transparent, und die Hintergrundfarbe der Maske oder eines darunterliegenden Objekts bleibt sichtbar.
- **Auto-Wrap:** Ermöglicht die Darstellung mehrzeiliger Texte innerhalb der definierten Width und Height.
- **Justierung:** Die Ausrichtung erfolgt pixelgenau innerhalb des Rahmens. Dies ist besonders wichtig für die vertikale Zentrierung bei unterschiedlichen Schriftarten.
- **Clipping:** Text, der über die definierte Width oder Height hinausgeht, wird vom VT abgeschnitten.

20.12.2 Statischer Text vs. Dynamische Variable

- **Statischer Text:** Der Text wird direkt im Objekt-Pool definiert und kann zur Laufzeit nur über das Kommando Change Attribute (AID 5 oder 7) in seinen Eigenschaften geändert werden.
- **Dynamischer Text:** Durch die Verknüpfung mit einer **String Variable** (AID 6) kann die Steuerung (ECU) den Textinhalt jederzeit per Change String Value aktualisieren, ohne das Objekt selbst neu laden zu müssen.

20.12.3 Ereignisse (Events - Tabelle B.21)

Das Output String Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn sich der angezeigte Wert ändert (z.B. Variable aktualisiert). Das VT zeichnet das Objekt neu.
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss (z.B. Maskenwechsel).
- **On Change Background Colour:** Reaktion auf Farbänderung.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.
- **On Change Size:** Reaktion auf Größenänderung.

20.12.4 Bedeutung für die Implementierung

Output Strings sind die primäre Methode für Statusmeldungen, Beschriftungen und Einheitenanzeigen. Für Texte, die in vielen Sprachen vorliegen, empfiehlt es sich, die Texte über Variablen einzusteuern oder für jede Sprache eine eigene Maske/einen eigenen Pool vorzuhalten.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - String \(Output\)](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.13 ID 12 – Output number – ISO 11783-6 – B.9.3

Das **Output Number** Objekt mit der **ID 12** dient zur Anzeige von numerischen Werten. Wie das *Input Number* Objekt unterstützt es automatische Skalierung und Formatierung, ist jedoch rein für die Anzeige (Read-only für den Bediener) konzipiert.

20.13.0.1 Attribute und Record Format (Tabelle B.23)

Die folgende Tabelle beschreibt den Aufbau des Output Number Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	12	3	Objekttyp = Output Number.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des Anzeigefeldes in Pixeln.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des Anzeigefeldes in Pixeln.
[3]	Background colour	Integer	1	0 – 255	8	Hintergrundfarbe.
[4]	Font attributes	Integer	2	0 – 65534	9 – 10	Objekt-ID eines Font Attributes Objekts.
[5]	Options	Bitmask	1	0 – 15	11	Bit 0: Transparent Bit 1: Display leading zeros Bit 2: Display zeros as blank Bit 3: Truncate (1=Abschnei-

						den, 0=Runden).
[6]	Variable reference	Integer	2	0 – 65534, 65535	12 – 13	Verweis auf ein Number Variable Objekt für den Rohwert.
[12]	Value	Integer	4	$0 - 2^{32}-1$	14 – 17	Rohwert (unsigned 32-bit). Nur wenn Variable Reference == NULL.
[7]	Offset	Signed Integer	4	$-2^{31} - 2^{31}-1$	18 – 21	Offset für die Skalierung.
[8]	Scale	Float	4	-	22 – 25	Skalierungsfaktor.
[9]	Number of decimals	Integer	1	0 – 7	26	Anzahl der Nachkommastellen.
[10]	Format	Boolean	1	0 oder 1	27	0 = Festkomma, 1 = Exponential.
[11]	Justification	Integer	1	0 – 15	28	Textausrichtung: Bits 0-1 (Horiz.), Bits 2-3 (Vert.).
-	Number of macros to follow	Integer	1	0 – 255	29	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.13.1 Die Skalierungslogik

Das VT berechnet den angezeigten Wert automatisch nach folgender Formel:

$$\text{Angezeigter Wert} = (\text{Rohwert} + \text{Offset}) \times \text{Skalierungsfaktor}$$

Beispiel: Ein Rohwert von 2500 mit einem Offset von 0 und einem Skalierungsfaktor von 0.01 wird als **25.00** angezeigt (bei 2 Nachkommastellen).

20.13.2 Formatierungsoptionen

- **Führende Nullen (Bit 1):** Das Feld wird links mit Nullen aufgefüllt (z. B. „0012“).
- **Null als Leerzeichen (Bit 2):** Wenn der skalierte Wert exakt Null ist, bleibt das Feld komplett leer.
- **Runden vs. Abschneiden (Bit 3):** Steuert, wie mit Werten verfahren wird, die mehr Nachkommastellen haben, als in AID 9 definiert sind.

20.13.3 Ereignisse (Events)

Das Output Number Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn sich der angezeigte Wert ändert (z.B. Variable aktualisiert oder Change Numeric Value Befehl). Das VT zeichnet das Objekt neu.
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss.
- **On Change Background Colour:** Reaktion auf Farbänderung.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.
- **On Change Size:** Reaktion auf Größenänderung.

20.13.4 Bedeutung für die Implementierung

Output Numbers sind ideal für die Anzeige von Sensordaten (Druck, Temperatur, Geschwindigkeit). Da die ECU lediglich Ganzzahlen (Rohwerte) übertragen muss, wird die CAN-Bus-Last minimiert und die Komplexität der Formatierung auf das Terminal verlagert. Die pixelgenaue Justierung (AID 11) sorgt dafür, dass die Zahlen auch bei verschiedenen Schriftarten exakt im Designrahmen ausgerichtet sind.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Number \(Output\)](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.14 ID 13 – Output line – ISO 11783-6 – B.10.2

Das **Output Line** Objekt mit der **ID 13** wird verwendet, um eine einfache Linie zwischen zwei Punkten innerhalb eines virtuellen Rechtecks zu zeichnen.

20.14.0.1 Attribute und Record Format (Tabelle B.27)

Die folgende Tabelle beschreibt den Aufbau des Output Line Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	13	3	Objekttyp = Output Line.
[1]	Line attributes	Integer	2	0 – 65534	4 – 5	Objekt-ID eines Line Attributes Objekts (Farbe, Breite, Stil).
[2]	Width	Integer	2	0 – 65535	6 – 7	Breite des umschließenden virtuellen Rechtecks.
[3]	Height	Integer	2	0 – 65535	8 – 9	Höhe des umschließenden virtuellen Rechtecks.
[4]	Line Direction	Integer	1	0 oder 1	10	0 = Oben-Links nach Unten-Rechts 1 = Unten-Links nach Oben-Rechts.
-	Number of macros to follow	Integer	1	0 – 255	11	Anzahl der folgenden Makro-Referenzen.

-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.14.1 Funktionsweise und Geometrie

Die Linie wird innerhalb eines gedachten Rechtecks gespannt, das durch die Position des Objekts sowie Width und Height definiert ist.

- **Line Direction 0:** Die Linie verläuft diagonal fallend.
 - Startpunkt: (X, Y)
 - Endpunkt: (X + Width - Line Width, Y + Height - Line Width)
- **Line Direction 1:** Die Linie verläuft diagonal steigend.
 - Startpunkt: (X, Y + Height - Line Width)
 - Endpunkt: (X + Width - Line Width, Y)

20.14.2 Ereignisse (Events - Tabelle B.26)

Das Output Line Objekt reagiert auf folgende Ereignisse:

- **On Change End Point:** Wird ausgelöst, wenn die Geometrie der Linie geändert wird.
- **On Change Attribute:** Wird ausgelöst, wenn sich die Linien-Eigenschaften (z. B. Farbe) ändern.
- **On Change Size:** Reaktion auf Größenänderung (z. B. durch Change Size Kommando).
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss.

20.14.3 Bedeutung für die Implementierung

Linien werden häufig als Trennelemente in Masken oder zur einfachen grafischen Darstellung von Zusammenhängen genutzt. Durch die Verknüpfung mit Variablen (über die Line Attributes) können Linien zur Laufzeit ihre Farbe ändern, um Zustände (z. B. Aktiv/Inaktiv) zu signalisieren.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Line \(Output\)](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.15 ID 14 – Output rectangle – ISO 11783-6 – B.10.3

Das **Output Rectangle** Objekt mit der **ID 14** dient zum Zeichnen von Rechtecken, die entweder nur als Umriss, gefüllt oder in Kombination dargestellt werden können.

20.15.0.1 Attribute und Record Format (Tabelle B.29)

Die folgende Tabelle beschreibt den Aufbau des Output Rectangle Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	14	3	Objekttyp = Output Rectangle.
[1]	Line attributes	Integer	2	0 – 65534	4 – 5	Objekt-ID eines Line Attributes Objekts für den Rahmen.

[2]	Width	Integer	2	0 – 65535	6 – 7	Breite des Rechtecks in Pixeln.
[3]	Height	Integer	2	0 – 65535	8 – 9	Höhe des Rechtecks in Pixeln.
[4]	Line suppression	Bitmask	1	0 – 15	10	Unterdrückung von Seiten: Bit 0=Oben, Bit 1=Rechts, Bit 2=Unten, Bit 3=Links. (1 = nicht zeichnen).
[5]	Fill attributes	Integer	2	0 – 65534, 65535	11 – 12	Objekt-ID eines Fill Attributes Objekts (für die Füllung) oder NULL für keine Füllung.
-	Number of macros to follow	Integer	1	0 – 255	13	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.15.1 Darstellung und Eigenschaften

Das Rechteck kombiniert Linien- und Fülleigenschaften:

- **Umriss:** Wenn AID 1 verlinkt ist, wird ein Rahmen gemäß den Line Attributes gezeichnet.
- **Füllung:** Wenn AID 5 verlinkt ist, wird das Innere des Rechtecks gemäß den Fill Attributes gefüllt.
- **Linienunterdrückung (AID 4):** Ermöglicht es, gezielt einzelne Kanten des Rechtecks nicht zu zeichnen. Dies ist nützlich für Tabellenstrukturen oder offene Rahmen.
- **Clipping:** Das Rechteck definiert durch Width und Height seine eigenen grafischen Grenzen.

20.15.2 Geometrische Berechnung

Die Ecken des Rechtecks ergeben sich aus der Startposition (StartX, StartY) des Objekts:

- **Ecke Oben-Links:** (StartX, StartY)
- **Ecke Unten-Rechts:** (StartX + Width - 1, StartY + Height - 1) Die Linienstärke (Line Width) muss bei der Planung berücksichtigt werden, da sie je nach VT-Implementierung nach innen oder außen wachsen kann.

20.15.3 Ereignisse (Events - Tabelle B.28)

Das Output Rectangle Objekt reagiert auf folgende Ereignisse:

- **On Change Size:** Wird ausgelöst, wenn die Größe des Rechtecks zur Laufzeit geändert wird.
- **On Change Attribute:** Wird ausgelöst, wenn sich Linien- oder Füllattribute (z. B. Farben) ändern.
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss.

20.15.4 Bedeutung für die Implementierung

Rechtecke sind die am häufigsten verwendeten grafischen Primitiven. Sie dienen als Hintergrund für Textfelder, als Umrandung von Gruppen oder zur Erstellung von Balken (z. B. durch dynamische Änderung der Width oder Height per ECU-Kommando). In Kombination mit transparenten Hintergründen lassen sich so komplexe Layouts realisieren.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Rectangle](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.16 ID 15 – Output ellipse – ISO 11783-6 – B.10.4

Das **Output Ellipse** Objekt mit der **ID 15** dient zum Zeichnen von Kreisen, Ellipsen sowie Kreisbögen, Segmenten und Sektoren (Tortendiagramm-Stücke).

20.16.0.1 Attribute und Record Format (Tabelle B.31)

Die folgende Tabelle beschreibt den Aufbau des Output Ellipse Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	15	3	Objekttyp = Ellipse.
[1]	Line attributes	Integer	2	0 – 65534	4 – 5	Objekt-ID eines Line Attributes Objekts (für den Umriss).
[2]	Width	Integer	2	0 – 65535	6 – 7	Breite des umschließenden virtuellen Rechtecks.
[3]	Height	Integer	2	0 – 65535	8 – 9	Höhe des umschließenden virtuellen Rechtecks.
[4]	Ellipse type	Integer	1	0 – 3	10	0=Closed Ellipse, 1=Open Ellipse (Bogen), 2=Segment (Sehne), 3=Sector (Tortestück).
[5]	Start angle	Integer	1	0 – 180	11	Startwinkel / 2 (in Grad, gegen den Uhrzeigersinn ab positiver X-Achse).

[6]	End angle	Integer	1	0 – 180	12	Endwinkel / 2 (in Grad, gegen den Uhrzeigersinn ab positiver X-Achse).
[7]	Fill attributes	Integer	2	0 – 65534, 65535	13 – 14	Objekt-ID eines Fill Attributes Objekts (für die Füllung) oder NULL für keine Füllung.
-	Number of macros to follow	Integer	1	0 – 255	15	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.16.1 Ellipsentypen und Geometrie

Die Ellipse wird in ein virtuelles Rechteck (Width x Height) eingepasst.

- **Closed Ellipse (0):** Eine vollständige Ellipse oder ein Kreis (wenn Width = Height).
- **Open Ellipse (1):** Nur der Bogen zwischen Start- und Endwinkel wird gezeichnet.
- **Segment (2):** Ein Kreisabschnitt (die Sehne zwischen den Winkelpunkten wird geschlossen).
- **Sektor (3):** Ein Kreisabschnitt (die Winkelpunkte werden mit dem Mittelpunkt verbunden, ideal für Tortendiagramme).

20.16.2 Winkelberechnung (Wichtig!)

Die Winkelwerte in AID 5 und 6 werden **halbiert** übertragen (Bereich 0-180 entspricht 0-360°).

- **90° (Oben):** Wert 45
- **180° (Links):** Wert 90
- **270° (Unten):** Wert 135

Besonderheit bei skalierten Ellipsen: Wenn die Ellipse kein Kreis ist (Width != Height), muss das VT sicherstellen, dass die Winkel mathematisch korrekt gezeichnet werden und nicht nur ein skaliertes Kreisbogen dargestellt wird (siehe ISO-Norm Figure B.8).

20.16.3 Ereignisse (Events - Tabelle B.30)

Das Output Ellipse Objekt reagiert auf folgende Ereignisse:

- **On Change Size:** Wird ausgelöst, wenn die Größe des Rechtecks zur Laufzeit geändert wird.
- **On Change Attribute:** Wird ausgelöst, wenn sich Linien- oder Füllattribute (z. B. Farben) ändern.
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss.

20.16.4 Bedeutung für die Implementierung

Ellipsen und Sektoren sind unverzichtbar für die Erstellung von analogen Zeigerinstrumenten (Meters) oder Fortschrittsanzeigen. Durch die dynamische Änderung des End angle per ECU-Kommando lassen sich füllende Kreisbögen realisieren, die intuitiv Zustände visualisieren.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Ellipse](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.17 ID 16 – Output polygon – ISO 11783-6 – B.10.5

Das **Output Polygon** Objekt mit der **ID 16** ermöglicht das Zeichnen komplexer, mehreckiger Formen durch die Definition einer Liste von Eckpunkten.

20.17.0.1 Attribute und Record Format (Tabelle B.33)

Die folgende Tabelle beschreibt den Aufbau des Output Polygon Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	16	3	Objekttyp = Polygon.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des umschließenden virtuellen Rechtecks.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des umschließenden virtuellen Rechtecks.
[3]	Line attributes	Integer	2	0 – 65534	8 – 9	Objekt-ID eines Line Attributes Objekts (für den Umriss).
[4]	Fill attributes	Integer	2	0 – 65534, 65535	10 – 11	Objekt-ID eines Fill Attributes Objekts (Füllung) oder NULL.
[5]	Polygon type	Integer	1	0 – 3	12	0=Konvex, 1=Nicht-konvex, 2=Komplex, 3=Offen.
-	Number of points	Integer	1	3 – 255	13	Anzahl der Eckpunkte (mindestens 3).
-	Number of macros to follow	Integer	1	0 – 255	14	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Point X}	Integer	2	0 – 65535	15 + ...	X-Position des Punktes relativ zur linken oberen Ecke des Objekts.
-	{Point Y}	Integer	2	0 – 65535	17 + ...	Y-Position des Punktes relativ zur linken oberen Ecke des Objekts.

						ren Ecke des Objekts.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Punkten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.17.1 Polygontypen und Füllregeln

Der Polygontyp (AID 5) informiert das VT über die Komplexität der Form, was die Effizienz des Zeichnens beeinflusst:

- **Convex (0):** Einfache Formen (z. B. Dreieck, Sechseck), bei denen jede horizontale Linie die Kanten nur zweimal schneidet.
- **Non-Convex (1):** Einbuchtungen sind möglich, aber Kanten überschneiden sich nicht.
- **Complex (2):** Kanten dürfen sich überschneiden (z. B. ein Stern). Hier wird die **Even-Odd-Regel** zur Füllung angewendet.
- **Open (3):** Die Punkte werden nur durch Linien verbunden; das Polygon wird nicht geschlossen und nicht gefüllt.

20.17.2 Geometrie und Punkte

- **Relative Position:** Alle Punkte (Point X, Point Y) beziehen sich auf die obere linke Ecke des Polygon-Objekts.
- **Automatisches Schließen:** Wenn der Typ nicht „Open“ ist und der letzte Punkt nicht mit dem ersten identisch ist, schließt das VT das Polygon automatisch durch eine Linie vom letzten zum ersten Punkt.
- **Anzahl der Punkte:** Es können bis zu 255 Punkte definiert werden.

20.17.3 Ereignisse (Events - Tabelle B.32)

Das Output Polygon Objekt reagiert auf folgende Ereignisse:

- **On Refresh:** Wird ausgelöst bei Change Polygon Point oder Change Polygon Scale Kommando.
- **On Change Attribute:** Wird ausgelöst, wenn sich Linien- oder Füllattribute ändern.
- **On Change Size:** Reaktion auf Größenänderung (des umschließenden Rechtecks).

20.17.4 Bedeutung für die Implementierung

Polygone sind sehr mächtig für die Darstellung von unregelmäßigen Flächen, wie z. B. Tankinhalten in asymmetrischen Behältern oder zur Visualisierung von Feldumrissen. Aufgrund der Rechenlast bei komplexen Polygonen sollten ECU-Entwickler nach Möglichkeit „konvexe“ Typen bevorzugen, wenn die Form dies zulässt.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Polygon](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.18 ID 17 – Output meter – ISO 11783-6 – B.11.2

Das **Output Meter** Objekt mit der **ID 17** ist eine Rundanzeige (Zeigerinstrument). Es visualisiert einen Zahlenwert durch die Position einer Nadel auf einem kreisförmigen Bogen.

20.18.0.1 Attribute und Record Format (Tabelle B.35)

Die folgende Tabelle beschreibt den Aufbau des Output Meter Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	17	3	Objekttyp = Output Meter.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite und Höhe des umschließenden Quadrats.
[2]	Needle colour	Integer	1	0 – 255	6	Farbe der Nadel.
[3]	Border colour	Integer	1	0 – 255	7	Farbe des Rahmens (wenn gezeichnet).
[4]	Arc and tick colour	Integer	1	0 – 255	8	Farbe des Bogens und der Ticks.
[5]	Options	Bitmask	1	0 – 15	9	Bit 0: Draw Arc Bit 1: Draw Border Bit 2: Draw Ticks Bit 3: Deflection Direction (0=min->max gegen Uhrz., 1=min->max im Uhrz.).
[6]	Number of ticks	Integer	1	0 – 255	10	Anzahl der Skalenstriche.
[7]	Start angle	Integer	1	0 – 180	11	Startwinkel / 2 (in Grad, gegen den Uhrzeigersinn ab positiver X-Achse).
[8]	End angle	Integer	1	0 – 180	12	Endwinkel / 2 (in Grad, gegen den Uhrzeigersinn ab positiver X-Achse).
[9]	Min value	Integer	2	0 – 65535	13 – 14	Wert am Startwinkel.
[10]	Max value	Integer	2	0 – 65535	15 – 16	Wert am Endwinkel.
[11]	Variable reference	Integer	2	0 – 65534, 65535	17 – 18	Verweis auf ein Number Variable Objekt.
[12]	Value	Integer	2	0 – 65535	19 – 20	Aktueller Wert. Nur wenn Variable

						Reference == NULL.
-	Number of macros to follow	Integer	1	0 – 255	21	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.18.1 Funktionsweise und Darstellung

Das Instrument wird in ein Quadrat eingepasst. Die Nadel bewegt sich auf einem Bogen, der durch Start- und Endwinkel definiert ist.

- **Winkel-Logik:** Wie bei der Ellipse werden Winkel halbiert übertragen (z. B. 45 für 90°).
- **Ticks (AID 6):** Bei zwei oder mehr Ticks wird einer am Anfang und einer am Ende des Bogens gezeichnet; weitere Ticks werden gleichmäßig dazwischen verteilt. Empfohlene Länge: 10 % der Meter-Breite.
- **Transparenz:** Das Meter-Objekt selbst ist transparent. Dadurch können Bitmaps (z. B. ein schönes Zifferblatt) dahinter platziert werden.

20.18.2 Deflektionsrichtung (AID 5, Bit 3)

Dies ist ein kritisches Attribut für die intuitive Bedienung:

- **0 (Anticlockwise):** Der Wert steigt gegen den Uhrzeigersinn.
- **1 (Clockwise):** Der Wert steigt im Uhrzeigersinn (Standard für die meisten analogen Instrumente).

20.18.3 Ereignisse (Events - Tabelle B.34)

Das Output Meter Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn sich der anzuzeigende Wert ändert (z.B. Variable aktualisiert). Das VT bewegt die Nadel.
- **On Change Attribute:** Wird ausgelöst, wenn sich Attribute ändern.
- **On Change Size:** Reaktion auf Größenänderung.
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss.

20.18.4 Bedeutung für die Implementierung

Das Output Meter ist ideal für die Visualisierung von Motordrehzahlen, Füllständen oder Druckwerten. Da es transparent ist, lassen sich durch Kombination mit Hintergrundgrafiken (ID 20) und verschiedenen Masken sehr ansprechende, analog wirkende Cockpit-Anzeigen gestalten.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Meter](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.19 ID 18 – Output linear bar graph – ISO 11783-6 – B.11.3

Das **Output Linear Bar Graph** Objekt mit der **ID 18** dient zur Anzeige von Werten in Form eines Balkens oder Thermometers. Es unterstützt verschiedene Ausrichtungen und bietet die Möglichkeit, einen Zielwert (Target) markant hervorzuheben.

20.19.0.1 Attribute und Record Format (Tabelle B.37)

Die folgende Tabelle beschreibt den Aufbau des Output Linear Bar Graph Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	18	3	Objekttyp = Linear Bar Graph.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des umschließenden Rechtecks in Pixeln.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des umschließenden Rechtecks in Pixeln.
[3]	Colour	Integer	1	0 – 255	8	Farbe der Balkenfüllung und des Rahmens.
[4]	Target line colour	Integer	1	0 – 255	9	Farbe der Zielwert-Linie (wenn gezeichnet).
[5]	Options	Bitmask	1	0 – 63	10	Bit 0: Draw BorderBit 1: Draw Target LineBit 2: Draw TicksBit 3: Type (0=Filled, 1=Line)Bit 4: Axis Orientation (0=Vertical, 1=Horizontal)Bit 5: Direction (0=Negativ/Absteigend, 1=Positiv/Aufsteigend).
[6]	Number of ticks	Integer	1	0 – 255	11	Anzahl der Skalenstriche.
[7]	Min value	Integer	2	0 – 65535	12 – 13	Minimalwert.
[8]	Max value	Integer	2	0 – 65535	14 – 15	Maximalwert.
[9]	Variable reference	Integer	2	0 – 65534, 65535	16 – 17	Verweis auf eine Number Variable für den aktuellen Wert.

[12]	Value	Integer	2	0 – 65535	18 – 19	Aktueller Wert. Nur wenn Variable Reference == NULL.
[10]	Target value var.	Integer	2	0 – 65534, 65535	20 – 21	Verweis auf eine Number Variable für den Zielwert.
[11]	Target value	Integer	2	0 – 65535	22 – 23	Aktueller Zielwert. Nur wenn Target value variable ref == NULL.
-	Number of macros to follow	Integer	1	0 – 255	24	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.19.1 Funktionsweise und Optionen

Der Balkengrafik wird in ein Rechteck eingepasst und ist standardmäßig transparent, sodass Hintergrundgrafiken sichtbar bleiben.

- **Ausrichtung (AID 5, Bit 4 & 5):** Ermöglicht horizontale (links-nach-rechts) oder vertikale (unten-nach-oben) Balken.
- **Darstellungstyp (Bit 3):** Neben dem klassischen füllenden Balken kann das Objekt auch als „Einzeleiger“ fungieren, bei dem nur eine Linie an der aktuellen Position gezeichnet wird.
- **Target Line:** Eine zusätzliche Markierung (z. B. ein roter Strich), die einen Sollwert oder einen Warnbereich kennzeichnet.

20.19.2 Skalierung

Der Balken wird proportional zum aktuellen Value zwischen Min value und Max value berechnet. Liegt der Wert außerhalb dieses Bereichs, wird der Balken entweder komplett leer oder komplett voll gezeichnet.

20.19.3 Ereignisse (Events - Tabelle B.36)

Das Output Linear Bar Graph Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn sich der Value oder der Target value ändert. Das VT aktualisiert die Grafik.
- **On Change Attribute:** Wird ausgelöst, wenn sich Attribute ändern.
- **On Change Size:** Reaktion auf Größenänderung.
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss.

20.19.4 Bedeutung für die Implementierung

Balkendiagramme sind ideal für Füllstandsanzeigen (Kraftstoff, Saatgut), Temperaturanzeigen oder Lastanzeigen. Durch die Option der Target Line kann dem Bediener sofort visualisiert werden, ob er sich im optimalen Arbeitsbereich befindet. Die Kombination mit Skalenstrichen (Ticks) erhöht die Ablesbarkeit.

20.19.5 Podcast

- [ISOBUS-Balkendiagramm: Das Output Linear Bar Graph Objekt der ISO 11783-6 entschlüsselt](#)

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.20 ID 19 – Output arched bar graph – ISO 11783-6 – B.11.4

Das **Output Arched Bar Graph** Objekt mit der **ID 19** ist eine bogenförmige Balkenanzeige. Es kombiniert die Eigenschaften eines linearen Balkendiagramms mit der kreisförmigen Geometrie eines Meter-Objekts.

20.20.0.1 Attribute und Record Format (Tabelle B.39)

Die folgende Tabelle beschreibt den Aufbau des Output Arched Bar Graph Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	19	3	Objekttyp = Arched Bar Graph.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des umschließenden Rechtecks in Pixeln.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des umschließenden Rechtecks in Pixeln.
[3]	Colour	Integer	1	0 – 255	8	Farbe der Balkenfüllung und des Rahmens.
[4]	Target line colour	Integer	1	0 – 255	9	Farbe der Zielwert-Linie (falls gezeichnet).
[5]	Options	Bitmask	1	0 – 31	10	Bit 0: Draw Border Bit 1: Draw Target Line Bit 3: Type (0=Gefüllt, 1=Line/Zeiger) Bit 4: Deflection (0=Gegen Uhrzeigersinn, 1=Im

						Uhrzeiger- sinn).
[6]	Start angle	Integer	1	0 – 180	11	Startwinkel / 2 (in Grad, gegen den Uhrzeiger- sinn ab positi- ver X-Achse).
[7]	End angle	Integer	1	0 – 180	12	Endwinkel / 2 (in Grad, gegen den Uhrzeiger- sinn ab positi- ver X-Achse).
[8]	Bar graph width	Integer	2	0 – 65535	13 – 14	Dicke des Bo- gens in Pixeln.
[9]	Min value	Integer	2	0 – 65535	15 – 16	Minimalwert.
[10]	Max value	Integer	2	0 – 65535	17 – 18	Maximalwert.
[11]	Variable refe- rence	Integer	2	0 – 65534, 65535	19 – 20	Verweis auf eine Number Variable für den aktuellen Wert.
[14]	Value	Integer	2	0 – 65535	21 – 22	Aktueller Roh- wert. Nur wenn Variable Reference == NULL.
[12]	Target value var.	Integer	2	0 – 65534, 65535	23 – 24	Verweis auf ei- ne Number Va- riable für den Zielwert.
[13]	Target value	Integer	2	0 – 65535	25 – 26	Aktueller Ziel- wert. Nur wenn Target value variable ref == NULL.
-	Number of macros to fol- low	Integer	1	0 – 255	27	Anzahl der fol- genden Ma- kro-Referen- zen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro aus- löst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführen- den Makros.

20.20.1 Funktionsweise und Geometrie

Der bogenförmige Balken wird basierend auf einem virtuellen Ellipsenobjekt innerhalb des umschließenden Rechtecks gezeichnet.

- **Balkendicke (AID 8):** Definiert, wie breit der Bogen selbst ist.

- **Winkel-Logik:** Die Winkelwerte werden (wie beim Meter-Objekt) halbiert übertragen (z. B. 45 für 90°).
- **Deflektion (AID 5, Bit 4):** Steuert, ob der Balken im Uhrzeigersinn (Clockwise) oder gegen den Uhrzeigersinn (Anticlockwise) „wächst“.
- **Transparenz:** Das Objekt ist transparent, was die Überlagerung mit Hintergrundbildern ermöglicht.

20.20.2 Ereignisse (Events - Tabelle B.38)

Das Output Arched Bar Graph Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn sich der Value oder der Target value ändert. Das VT aktualisiert die Grafik.
- **On Change Attribute:** Wird ausgelöst, wenn sich Attribute ändern.
- **On Change Size:** Reaktion auf Größenänderung.
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss.

20.20.3 Bedeutung für die Implementierung

Arched Bar Graphs sind ideal für moderne Cockpit-Designs, bei denen mehrere Skalen platzsparend ineinander verschachtelt werden (z. B. Temperatur und Kraftstoff). Durch die Target line kann dem Bediener ein Sollbereich visualisiert werden, während die bogenförmige Form eine intuitive Erfassung des Füllstandes ermöglicht.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Arched Bar Graph](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.21 ID 20 – Picture graphic – ISO 11783-6 – B.12.2

Das **Picture Graphic** Objekt mit der **ID 20** dient zur Anzeige von Rastergrafiken (Bitmaps) auf dem Virtuellen Terminal. Es ermöglicht die Integration von Logos, Icons und komplexen visuellen Elementen.

20.21.0.1 Attribute und Record Format (Tabelle B.41)

Die folgende Tabelle beschreibt den Aufbau des Picture Graphic Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	20	3	Objekttyp = Picture Graphic.
[1]	Width	Integer	2	0 – 65535	4 – 5	Zielbreite in Pixeln (Höhe wird proportional skaliert).
[4]	Actual width	Integer	2	0 – 65535	6 – 7	Tatsächliche Breite der Rohdaten.
[5]	Actual height	Integer	2	0 – 65535	8 – 9	Tatsächliche Höhe der Rohdaten.
[6]	Format	Integer	1	0 – 2	10	0=Monochrom (1 Bit), 1=16 Farben (4 Bit), 2=256 Farben (8 Bit).

[2]	Options	Bitmask	1	0 – 7	11	Bit 0: Transparenz (0=Opak, 1=Transp)Bit 1: BlinkenBit 2: Datenformat (0=Raw, 1=Run-Length Encoded).
[3]	Transparency colour	Integer	1	0 – 255	12	Farbindex, der transparent dargestellt wird.
-	Number of bytes in raw data	Integer	4	$0 - 2^{32}-1$	13 – 16	Größe der Bilddaten.
-	Number of macros to follow	Integer	1	0 – 255	17	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {raw data}	Integer	1	0 – 255	18 ...	Bilddaten (Bytes).
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Bilddaten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.21.1 Funktionsweise und Darstellung

Das Picture Graphic-Objekt speichert Pixelgrafiken in binärer Form innerhalb der Objektpool-Datei (.IOP).

20.21.1.1 Skalierung und Formate

- **Seitenverhältnis:** Das VT skaliert die Grafik basierend auf der Ziel-Breite (width). Um Verzerrungen zu vermeiden, berechnet das VT die Ziel-Höhe automatisch aus dem Verhältnis von Actual width zu Actual height.
- **Farbtiefe:** Unterstützt werden 1-Bit (Monochrom), 4-Bit (16 Farben) und 8-Bit (256 Farben).

20.21.1.2 Transparenz und Effekte

- **Transparency (Bit 0):** Wenn aktiviert, wird die in AID 3 definierte Farbe nicht gezeichnet; stattdessen scheint der Hintergrund durch.
- **Flashing (Bit 1):** Das Bild kann blinken (Frequenz und Stil hängen vom VT ab).
- **RLE-Komprimierung (Bit 2):** Run-Length Encoding kann bei einfachen Grafiken (viele gleichfarbige Flächen) Speicherplatz sparen.

20.21.2 Ereignisse (Events - Tabelle B.40)

Das Picture Graphic Objekt reagiert auf folgende Ereignisse:

- **On Refresh:** Wird ausgelöst bei Änderungen von Optionen (z.B. Transparenz, Blinken) oder bei Masken-Refresh.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.21.3 Bedeutung für die Implementierung

Picture Graphics sind essenziell für ein modernes HMI.

- **Verschwendung vermeiden:** Da Bitmaps viel Speicher im VT belegen, sollten sie so klein wie möglich gehalten werden.
- **Ziederverwendung:** Ein Bild kann im Pool einmal definiert und von vielen Objekten (z. B. mehreren Buttons) referenziert werden.
- **Icons:** Sie werden häufig als „Designatoren“ für Softkeys (ID 5) oder als Symbole in Containern (ID 3) verwendet.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Picture Graphic object](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.22 ID 21 – Number variable – ISO 11783-6 – B.13.2

Das **Number Variable** Objekt mit der ID 21 ist ein reines Datenobjekt. Es speichert einen numerischen Wert, der von anderen Anzeige- oder Eingabeobjekten referenziert werden kann.

20.22.0.1 Attribute und Record Format (Tabelle B.43)

Die folgende Tabelle beschreibt den Aufbau des Number Variable Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	21	3	Objekttyp = Number Variable.
[1]	Value	Integer	4	0 – $2^{32}-1$	4 – 7	32-Bit unsigned Integer-Wert.

20.22.1 Funktionsweise und Referenzierung

Variablen sind keine sichtbaren Objekte. Sie werden niemals direkt in eine Maske oder einen Container als „Child“ eingefügt, sondern dienen als Datenquelle für andere Objekte:

- **Referenzierung:** Objekte wie *Input Number* (ID 9), *Output Number* (ID 12) oder *Output Meter* (ID 17) verweisen über ihr Attribut *Variable reference* auf die ID einer Number Variable.
- **Zentrale Datenhaltung:** Mehrere Anzeigeobjekte können dieselbe Variable referenzieren. Wird der Wert der Variable geändert, aktualisiert das VT automatisch alle betroffenen Anzeigen.

20.22.2 Ereignisse (Events - Tabelle B.42)

Das Number Variable Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn sich der Wert ändert (durch *Change Numeric Value* Befehl oder Eingabe des Bedieners). Das VT zeichnet alle Objekte neu, die diese Variable referenzieren.

20.22.3 Bedeutung für die Implementierung

Number Variables sind das Rückgrat der Kommunikation zwischen Maschine und Terminal.

- **Effizienz:** Anstatt jedes Anzeigeobjekt einzeln zu aktualisieren, ändert die ECU nur den Wert der zentralen Variable.
- **Konsistenz:** Durch die Verwendung von Variablen wird sichergestellt, dass an verschiedenen Stellen der Benutzeroberfläche (z. B. Hauptmaske und Einstellungsmenü) immer derselbe aktuelle Wert angezeigt wird.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Number Variable](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.23 ID 22 – String variable – ISO 11783-6 – B.13.3

Das **String Variable** Objekt mit der **ID 22** dient zur Speicherung von Textzeichenfolgen, die von Anzeige- oder Eingabeobjekten referenziert werden können.

20.23.0.1 Attribute und Record Format (Tabelle B.44)

Die folgende Tabelle beschreibt den Aufbau des String Variable Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	22	3	Objekttyp = String Variable.
-	Length	Integer	2	0 – 65535	4 – 5	Maximale feste Länge der Zeichenkette in Bytes.
-	Value	String	Length	-	6 ...	String aus Zeichen. Muss mit Leerzeichen aufgefüllt werden, um die Länge zu erreichen.

20.23.1 Funktionsweise und Besonderheiten

Wie die *Number Variable* ist auch die *String Variable* ein reines Datenobjekt ohne eigene visuelle Darstellung.

- **Feste Länge:** Die Länge der Variable wird bei der Erstellung im Pool festgelegt und kann zur Laufzeit nicht mehr vergrößert werden.
- **Padding:** Wenn ein kürzerer String als die definierte Länge gespeichert wird, füllt das VT den Rest automatisch mit Leerzeichen auf.
- **Datentypen:** Unterstützt sowohl 8-Bit Zeichen (Standard) als auch WideStrings (für Sonderzeichen). Die Steuerung (ECU) kann den Typ zur Laufzeit zwischen diesen Formaten umschalten.

20.23.2 Referenzierung und Aktualisierung

- **Referenzierung:** Objekte wie *Input String* (ID 8) oder *Output String* (ID 11) verweisen über ihr Attribut *Variable reference* auf die ID einer String Variable.
- **Automatisches Redraw:** Sobald die ECU den Wert der Variable per *Change String Value* Kommando ändert, aktualisiert das VT automatisch alle sichtbaren Objekte, die diese Variable nutzen.

20.23.3 Ereignisse (Events - Tabelle B.42)

Das String Variable Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn sich der Wert ändert (durch *Change String Value* Befehl oder Eingabe des Bedieners). Das VT zeichnet alle Objekte neu, die diese Variable referenzieren.

20.23.4 Bedeutung für die Implementierung

String Variablen sind essenziell für dynamische Texte wie Fehlermeldungen im Klartext, Namen von Arbeitsaufträgen oder Fahrernamen. Da Texteingaben und -änderungen über den CAN-Bus (ISOBUS) ressourcenintensiv sind, sollten String Variablen so kurz wie möglich definiert werden.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - String Variable](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.24 ID 23 – Font attributes – ISO 11783-6 – B.14.2

Das **Font Attributes** Objekt mit der **ID 23** definiert das Erscheinungsbild von Texten (Schriftart, Größe, Farbe, Stil). Es ist ein zentrales Attribut-Objekt, das von allen textbasierten Anzeige- und Eingabeobjekten referenziert wird.

20.24.0.1 Attribute und Record Format (Tabelle B.46)

Die folgende Tabelle beschreibt den Aufbau des Font Attributes Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	23	3	Objekttyp = Font Attributes.
[1]	Font colour	Integer	1	0 – 255	4	Textfarbe.
[2]	Font size	Integer	1	0 – 255	5	Schriftgröße (siehe unten).
[3]	Font type	Integer	1	0 – 255	6	Zeichensatz (siehe ISO-Tabelle K.1).
[4]	Font style	Bitmask	1	0 – 255	7	Bit 0: Bold Bit 1: Crossed Out Bit 2: Underlined Bit 3: Italic Bit 4: Inverted Bit 5: Flashing (Inverted/Style) Bit 6: Flashing (Hidden/Style) Bit 7: Proportional (0=Fixfont, 1=Prop.).
-	Number of macros to follow	Integer	1	0 – 255	8	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.24.1 Schriftgrößen und Render-Modi

Die Interpretation von AID 2 hängt stark von Bit 7 in den Font style Optionen ab:

20.24.1.1 Nicht-proportionale Schriftarten (Bit 7 = 0)

Hier werden vordefinierte Rastergrößen verwendet (Breite x Höhe in Pixeln):

- **0:** 6x8
- **1:** 8x8
- **2:** 8x12
- ...
- **14:** 128x192

20.24.1.2 Proportionale Schriftarten (Bit 7 = 1)

In diesem Modus repräsentiert der Wert in AID 2 direkt die **Schrifthöhe in Pixeln**. Die Breite der einzelnen Zeichen variiert.

20.24.2 Ereignisse (Events - Tabelle B.45)

Das Font Attributes Objekt reagiert auf folgende Ereignisse:

- **On Change Font Attributes:** Wird ausgelöst durch das Kommando Change Font Attributes. Das VT aktualisiert alle Objekte, die diese Attribute verwenden.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.24.3 Bedeutung für die Implementierung

Font Attributes erlauben ein konsistentes Design. Anstatt bei jedem Textobjekt Farbe und Größe einzeln zu definieren, verweisen alle Objekte auf ein gemeinsames Attribut-Objekt. Ändert man dieses eine Objekt (z. B. von weißer auf gelbe Schrift), ändert sich das gesamte HMI-Erscheinungsbild sofort.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Font Attribute](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.25 ID 24 – Line attributes – ISO 11783-6 – B.14.3

Das **Line Attributes** Objekt mit der **ID 24** definiert die grafischen Eigenschaften von Linien und Umrissen (Farbe, Breite, Stil). Es wird von allen geometrischen Objekten wie *Line* (ID 13), *Rectangle* (ID 14), *Ellipse* (ID 15) und *Polygon* (ID 16) referenziert.

20.25.0.1 Attribute und Record Format (Tabelle B.48)

Die folgende Tabelle beschreibt den Aufbau des Line Attributes Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	24	3	Objekttyp = Line Attributes.
[1]	Line colour	Integer	1	0 – 255	4	Linienfarbe.
[2]	Line width	Integer	1	0 – 255	5	Linienstärke in Pixeln. 0 = nicht zeichnen.
[3]	Line art	Bitmask	2	0 – 65535	6 – 7	Bitmuster für Linienstil (1=Zeichnen, 0=Hintergrund).
-	Number of macros to follow	Integer	1	0 – 255	8	Anzahl der folgenden Makro-Referenzen.

-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.25.1 Linienstärken und Darstellung

Das VT verwendet einen quadratischen „Pinsel“ der Größe `Line width` x `Line width`, um die Linie zu zeichnen.

- **Breite = 0:** Die Linie wird nicht gezeichnet.
- **Breite > 1:** Die Linie erscheint dicker.

20.25.2 Linienstil (Line Art - AID 3)

Über eine 16-Bit-Maske wird definiert, ob eine Linie durchgezogen, gestrichelt oder punktiert erscheint:

- Jedes gesetzte Bit (1) steht für einen gezeichneten Pinselstrich.
- Jedes nicht gesetzte Bit (0) steht für eine Lücke (Hintergrund scheint durch).
- **Beispiel 0xFFFF:** Durchgezogene Linie (alle Bits 1).
- **Beispiel 0xCCCC (11001100...):** Gestrichelte Linie.
- **Besonderheit:** Die Länge eines Strichs skaliert mit der `Line width`.

20.25.3 Ereignisse (Events - Tabelle B.47)

Das `Line Attributes` Objekt reagiert auf folgende Ereignisse:

- **On Change Line Attributes:** Wird ausgelöst durch das Kommando `Change Line Attributes`. Das VT aktualisiert alle Objekte, die dieses Attribut verwenden.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.25.4 Bedeutung für die Implementierung

`Line Attributes` ermöglichen eine effiziente Steuerung der grafischen Darstellung. Durch das Ändern eines einzigen Attribut-Objekts können beispielsweise alle Umrandungen in einer Maske gleichzeitig von „Dünn/Schwarz“ auf „Dick/Rot“ umgeschaltet werden, um einen Alarmzustand zu visualisieren.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Line Attribute](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.26 ID 25 – Fill attributes – ISO 11783-6 – B.14.4

Das `Fill Attributes` Objekt mit der `ID 25` definiert, wie geschlossene geometrische Formen (Rechtecke, Ellipsen, Polygone) gefüllt werden.

20.26.0.1 Attribute und Record Format (Tabelle B.50)

Die folgende Tabelle beschreibt den Aufbau des `Fill Attributes` Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	25	3	Objekttyp = Fill Attributes.
[1]	Fill type	Integer	1	0 – 3	4	0=Keine Füllung, 1=Linien-

						farbe, 2=Füllfarbe, 3=Muster.
[2]	Fill colour	Integer	1	0 – 255	5	Füllfarbe (nur bei Typ 2 relevant).
[3]	Fill pattern	Integer	2	0 – 65534, 65535	6 – 7	Objekt-ID eines Picture Graphic Objekts für Musterfüllung (nur bei Typ 3).
-	Number of macros to follow	Integer	1	0 – 255	8	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.26.1 Fülltypen und Logik

Über AID 1 wird gesteuert, welche Quelle für die Flächenfüllung genutzt wird:

- **No Fill (0):** Die Fläche bleibt transparent; nur der Umriss (falls definiert) wird gezeichnet.
- **Line Colour (1):** Die Fläche wird in derselben Farbe gefüllt wie die Umrandung (aus den Line Attributes).
- **Specified Colour (2):** Es wird die in AID 2 definierte Farbe verwendet.
- **Pattern (3):** Die Fläche wird mit einer sich wiederholenden Grafik gefüllt (Kachelung).

20.26.2 Verwendung von Füllmustern (Wichtig!)

Wenn ein Muster (AID 3) verwendet wird, gelten strenge Regeln für die referenzierte Grafik:

- **Ausrichtung:** Bei monochromen Grafiken muss die Breite durch 8 teilbar sein. Bei 16-Farben-Grafiken muss sie durch 2 teilbar sein.
- **Reihenfolge:** Bei dynamischen Änderungen muss erst das `Fill pattern` und dann der `Fill type` gesetzt werden, um Fehler im VT zu vermeiden.

20.26.3 Ereignisse (Events - Tabelle B.49)

Das Fill Attributes Objekt reagiert auf folgende Ereignisse:

- **On Change Fill Attributes:** Wird ausgelöst durch das Kommando `Change Fill Attributes`. Das VT aktualisiert alle Objekte, die dieses Attribut verwenden.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.26.4 Bedeutung für die Implementierung

Fill Attributes sind unverzichtbar für die Gestaltung der Benutzeroberfläche. Sie ermöglichen es, Flächen hervorzuheben (z. B. gelbe Füllung für einen Warnbereich in einem Balkendiagramm) oder texturierte Hintergründe zu schaffen. Durch den Wechsel des Fülltyps zur Laufzeit können Zustände (z. B. „Tank leer“ -> rot blinkende Füllung) sehr auffällig visualisiert werden.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Fill Attribute](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.27 ID 26 – Input attributes – ISO 11783-6 – B.14.5

Das **Input Attributes** Objekt mit der **ID 26** dient zur Validierung von Texteingaben. Es legt fest, welche Zeichen ein Bediener in ein verknüpftes *Input String* Objekt eingeben darf.

20.27.0.1 Attribute und Record Format (Tabelle B.52)

Die folgende Tabelle beschreibt den Aufbau des Input Attributes Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	26	3	Objekttyp = Input Attributes.
[1]	Validation type	Integer	1	0 – 1	4	0=Erlaubte Zeichen (Liste), 1=Verbotene Zeichen (Liste).
-	Length	Integer	1	0 – 255	5	Länge des Validierungs-Strings in Bytes.
-	Validation string	String	Length	-	6 ...	Liste der Zeichen (8-Bit String).
-	Number of macros to follow	Integer	1	0 – 255	var.	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.27.1 Funktionsweise und Validierung

Das Objekt wirkt als Filter für die Tastatureingabe am VT:

- **Referenzierung:** Ein *Input String* Objekt (ID 8) verweist auf dieses Objekt.
- **Filter-Logik:** Wenn der *Validation type* auf 0 steht, lässt das VT nur die Zeichen zu, die im *Validation string* enthalten sind. Steht er auf 1, werden alle Zeichen außer den gelisteten akzeptiert.
- **Einschränkung:** Dieses Objekt unterstützt ausschließlich **8-Bit Strings**. Wenn das verknüpfte Eingabefeld einen *WideString* verwendet, findet keine Validierung statt.

20.27.2 Ereignisse (Events - Tabelle B.51)

Das Input Attributes Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst durch das Kommando *Change String Value*. Das VT aktualisiert den Validierungs-String.

20.27.3 Bedeutung für die Implementierung

Input Attributes sind ein wichtiges Werkzeug zur Vermeidung von Fehlbedienungen.

- **Beispiel Numerisch:** Ein Validierungsstring „0123456789.“ begrenzt ein Textfeld auf rein numerische Zeichen.
- **Beispiel Sonderzeichen:** Verbot von Zeichen wie „;“ oder „““, die in Datenbanken oder Dateisystemen Probleme verursachen könnten.

20.27.3.1 Hinweis: Extended Input Attributes (ID 38)

Für die Validierung von WideStrings (Unicode) muss das *Extended Input Attributes* Objekt verwendet werden, welches die Definition ganzer Code-Bereiche (Code Planes) erlaubt.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.28 ID 27 – Object pointer – ISO 11783-6 – B.15

Das **Object Pointer** Objekt mit der **ID 27** dient als dynamischer Platzhalter. Es ermöglicht es, an einer fest definierten Stelle in einer Maske oder einem Container das referenzierte Objekt zur Laufzeit auszutauschen.

20.28.0.1 Attribute und Record Format (Tabelle B.55)

Die folgende Tabelle beschreibt den Aufbau des Object Pointer Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	27	3	Objekttyp = Object Pointer.
[1]	Value	Integer	2	0 – 65534, 65535	4 – 5	Objekt-ID des referenzierten Objekts oder NULL (65535).

20.28.1 Funktionsweise und Anwendung

Ein Object Pointer wird wie ein normales Child-Objekt in eine Maske eingebunden. Anstatt jedoch selbst etwas zu zeichnen, „leitet“ er die Anzeige an das Objekt weiter, dessen ID in AID 1 gespeichert ist.

- **Dynamischer Austausch:** Die ECU kann den Value (AID 1) per Change Numeric Value Kommando jederzeit ändern. Das VT blendet daraufhin das alte Objekt aus und das neue an derselben Stelle ein.
- **Platzhalter-Funktion:** Er eignet sich hervorragend für Status-Icons (z. B. wechselnde Symbole für verschiedene Maschinenzustände), ohne dass mehrere Objekte übereinandergelegt und einzeln versteckt werden müssen.
- **NULL-Pointer:** Wird der Wert auf 65535 gesetzt, wird an dieser Stelle nichts gezeichnet.

20.28.2 Ereignisse (Events - Tabelle B.54)

Das Object Pointer Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst durch das Kommando Change Numeric Value. Das VT versteckt das vorherige Objekt und zeigt das neue an. Die Eltern-Maske wird aktualisiert.

20.28.3 Bedeutung für die Implementierung

Object Pointer reduzieren die Komplexität der Maskensteuerung erheblich. Anstatt viele Objekte manuell per Hide/Show zu verwalten, muss die ECU nur eine einzige ID im Pointer ändern. Dies spart CAN-Bus-Bandbreite und vereinfacht die Programmlogik auf der Maschinensteuerung.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Object Pointer](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen zu Datentypen und Nachrichtenformaten wird auf die offizielle ISO 11783-6:2018 verwiesen.

20.29 ID 28 – Macro – ISO 11783-6 – B.16

Das **Macro** Objekt mit der **ID 28** erlaubt es, eine Sequenz von Befehlen im Virtuellen Terminal zu speichern und bei bestimmten Ereignissen (Events) automatisch auszuführen. Dies reduziert die notwendige Kommunikation über den ISOBUS, da einfache UI-Logik direkt im Terminal abläuft.

20.29.0.1 Attribute und Record Format (Tabelle B.56)

Die folgende Tabelle beschreibt den Aufbau des Macro Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 255 (VT v4) 0 – 65534 (VT v5+)	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	28	3	Objekttyp = Macro.
-	Number of bytes to follow	Integer	2	0 – 65535	4 – 5	Anzahl der Bytes für die Befehlsliste.
-	Repeat: {Command}	Binary	6 – n	-	6 ...	Liste von Befehlspaketen. Jeder Befehl muss ein Vielfaches von 8 Bytes lang sein (Padding mit FFh).

20.29.1 Funktionsweise und Struktur

Ein Makro besteht aus einer Liste von VT-Kommandos (siehe ISO 11783-6, Anhang F).

- **Padding:** Jeder Befehl innerhalb eines Makros muss auf eine Länge von **8 Byte** aufgefüllt werden (mit 0xFF), falls das eigentliche Kommando kürzer ist (z.B. Change Numeric Value).
- **Ausführung:** Makros können durch Events (z. B. On Press eines Buttons) oder durch das Kommando Execute Macro von der ECU gestartet werden.
- **Konsistenz:** Die ECU ist dafür verantwortlich, dass Makros nur auf Objekte verweisen, die tatsächlich im Pool existieren.

20.29.2 Verfügbare Makro-Befehle (Auszug)

Makros können fast alle kommandierenden VT-Funktionen nutzen:

- **Sichtbarkeit:** Hide/Show Object (Ein-/Ausblenden von Containern).
- **Interaktion:** Enable/Disable Object (Sperren von Buttons/Eingaben), Select Input Object (Fokus setzen).
- **Werte:** Change Numeric Value (Variablen oder Pointer ändern), Change String Value.
- **Geometrie:** Change Child Location/Position (Objekte verschieben/scrollen), Change Size, Change End Point.
- **Darstellung:** Change Background Color, Change Font/Line/Fill Attributes.
- **Navigation:** Change Active Mask (Maskenwechsel), Change Soft Key Mask.
- **Audio:** Control Audio Device (Signaltöne ausgeben).
- **Listen:** Change List Item (Inhalt von Input-Listen ändern).

20.29.3 Ereignisse (Events)

Makros lösen selbst keine Events aus, werden aber durch Events anderer Objekte gestartet. Das Makro Objekt unterstützt die Kommandos:

- Execute Macro
- Execute Extended Macro
- Get Attribute Value

20.29.4 Bedeutung für die Implementierung

Makros sind ein mächtiges Werkzeug zur **Performance-Optimierung**:

1. **Reaktionszeit:** Ein Maskenwechsel direkt nach einem Tastendruck erfolgt per Makro ohne CAN-Verzögerung.
2. **Entlastung:** Die ECU muss sich nicht um rein grafische Belange kümmern (z. B. das Umschalten eines Icons beim Drücken eines Buttons).
3. **Komplexität:** Mehrere Aktionen können in ein einziges Makro gepackt werden (z. B. „Variable auf 0 setzen“ UND „Erfolgsmeldung einblenden“ UND „Ton abspielen“).

Hinweis: Für detaillierte Spezifikationen zu den einzelnen Befehlscodes wird auf die offizielle ISO 11783-6:2018, Anhang F verwiesen.

20.30 ID 29 – Auxiliary Function Type 1 – ISO 11783-6 – J.4.2

Das **Auxiliary Function Type 1** Objekt mit der **ID 29** definiert die Attribute und das Design eines Hilfsfunktions-Bedienelements (Auxiliary Function).

Hinweis: Dieses Objekt gilt ab VT Version 3 als veraltet (obsolete) und wird durch den Typ 2 (ID 31) ersetzt. Es sollte in neuen Projekten nicht mehr verwendet werden.

20.30.0.1 Attribute und Record Format (Tabelle J.1)

Die folgende Tabelle beschreibt den Aufbau des Auxiliary Function Type 1 Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	29	3	Objekttyp = Auxiliary Function Type 1.
[1]	Background colour	Integer	1	0 – 255	4	Hintergrundfarbe.
[2]	Function type	Integer	1	0 – 2	5	0=Schaltend (Latch Boolean), 1=Analog, 2=Tastend (Non-latching Boolean).
-	Number of objects to follow	Integer	1	1 – 255	6	Anzahl der direkt enthaltenen Objekte.

						te (Designator).
-	Repeat: {Object ID}	Integer	2	0 – 65534	7 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	9 + ...	X-Position relativ zur oberen linken Ecke.
-	{Y Location}	Signed Integer	2	-32768 bis +32767	11 + ...	Y-Position relativ zur oberen linken Ecke.

20.30.1 Funktionsweise

Das VT nutzt diese Attribute, um die Zuweisung zu einem kompatiblen Auxiliary Input zu erzwingen. Das Designator-Symbol muss in den Bereich eines Softkeys passen; überstehende Teile werden abgeschnitten (Clipped). Das Objekt dient als Funktionsdefinition, die einem physischen Hilfseingabegerät (Auxiliary Input) zugewiesen werden kann.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, Anhang J verwiesen.

20.31 ID 30 – Auxiliary Input Type 1 – ISO 11783-6 – J.4.4

Das **Auxiliary Input Type 1** Objekt mit der **ID 30** definiert die Eigenschaften eines physischen Bedienelements (z. B. Joystick, Schalter) eines Auxiliary Input Geräts.

Hinweis: Dieses Objekt gilt ab VT Version 3 als veraltet (obsolete) und wird durch den Typ 2 (ID 32) ersetzt.

20.31.0.1 Attribute und Record Format (Tabelle J.3)

Die folgende Tabelle beschreibt den Aufbau des Auxiliary Input Type 1 Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	30	3	Objekttyp = Auxiliary Input Type 1.
[1]	Background colour	Integer	1	0 – 255	4	Hintergrundfarbe.
[2]	Function type	Integer	1	0 – 2	5	0=Schaltend (Latch Boolean), 1=Analog, 2=Tastend (Non-latching Boolean).
[3]	Input ID	Integer	1	0 – 255	6	ID des physischen

						Eingangs, auf den sich dieses Objekt bezieht.
-	Number of objects to follow	Integer	1	1 – 255	7	Anzahl der direkt enthaltenen Objekte (Designator).
-	Repeat: {Object ID}	Integer	2	0 – 65534	8 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	10 + ...	X-Position relativ zur oberen linken Ecke.
-	{Y Location}	Signed Integer	2	-32768 bis +32767	12 + ...	Y-Position relativ zur oberen linken Ecke.

20.31.1 Funktionsweise

Dieses Objekt beschreibt ein physisches Eingabeelement (z. B. einen Taster oder eine Achse eines Joysticks) an einem Auxiliary Input Device. Es wird vom Auxiliary Input Device bereitgestellt, damit das VT die verfügbaren Eingänge und deren Eigenschaften kennt. Die Input ID verknüpft dieses Objekt mit den Statusnachrichten des Geräts.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, Anhang J verwiesen.

20.32 ID 31 – Auxiliary Function Type 2 – ISO 11783-6 – J.4.3

Das **Auxiliary Function Type 2** Objekt mit der **ID 31** ist die moderne Definition für Hilfsfunktionen im ISOBUS (ab VT Version 3). Es bietet erweiterte Möglichkeiten zur Zuweisung von Bedienelementen.

20.32.0.1 Attribute und Record Format (Tabelle J.2)

Die folgende Tabelle beschreibt den Aufbau des Auxiliary Function Type 2 Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	31	3	Objekttyp = Auxiliary Function Type 2.
[1]	Back-ground colour	Integer	1	0 – 255	4	Hintergrundfarbe.
[2]	Function attributes	Bitmask	1	-	5	Bitmaske für Funktionstyp und Steue-

						rung (siehe unten).
-	Number of objects to follow	Integer	1	1 – 255	6	Anzahl der direkt enthaltenen Objekte (Designator).
-	Repeat: {Object ID}	Integer	2	0 – 65534	7 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	9 + ...	X-Position relativ zur oberen linken Ecke.
-	{Y Location}	Signed Integer	2	-32768 bis +32767	11 + ...	Y-Position relativ zur oberen linken Ecke.

20.32.0.2 Function Attributes (Bitmaske AID 2)

- **Bit 0–3:** Auxiliary function type (siehe Tabelle J.5)
 - 0: Boolean Latching (Schaltend)
 - 1: Analogue (Analog)
 - 2: Boolean Momentary (Tastend)
 - 3: Boolean Latching (Dual)
 - 4: Analogue (Dual)
 - 5: Boolean Momentary (Dual)
 - ...
- **Bit 4–5:** Reserved
- **Bit 6:** Assignment Restriction (0 = Frei, 1 = Eingeschränkt, siehe ISO 11783-6).
- **Bit 7:** Single Assignment (1 = Darf nur alleine auf einem Input liegen).

20.32.1 Besonderheiten

Dieses Objekt wird aktiv für das „Auxiliary Mapping“ im Terminal genutzt. Es erlaubt der ECU, dem Terminal mitzuteilen, welche Funktionen (z. B. „Heber auf/ab“) zur Verfügung stehen, damit der Benutzer diese auf Tasten oder Joysticks legen kann. Es ersetzt den veralteten Typ 1 (ID 29).

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, Anhang J verwiesen.

20.33 ID 32 – Auxiliary Input Type 2 – ISO 11783-6 – J.4.5

Das **Auxiliary Input Type 2** Objekt mit der **ID 32** definiert ein physisches Bedienelement eines Auxiliary Input Geräts (z. B. eine Taste auf einem Joystick) ab VT Version 3.

20.33.0.1 Attribute und Record Format (Tabelle J.4)

Die folgende Tabelle beschreibt den Aufbau des Auxiliary Input Type 2 Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.

[0]	Type	Integer	1	32	3	Objekttyp = Auxiliary Input Type 2.
[1]	Background colour	Integer	1	0 – 255	4	Hintergrundfarbe.
[2]	Function attributes	Bitmask	1	-	5	Bitmaske für Funktionstyp und Steuerung (siehe unten).
[3]	Input ID	Integer	1	0 – 255	6	ID des physikalischen Eingangs.
-	Number of objects to follow	Integer	1	1 – 255	7	Anzahl der direkt enthaltenen Objekte (Designator).
-	Repeat: {Object ID}	Integer	2	0 – 65534	8 + ...	Objekt-ID eines enthaltenen Objekts.
-	{X Location}	Signed Integer	2	-32768 bis +32767	10 + ...	X-Position relativ zur oberen linken Ecke.
-	{Y Location}	Signed Integer	2	-32768 bis +32767	12 + ...	Y-Position relativ zur oberen linken Ecke.

20.33.0.2 Function Attributes (Bitmaske AID 2)

- **Bit 0–3:** Auxiliary function type (siehe Tabelle J.5)
 - 0: Boolean Latching (Schaltend)
 - 1: Analogue (Analog)
 - 2: Boolean Momentary (Tastend)
 - 3: Boolean Latching (Dual)
 - ...
- **Bit 4–5:** Reserved
- **Bit 6:** Reserved
- **Bit 7:** Single Assignment (1 = Darf nur einer Funktion zugewiesen werden).

20.33.1 Funktionsweise

Das Terminal nutzt diese Informationen, um dem Benutzer die verfügbaren Tasten und deren physikalische Eigenschaften anzuzeigen. Wenn der Benutzer eine Taste drückt, sendet das Eingabegerät eine Statusmeldung mit dem aktuellen Wert (Boolean oder Analog) und der Input ID an den ISOBUS. Es ersetzt den veralteten Typ 1 (ID 30).

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, Anhang J verwiesen.

20.34 ID 33 – Auxiliary Control Designator Type 2 Object Pointer – ISO 11783-6 – J.4.7

Das **Auxiliary Control Designator Type 2 Object Pointer** Objekt mit der **ID 33** ermöglicht es einer Working Set, die aktuell zugewiesenen Hilfsfunktionen (Auxiliary Functions) und deren Bedienelemente (Inputs) grafisch in einer Maske darzustellen.

20.34.0.1 Attribute und Record Format (Tabelle J.6)

Die folgende Tabelle beschreibt den Aufbau des Auxiliary Control Designator Type 2 Objekts im Objekt-pool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	33	3	Objekttyp = Auxiliary Control Designator Type 2 Object Pointer.
[1]	Pointer Type	Integer	1	0 – 3	4	Typ der Referenz (siehe unten).
[2]	Auxiliary Object ID	Integer	2	0 – 65534, 65535	5 – 6	Objekt-ID einer Auxiliary Function (Type 2) oder eines Auxiliary Inputs (Type 2).

20.34.0.2 Pointer Types (AID 1)

- **0:** Das VT zeigt den Designator des durch Auxiliary Object ID angegebenen Objekts.
- **1:** Das VT zeigt den Designator des Objekts, das dem durch Auxiliary Object ID referenzierten Objekt zugewiesen ist (z. B. wenn AID 2 eine Funktion ist, wird der zugewiesene Input gezeigt).
- **2:** Das VT zeigt den Designator des Working Sets, das Besitzer des durch Auxiliary Object ID angegebenen Objekts ist.
- **3:** Das VT zeigt den Designator des Working Sets, das Besitzer des Objekts ist, welches dem durch Auxiliary Object ID referenzierten Objekt zugewiesen ist.

20.34.1 Ereignisse (Events - Tabelle J.7)

Das Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn die Zuweisung (Mapping) geändert wird oder das referenzierte Objekt geändert wird. Das VT aktualisiert die Anzeige.

20.34.2 Nutzen für den Entwickler

Mit diesem Objekt kann eine grafische Übersicht erstellt werden, die dem Benutzer zeigt: „Diese Funktion auf meiner Maschine wird aktuell von Taste Y am Joystick Z gesteuert“. Da die Zuweisung (Mapping) oft durch den Benutzer am Terminal erfolgt, erlaubt dieser Pointer eine dynamische Anzeige ohne Vorwissen der ECU über die tatsächliche Belegung.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, Anhang J verwiesen.

20.35 ID 34 – Window Mask – ISO 11783-6 – B.19

Das **Window Mask** Objekt mit der **ID 34** (eingeführt mit VT Version 4) ermöglicht es, einen Teilbereich des Bildschirms zu definieren, der unabhängig von der Haupt-Datenmaske aktualisiert oder von anderen Working Sets mit Inhalten gefüllt werden kann.

20.35.0.1 Attribute und Record Format (Tabelle B.61)

Die folgende Tabelle beschreibt den Aufbau des Window Mask Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	34	3	Objekttyp = Window Mask.
-	Width	Integer	1	1 – 2	4	Breite in User-Layout-Spalten (nur für Typ 0 Free Form relevant, sonst ignoriert).
-	Height	Integer	1	1 – 6	5	Höhe in User-Layout-Zeilen (nur für Typ 0 Free Form relevant, sonst ignoriert).
-	Window Type	Integer	1	0 – 18	6	Typ des Fensters (siehe unten).
[1]	Back-ground colour	Integer	1	0 – 255	7	Hintergrundfarbe (nur für Typ 0 relevant).
[2]	Options	Bitmask	1	0 – 3	8	Bit 0: Available (0=Nicht verfügbar/geblankt, 1=Verfügbar) Bit 1: Transparent (1=Hintergrund transparent, nur Typ 0).
[3]	Name	Integer	2	0 – 65534	9 – 10	Objekt-ID eines Output String (Name

						für Mapping-Screen).
-	Window Title	Integer	2	0 – 65534, 65535	11 – 12	Objekt-ID eines Output String (Titel im Fenster).
-	Window Icon	Integer	2	0 – 65534, 65535	13 – 14	Objekt-ID eines Output Objects (Icon für Mapping-Screen).
-	Number of object references to follow	Integer	1	0 – 2	15	Anzahl der referenzierten Objekte (abhängig vom Window Type).
-	Number of objects to follow	Integer	1	0 – 255	16	Anzahl der direkt enthaltenen Objekte (nur für Typ 0 Free Form).
-	Number of macros to follow	Integer	1	0 – 255	17	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65535	18 + ...	Referenzierte Objekte (für vordefinierte Typen).
-	Repeat: {Object ID}	Integer	2	0 – 65534	var.	Enthaltene Objekte (für Typ 0 Free Form).
-	{X Location}	Signed Integer	2	-32768 bis +32767	var.	X-Position relativ zum Fenster.
-	{Y Location}	Signed Integer	2	-32768 bis +32767	var.	Y-Position relativ zum Fenster.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.35.0.2 Fenstertypen (Auszug aus B.19.2)

- **0:** Free Form (Freie Gestaltung, Working Set definiert Inhalt).
- **1:** 1x1 Numeric Output mit Einheiten.

- **2:** 1x1 Numeric Output ohne Einheiten.
- **3:** 1x1 String Output.
- **4:** 1x1 Numeric Input mit Einheiten.
- ...
- **7:** 1x1 Horizontal Linear Bargraph.
- **8:** 1x1 Single Button.
- ...
- **10:** 2x1 Numeric Output mit Einheiten.

20.35.1 Ereignisse (Events - Tabelle B.60)

Das Window Mask Objekt reagiert auf folgende Ereignisse:

- **On Show:** Wenn das Fenster als Teil einer User-Layout Maske sichtbar wird.
- **On Hide:** Wenn das Fenster ausgeblendet wird.
- **On Refresh:** Bei Änderungen an Kind-Objekten.
- **On Change Background Colour:** Reaktion auf Farbänderung.
- **On Change Child Location / Position:** Aktualisierung der Kind-Objekte.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.
- **On Pointing Event:** Touch-Ereignisse (nur bei Free Form Window).

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - Window Mask](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.19 verwiesen.

20.36 ID 35 – Key Group – ISO 11783-6 – B.20

Das **Key Group** Objekt mit der **ID 35** dient dazu, eine Gruppe von Softkeys zusammenzufassen. Dies wird primär in Verbindung mit **User-Layout Soft Key Masks** verwendet (ab VT Version 4).

20.36.0.1 Attribute und Record Format (Tabelle B.63)

Die folgende Tabelle beschreibt den Aufbau des Key Group Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	35	3	Objekttyp = Key Group.
[1]	Options	Bitmask	1	0 – 3	4	Bit 0: Available (0=Nicht verfügbar/geblenkt, 1=Verfügbar)Bit 1: Transparent (1=Hintergrundfarbe der Keys wird ignoriert).
[2]	Name	Integer	2	0 – 65534	5 – 6	Objekt-ID eines Output String oder Object Pointers (Name für Mapping-Screen).
-	Key Group Icon	Integer	2	0 – 65534, 65535	7 – 8	Objekt-ID eines Output

						Objects (Icon für Mapping-Screen).
-	Number of objects to follow	Integer	1	1 – 4	9	Anzahl der Key-Objekte in dieser Gruppe. Max. 4.
-	Number of macros to follow	Integer	1	0 – 255	10	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	11 + ...	Objekt-ID eines Key-Objekts oder Object Pointers auf Key.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.36.1 Ereignisse (Events - Tabelle B.62)

Das Key Group Objekt reagiert auf folgende Ereignisse:

- **On Change Attribute:** Reaktion auf generelle Attributänderungen.

20.36.2 Bedeutung und Funktionsweise

Das Key Group Objekt dient dazu, eine logisch zusammengehörige Gruppe von Softkeys zu definieren (z.B. „Hydraulikfunktionen“). Diese Gruppe wird primär in **User-Layout Soft Key Masks** verwendet.

- **User-Mapping:** Der Benutzer kann am Terminal entscheiden, an welcher Position er diese Gruppe von Tasten in seiner Softkey-Leiste haben möchte. Das VT zwingt den Benutzer, die Gruppe *als Ganzes* zu platzieren, um die logische Zusammengehörigkeit nicht zu zerstören.
- **Transparenz:** Es wird empfohlen, Key Groups transparent zu gestalten, damit das VT die Hintergrundfarbe der Tasten einheitlich setzen kann.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.20 verwiesen.

20.37 ID 36 – Graphics Context Object – ISO 11783-6 – B.18

Das **Graphics Context Object (GCO)** mit der **ID 36** (ab VT Version 4) stellt einen dynamischen Grafikpuffer (Canvas) zur Verfügung, in den die ECU zur Laufzeit zeichnen kann. Es ist vergleichbar mit einer Bitmap, deren Inhalt pixelgenau verändert werden kann.

20.37.0.1 Attribute und Record Format (Tabelle B.59)

Die folgende Tabelle beschreibt den Aufbau des Graphics Context Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-----	------	-----	---------------	----------------	-------------	--------------

-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	36	3	Objekttyp = Graphics Context.
[1]	Viewport Width	Integer	2	0 – 32767	4 – 5	Breite des sichtbaren Fensters (Anzeige im VT).
[2]	Viewport Height	Integer	2	0 – 32767	6 – 7	Höhe des sichtbaren Fensters.
[3]	Viewport X	Signed Integer	2	-32768 bis +32767	8 – 9	X-Position des Viewports relativ zum Canvas (0 = Links).
[4]	Viewport Y	Signed Integer	2	-32768 bis +32767	10 – 11	Y-Position des Viewports relativ zum Canvas (0 = Oben).
[5]	Canvas Width	Integer	2	0 – 32767	12 – 13	Gesamtbreite des Grafikspeichers.
[6]	Canvas Height	Integer	2	0 – 32767	14 – 15	Gesamthöhe des Grafikspeichers.
[7]	Viewport Zoom	Float	4	-32.0 bis +32.0	16 – 19	Zoomfaktor des Viewports (siehe Tabelle F.1).
[8]	Graphics Cursor X	Signed Integer	2	-32768 bis +32767	20 – 21	Aktuelle X-Position des Zeichencursors.
[9]	Graphics Cursor Y	Signed Integer	2	-32768 bis +32767	22 – 23	Aktuelle Y-Position des Zeichencursors.
[10]	Foreground Colour	Integer	1	0 – 255	24	Vordergrundfarbe für Zeichenoperationen.
[11]	Background Colour	Integer	1	0 – 255	25	Hintergrundfarbe. Wird beim Parsen zum Füllen genutzt.

[12]	Font Attributes	Integer	2	0 – 65534, 65535	26 – 27	Referenz auf Font Attributes Objekt (für Textbefehle).
[13]	Line Attributes	Integer	2	0 – 65534, 65535	28 – 29	Referenz auf Line Attributes Objekt (für Linienbefehle).
[14]	Fill Attributes	Integer	2	0 – 65534, 65535	30 – 31	Referenz auf Fill Attributes Objekt (für Füllbefehle).
[15]	Format	Integer	1	0 – 2	32	Farbtiefe des Canvas: 0=1-Bit, 1=4-Bit, 2=8-Bit.
[16]	Options	Bitmask	1	0 – 3	33	Bit 0: TransparenzBit 1: Farbe (0=FG/BG Attribute, 1=Line/Font/Fill Attr).
[17]	Transparency Colour	Integer	1	0 – 255	34	Transparenzfarbe (wenn Options Bit 0 gesetzt).

20.37.1 Funktionsweise und Struktur

Das GCO besteht aus zwei Hauptkomponenten:

1. **Canvas:** Ein persistenter Grafikspeicher (Bitmap) mit definierter Größe (Canvas Width/Height). Der Inhalt bleibt erhalten, auch wenn das Objekt nicht angezeigt wird.
2. **Viewport:** Ein „Fenster“, das einen Ausschnitt des Canvas anzeigt. Der Viewport definiert die Größe des Objekts auf der Maske. Durch Ändern von Viewport X/Y kann der Inhalt gescrollt (gepannt) werden.

20.37.2 Graphics Context Commands

Die Manipulation des Canvas erfolgt über spezielle Befehle (siehe ISO 11783-6, Anhang F), wie z. B.:

- Set Graphics Cursor: Setzt die Schreibposition.
- Draw Point / Line / Rectangle / Polygon / Ellipse: Zeichnet geometrische Formen.
- Draw Text: Schreibt Text an die Cursorposition.
- Copy Canvas: Kopiert Bereiche innerhalb des Canvas.

20.37.3 Ereignisse (Events - Tabelle B.58)

Das Graphics Context Objekt reagiert auf folgende Ereignisse:

- **On Change Attribute:** Wird ausgelöst bei Änderungen von Attributen (z. B. Viewport-Position, Zoom). Das VT aktualisiert die Anzeige.
- **On Change Background Colour:** Füllt das Objekt (den Canvas) mit der neuen Hintergrundfarbe. **Achtung:** Löscht den bisherigen Inhalt!

20.37.4 Anwendungsbeispiel

Typische Anwendungen sind **GPS-Karten** (Swath-Logging), bei denen die ECU kontinuierlich die gefahrene Spur als Linie in den GCO zeichnet.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.18 verwiesen.

20.38 ID 37 – Output List – ISO 11783-6 – B.9.4

Das **Output List** Objekt mit der **ID 37** (ab VT Version 4) wird verwendet, um eines von mehreren Objekten aus einer Liste anzuzeigen. Welches Objekt aktuell sichtbar ist, wird über einen Index (Value) gesteuert.

20.38.0.1 Attribute und Record Format (Tabelle B.25)

Die folgende Tabelle beschreibt den Aufbau des Output List Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	37	3	Objekttyp = Output List.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des Anzeigebereichs in Pixeln. Clipping erfolgt außerhalb.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des Anzeigebereichs in Pixeln. Clipping erfolgt außerhalb.
[3]	Variable reference	Integer	2	0 – 65534, 65535	8 – 9	Verweis auf ein Number Variable Objekt für den Index.
[4]	Value	Integer	1	0 – 255	10	Aktueller Index (0-254). 255 = keine Anzeige. Nur wenn Variable Ref == NULL.
-	Number of list items	Integer	1	0 – 255	11	Anzahl der Objekte in der Liste.
-	Number of macros to follow	Integer	1	0 – 255	12	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534, 65535	13 + ...	Objekt-ID eines Listenelements (was angezeigt werden soll). NULL = Leerer Platzhalter.

-	Repeat: {Event ID}	Integer	1	0 – 255	var.	(Nach Objekten) Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.38.1 Funktionsweise

Die Output List verhält sich ähnlich wie eine Animation, wird aber manuell über den Index gesteuert.

- **Index:** Der angezeigte Inhalt wird durch den Wert (Value oder Variable) bestimmt. Index 0 zeigt das erste Objekt in der Liste.
- **Spezialwert 255:** Bei Wert 255 wird nichts angezeigt (das Objekt ist unsichtbar).
- **NULL-Pointer:** Wenn ein Listeneintrag die ID NULL (65535) hat, wird für diesen Index ebenfalls nichts angezeigt.

20.38.2 Ereignisse (Events - Tabelle B.24)

Das Output List Objekt reagiert auf folgende Ereignisse:

- **On Change Value:** Wird ausgelöst, wenn sich der Index ändert. Das VT aktualisiert die Anzeige.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.
- **On Change Size:** Reaktion auf Größenänderung.
- **On Refresh:** Wird ausgelöst, wenn das VT das Objekt neu zeichnen muss.

Weitere Informationen und Beispiele finden sich im [ISOBUS Wiki - List \(Output\)](#) von Tobias Tenberg.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.9.4 verwiesen.

20.39 ID 38 – Extended Input Attributes – ISO 11783-6 – B.14.6

Das **Extended Input Attributes** Objekt mit der **ID 38** (ab VT Version 4) dient der Validierung von Texteingaben (Input String) bei Verwendung von **WideStrings** (Unicode).

20.39.0.1 Attribute und Record Format (Tabelle B.53)

Die folgende Tabelle beschreibt den Aufbau des Extended Input Attributes Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	38	3	Objekttyp = Extended Input Attributes.
[1]	Validation type	Integer	1	0 – 1	4	0=Erlaubte Zeichen (Whitelist), 1=Verbotene Zeichen (Blacklist).
-	Number of code planes to follow	Integer	1	1 – 17	5	Anzahl der definierten Unicode-Ebenen.
-	Repeat: {Code plane number}	Integer	1	0 – 16	6 ...	Nummer der Unicode-Ebene (0 = BMP).

-	{Number of character ranges to follow}	Integer	1	1 – 255	7 ...	Anzahl der Bereiche in dieser Ebene.
-	Repeat: {{First character}}	Integer	2	0 – 65535	8 ...	Startzeichen des Bereichs (WideChar).
-	{{Last character}}	Integer	2	0 – 65535	10 ...	Endzeichen des Bereichs (WideChar).

20.39.1 Funktionsweise

Während das einfache *Input Attributes* Objekt (ID 26) nur 8-Bit Zeichen unterstützt, ermöglicht dieses Objekt die feingranulare Steuerung erlaubter Unicode-Zeichen (WideString). Dies wird für Sprachen benötigt, die mehr als 256 Zeichen umfassen (z. B. Asiatisch, Kyrillisch). Die Validierung erfolgt durch Definition von Bereichen innerhalb von Unicode-Ebenen.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.14.6 verwiesen.

20.40 ID 39 – Colour Map – ISO 11783-6 – B.17

Das **Colour Map** Objekt mit der **ID 39** (optional ab VT Version 4/5, Pflicht ab VT Version 6) ermöglicht es einer Working Set, die Farbtabelle des Terminals zur Laufzeit umzudefinieren.

20.40.0.1 Attribute und Record Format (Tabelle B.57)

Die folgende Tabelle beschreibt den Aufbau des Colour Map Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	39	3	Objekttyp = Colour Map.
-	Number of colour indexes to follow	Integer	2	2, 16, 256	4 – 5	Anzahl der folgenden Einträge. Muss zur Farbtiefe des VTs passen (siehe Get Hardware).
-	Repeat: {Colour Map}	Integer	var.	0 – 255	6 ...	Liste von Farb-Indizes.

20.40.0.2 Struktur der Einträge

Die Einträge definieren eine Umleitung der Farb-Indizes.

- **Beispiel:** Wenn der Eintrag an Index 0 den Wert 1 hat, wird überall dort, wo im Design Farbe 0 (Schwarz) verwendet wurde, nun Farbe 1 (Weiß) angezeigt.
- Die Länge der Einträge hängt vom VT-Typ ab:
 - **Typ 0 (Monochrom):** 2 Einträge.
 - **Typ 1 (16 Farben):** 16 Einträge.
 - **Typ 2 (256 Farben):** 256 Einträge.

20.40.1 Funktionsweise

Normalerweise verwendet ein Terminal eine Standard-Farbpalette (z. B. 256 Farben gemäß ISO). Mit der Colour Map kann eine Working Set definieren, dass z. B. der Farbindex 1 (Standard: Rot) stattdessen als Blau dargestellt werden soll.

- **Indirektion:** Dies bietet eine Ebene der Indirektion. Anstatt jedes Objekt einzeln zu ändern, kann durch einfaches Umschalten der Colour Map das gesamte Farbschema der Anwendung gewechselt werden (z. B. Tag-/Nacht-Modus).
- **Aktivierung:** Eine Colour Map wird über das Kommando Select Colour Map aktiviert.

20.40.2 Anmerkung zu VT Version 6

Ab Version 6 wird zusätzlich das *Colour Palette* Objekt (ID 45) unterstützt, mit dem die RGB-Werte der Farben selbst geändert werden können. Die Colour Map mappt Indizes auf Indizes, während die Colour Palette Indizes auf RGB-Werte mappt.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.17 verwiesen.

20.41 ID 40 – Object Label Reference List – ISO 11783-6 – B.21

Das **Object Label Reference List** Objekt mit der **ID 40** (ab VT Version 5) wird verwendet, um Objekten (wie Variablen oder Eingabefeldern) eine Liste von Beschriftungsobjekten (Labels) zuzuordnen.

20.41.0.1 Attribute und Record Format (Tabelle B.64)

Die folgende Tabelle beschreibt den Aufbau des Object Label Reference List Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	40	3	Objekttyp = Object Label Reference List.
[1]	Number of Labelled objects	Integer	2	0 – 65535	4 – 5	Anzahl der folgenden Label-Zuweisungen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	6 – 7 ...	Objekt-ID des zu beschrifteten Objekts (z. B. Input Number).
-	{String Variable reference}	Integer	2	0 – 65535	8 – 9 ...	Objekt-ID einer String Variable mit dem Labeltext (oder FFFFh = kein Text).
-	{Font type}	Integer	1	0 – 255	10 ...	Schriftart (siehe Annex K). Ignoriert bei WideString oder NULL.
-	{Object Label graphic representation}	Integer	2	0 – 65535	11 – 12 ...	Objekt-ID einer Grafik (Icon) für das Label

						(oder FFFFh = keine Grafik).
--	--	--	--	--	--	------------------------------

20.41.1 Bedeutung und Funktionsweise

Das Objekt dient dazu, anderen Objekten (z. B. Working Set, Input-Felder) einen Namen (Text) und ein Icon (Grafik) zuzuweisen. Diese „Labels“ werden vom VT verwendet:

- **Working Set Label:** Das Label für das Working Set Objekt wird in der Liste der aktiven Arbeitsgruppen angezeigt.
- **Input Labels:** Bei Eingabefeldern zeigt das VT das Label (Text/Grafik) im Popup-Editor an, damit der Benutzer weiß, welchen Wert er gerade bearbeitet (z. B. „Sämenge“ statt nur „120“).

Es ist nur **eine** Object Label Reference List pro Objektpool erlaubt.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.21 verwiesen.

20.42 ID 41 – External Object Definition – ISO 11783-6 – B.22

Das **External Object Definition** Objekt mit der **ID 41** (ab VT Version 5) ist Teil des Mechanismus für **Working-Set-übergreifende Objektreferenzen**. Es definiert, welche Objekte eines eigenen Pools von anderen Working Sets referenziert werden dürfen.

20.42.0.1 Attribute und Record Format (Tabelle B.66)

Die folgende Tabelle beschreibt den Aufbau des External Object Definition Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	41	3	Objekttyp = External Object Definition.
[1]	Options	Bitmask	1	0 – 1	4	Bit 0: Enabled (1=Freigabe aktiv, 0=Deaktiviert).
[2]	NAME 0	Integer	4	0 – 2 ³² -1	5 – 8	Byte 1–4 des NAME (ISO 11783-5) des berechtigten Working Sets.
[3]	NAME 1	Integer	4	0 – 2 ³² -1	9 – 12	Byte 5–8 des NAME des berechtigten Working Sets.
-	Number of objects to follow	Integer	1	0 – 255	13	Anzahl der freigegebenen Objekte.
-	Repeat: {Object ID}	Integer	2	0 – 65534, 65535	14 + ...	Objekt-ID eines Objekts aus dem eigenen Pool, das für das externe WS freigegeben wird.

20.42.1 Funktionsweise

Damit Working Set A ein Objekt von Working Set B anzeigen kann (via *External Object Pointer*), muss Working Set B dieses Objekt explizit in einer *External Object Definition* für Working Set A freigeben. Dies dient der Sicherheit und Kontrolle über die eigenen Pool-Ressourcen.

- **Empfehlung:** Das Enabled-Bit sollte beim Laden des Pools zunächst auf 0 gesetzt sein und erst zur Laufzeit aktiviert werden, wenn der NAME des Partners bekannt und aktuell ist.

20.42.2 Ereignisse (Events - Tabelle B.65)

Das External Object Definition Objekt reagiert auf folgende Ereignisse:

- **On Change Attribute:** Wird ausgelöst durch das Kommando `Change Attribute` (z. B. Aktivieren/Deaktivieren). Das VT evaluiert alle aktuell angezeigten External Object Pointer neu, die auf Objekte dieses WS verweisen.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.22 verwiesen.

20.43 ID 42 – External Reference NAME – ISO 11783-6 – B.23

Das **External Reference NAME** Objekt mit der ID 42 (ab VT Version 5) identifiziert ein anderes Working Set auf dem Bus über seinen eindeutigen ISO-NAME.

20.43.0.1 Attribute und Record Format (Tabelle B.68)

Die folgende Tabelle beschreibt den Aufbau des External Reference NAME Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	42	3	Objekttyp = External Reference NAME.
[1]	Options	Bitmask	1	0 – 1	4	Bit 0: Enabled (1=Referenz aktiv, 0=Deaktiviert).
[2]	NAME 0	Integer	4	0 – 2 ³² -1	5 – 8	Byte 1–4 des NAME (ISO 11783-5) des referenzierten Working Sets.
[3]	NAME 1	Integer	4	0 – 2 ³² -1	9 – 12	Byte 5–8 des NAME des referenzierten Working Sets.

20.43.1 Funktionsweise

Dieses Objekt fungiert als „Adressbucheintrag“. Wenn eine ECU ein Objekt von einer anderen ECU einbinden möchte, nutzt sie diesen Eintrag, um dem Terminal mitzuteilen, von welcher ECU (identifiziert durch den NAME) das Objekt stammt.

- **Aktivierung:** Sollte initial deaktiviert sein. Wenn das referenzierte Working Set online ist, kann die ECU dieses Objekt aktivieren.

20.43.2 Ereignisse (Events - Tabelle B.67)

Das External Reference NAME Objekt reagiert auf folgende Ereignisse:

- **On Change Attribute:** Wird ausgelöst durch das Kommando Change Attribute. Das VT evaluiert alle aktuell angezeigten External Object Pointer neu, die auf dieses NAME-Objekt verweisen.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.23 verwiesen.

20.44 ID 43 – External Object Pointer – ISO 11783-6 – B.24

Das **External Object Pointer** Objekt mit der ID 43 (ab VT Version 5) ermöglicht es einer Working Set, Objekte anzuzeigen, die sich physisch im Objekt-Pool einer **anderen** Working Set befinden.

20.44.0.1 Attribute und Record Format (Tabelle B.70)

Die folgende Tabelle beschreibt den Aufbau des External Object Pointer Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	43	3	Objekttyp = External Object Pointer.
[1]	Default Object ID	Integer	2	0 – 65534, 65535	4 – 5	ID eines lokalen Ersatzobjekts, falls externer Zugriff fehlschlägt.
[2]	External Reference NAME ID	Integer	2	0 – 65534, 65535	6 – 7	ID eines External Reference NAME Objekts, das das Ziel-WS identifiziert.
[3]	External Object ID	Integer	2	0 – 65534, 65535	8 – 9	ID des Objekts im fremden Pool.

20.44.1 Funktionsweise und Regeln

Das Objekt ermöglicht die Anzeige von Objekten aus fremden Objekt-pools.

- **Anzeige:** Das Terminal sucht das Objekt im Pool der durch den NAME identifizierten ECU und zeichnet es an die Stelle des Pointers.
- **Kontext:** Ereignisse (z. B. Button Press) und Makros werden im Kontext des **Original-Working-Sets** (Besitzer des Objekts) ausgeführt. Nachrichten (z. B. Button Activation) gehen an den Besitzer.
- **Fallback:** Wenn das externe Objekt nicht gefunden wird, nicht freigegeben ist oder ungültig ist, wird das Default Object angezeigt.
- **Sicherheit:** Das Ziel-Objekt muss durch die besitzende ECU via *External Object Definition* (ID 41) freigegeben worden sein.

20.44.2 Ereignisse (Events - Tabelle B.69)

Das External Object Pointer Objekt reagiert auf folgende Ereignisse:

- **On Change Attribute:** Wird ausgelöst durch das Kommando Change Attribute. Das VT evaluiert den Pointer neu und zeichnet ggf. neu.

20.44.3 Anwendungsbeispiel

Ein Traktor (Working Set A) möchte in seiner Hauptmaske den Füllstand eines gezogenen Düngestreuers (Working Set B) anzeigen. Er nutzt dazu einen *External Object Pointer*, der auf die Füllstandsanzeige im Pool des Streuers verweist.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.24 verwiesen.

20.45 ID 44 – Animation – ISO 11783-6 – B.25

Das **Animation** Objekt mit der **ID 44** (ab VT Version 5) ermöglicht die automatische oder manuelle Wiedergabe einer Sequenz von Objekten (Frames).

20.45.0.1 Attribute und Record Format (Tabelle B.72)

Die folgende Tabelle beschreibt den Aufbau des Animation Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objekt-pool.
[0]	Type	Integer	1	44	3	Objekttyp = Animation.
[1]	Width	Integer	2	0 – 65535	4 – 5	Breite des Animations-bereichs in Pixeln.
[2]	Height	Integer	2	0 – 65535	6 – 7	Höhe des Animations-bereichs in Pixeln.
[3]	Refresh Interval	Integer	2	0 – 65535	8 – 9	Zeit in ms zwischen den Frames (0 = Timer aus).
[4]	Value	Integer	1	0 – 255	10	Aktueller Listenindex des angezeigten Objekts (0-254).
[5]	Enabled	Integer	1	0 oder 1	11	0=Stopped (Deaktiviert), 1=Animating (Aktiviert).
[6]	First Child Index	Integer	1	0 – 254	12	Startindex der Animationssequenz.
[7]	Last Child Index	Integer	1	0 – 254	13	Endindex der Animationssequenz.
[8]	Default Child Index	Integer	1	0 – 254	14	Index des Objekts, das angezeigt wird,

						wenn die Animation deaktiviert ist (abhängig von Options).
[9]	Options	Bitmask	1	0 – 7	15	Bit 0: Sequence (0=Single Shot, 1=Loop)Bits 1–2: Disabled Behaviour (0=Pause, 1=Reset to First, 2=Default Object, 3=Blank).
-	Number of objects to follow	Integer	1	0 – 255	16	Anzahl der enthaltenen Frames (Objekte).
-	Number of macros to follow	Integer	1	0 – 255	17	Anzahl der folgenden Makro-Referenzen.
-	Repeat: {Object ID}	Integer	2	0 – 65534	18 + ...	Objekt-ID eines Frames (Child-Objekt).
-	{X Location}	Signed Integer	2	-32768 bis +32767	20 + ...	X-Position relativ zur Animation.
-	{Y Location}	Signed Integer	2	-32768 bis +32767	22 + ...	Y-Position relativ zur Animation.
-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.45.1 Funktionsweise

Das Animationsobjekt verwaltet eine Liste von Kind-Objekten. Wenn Enabled auf 1 steht, inkrementiert das Terminal den Value (Index) automatisch im Rhythmus des Refresh Interval.

- **Loop:** Nach Erreichen des Last Child Index wird wieder beim First Child Index begonnen.
- **Single Shot:** Die Animation stoppt beim letzten Frame.
- **Deaktivierung:** Das Verhalten beim Stoppen (Pause, Reset auf 1. Frame, Default-Bild oder Leer) wird über die Optionen gesteuert.

20.45.2 Ereignisse (Events - Tabelle B.71)

Das Animation Objekt reagiert auf folgende Ereignisse:

- **On Enable / On Disable:** Zustandsänderung.
- **On Change Value:** Wenn der Index (manuell oder automatisch) geändert wird. Das VT zeichnet das neue Objekt.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.
- **On Change Size:** Reaktion auf Größenänderung.
- **On Refresh:** Wird ausgelöst durch den Timer (Refresh Interval) oder andere Refresh-Bedingungen.

20.45.3 Empfehlung

Um die Performance des Terminals nicht zu überlasten, sollten die Einzelobjekte klein gehalten werden. Ein Refresh-Intervall von mindestens 200 ms wird empfohlen.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.25 verwiesen.

20.46 ID 45 – Colour Palette – ISO 11783-6 – B.26

Das **Colour Palette** Objekt mit der **ID 45** (ab VT Version 6) erlaubt es einer Working Set, die Standard-Farbpalette des Terminals vollständig durch eigene ARGB-Farbdefinitionen zu ersetzen.

20.46.0.1 Attribute und Record Format (Tabelle B.73)

Die folgende Tabelle beschreibt den Aufbau des Colour Palette Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	45	3	Objekttyp = Colour Palette.
[1]	Options	Bitmask	1	0	4	Reserviert (sollte 0 sein).
-	Number of ARGB values to follow	Integer	2	0 – 256	5 – 6	Anzahl der folgenden Farbdefinitionen.
-	Repeat: {B}	Integer	1	0 – 255	7 + ...	Blau-Wert.
-	{G}	Integer	1	0 – 255	8 + ...	Grün-Wert.
-	{R}	Integer	1	0 – 255	9 + ...	Rot-Wert.
-	{A}	Integer	1	0 – 255	10 + ...	Alpha-Wert (0=Transparent, 255=Opak).

20.46.1 Funktionsweise

Wenn ein Colour Palette Objekt aktiv ist, ersetzt es die Standardfarben des VTs für dieses Working Set. Die Farben werden beginnend bei Index 0 aufgefüllt.

- **Aktivierung:** Eine Colour Palette kann über das *Working Set Special Controls* Objekt (ID 47) als Standard gesetzt oder zur Laufzeit per *Select Colour Map or Palette* Kommando aktiviert werden.
- **Transparenz:** Der Alpha-Kanal ermöglicht halbtransparente Farben. Dies sollte jedoch mit Vorsicht genutzt werden, insbesondere bei überlagerten Objekten.

20.46.2 Unterschied zur Colour Map (ID 39)

Während die *Colour Map* nur die Zuweisung (Indizierung) vorhandener Terminal-Farben ändert, definiert die *Colour Palette* die physikalischen Farben (ARGB-Werte) neu. Dies ermöglicht echtes Branding in Firmenfarben.

20.46.3 Ereignisse (Events)

Das Objekt selbst löst keine Events aus, beeinflusst aber die Darstellung aller Objekte, die Farben verwenden.

- **On Change Attribute:** Wird ausgelöst, wenn Attribute geändert werden. Das VT muss ggf. den gesamten Bildschirm neu zeichnen.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.26 verwiesen.

20.47 ID 46 – Graphic Data (PNG) – ISO 11783-6 – B.27

Das **Graphic Data** Objekt mit der **ID 46** (ab VT Version 6) dient zur Speicherung von Rohdaten für Grafiken, insbesondere im **PNG-Format**.

20.47.0.1 Attribute und Record Format (Tabelle B.74)

Die folgende Tabelle beschreibt den Aufbau des Graphic Data Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	46	3	Objekttyp = Graphic Data.
[1]	Format	Integer	1	0	4	Grafikformat: 0 = PNG (max. 32-bit RGBA).
-	Number of bytes in raw data	Integer	4	0 – 2 ³² -1	5 – 8	Anzahl der Bytes in den Rohdaten.
-	Repeat: {raw data}	Integer	1	0 – 255	9 ...	Rohdaten der Grafik (Bytes).

20.47.1 Besonderheiten

Das **Graphic Data** Objekt (ab VT Version 6) dient zur Speicherung von Rohdaten für Grafiken, insbesondere im **PNG-Format**. Im Gegensatz zum klassischen *Picture Graphic* Objekt (ID 20), das auf einfachen Bitmaps basierte, nutzt dieses Objekt den Industriestandard PNG.

- **Eigenständigkeit:** Das Objekt enthält seine eigene Farbpalette (innerhalb der PNG-Daten) und wird daher **nicht** von der *Colour Map* (ID 39) oder *Colour Palette* (ID 45) der Working Set beeinflusst.

20.47.2 Verwendung

Dieses Objekt wird normalerweise nicht direkt in einer Maske platziert, sondern von einem **Scaled Graphic** Objekt (ID 48) referenziert, um es anzuzeigen und auf die gewünschte Größe zu skalieren. Es kann auch in einem *External Object Pointer* (ID 43) referenziert werden.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.27 verwiesen.

20.48 ID 47 – Working Set Special Controls – ISO 11783-6 – B.29

Das **Working Set Special Controls** Objekt mit der **ID 47** (ab VT Version 6) dient zur zentralen Steuerung von pool-weiten Einstellungen wie Farben und Sprachen.

20.48.0.1 Attribute und Record Format (Tabelle B.78)

Die folgende Tabelle beschreibt den Aufbau des Working Set Special Controls Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	47	3	Objekttyp = Working Set Special Controls.
[1]	Number of Bytes to follow	Integer	2	5 – 65535	4 – 5	Anzahl der Bytes in diesem Objekt nach diesem Attribut. Ermöglicht Parsing-Kompatibilität.
[2]	Object ID of Colour Map object	Integer	2	0 – 65534, 65535	6 – 7	ID der initial zu verwendenden Colour Map (ID 39) oder NULL.
[3]	Object ID of Colour Palette object	Integer	2	0 – 65534, 65535	8 – 9	ID der initial zu verwendenden Colour Palette (ID 45) oder NULL.
-	Number of languages pairs to follow	Integer	1	0 – 255	10	Anzahl der folgenden Sprach/Länder-Code-Paare.
-	Repeat: {Language Code}	String	2	-	11 – 12 ...	2-Buchstaben Code gemäß ISO 639-1 (z. B. „de“).
-	{Country Code}	String	2	-	13 – 14 ...	2-Buchstaben Code gemäß ISO 3166-1 (z. B. „DE“) oder „20 20“ (Hex) falls nicht anwendbar.

20.48.1 Bedeutung und Funktionsweise

Dieses Objekt ist der zentrale Anlaufpunkt für das Terminal beim Laden des Pools (ab VT Version 6).

- **Farben:** Es definiert, welche *Colour Map* und *Colour Palette initial* beim Aktivieren des Pools verwendet werden sollen.

- **Sprachen:** Es definiert eine Liste von unterstützten Sprachen, die die Liste im *Working Set* Objekt (ID 0) **ersetzt**. Durch die Kombination von Sprach- und Ländercode (z. B. pt + BR vs. pt + PT) ist eine präzisere Auswahl möglich.
- **Erweiterbarkeit:** Das Attribut *Number of Bytes to follow* erlaubt es, das Objekt in Zukunft um neue Attribute zu erweitern, ohne ältere VTs zu verwirren (diese überspringen einfach die unbekanntes Bytes).

Es darf maximal **ein** Working Set Special Controls Objekt pro Objektpool geben.

20.48.2 Ereignisse (Events - Tabelle B.77)

Das Objekt reagiert auf folgende Ereignisse:

- **On Refresh:** Wird ausgelöst, wenn sich Einstellungen ändern, die einen kompletten Neuaufbau der Darstellung erfordern (z. B. Farbpalette).

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.29 verwiesen.

20.49 ID 48 – Scaled Graphic – ISO 11783-6 – B.28

Das **Scaled Graphic** Objekt mit der **ID 48** (ab VT Version 6) dient zur Anzeige und Skalierung von Grafikobjekten.

20.49.0.1 Attribute und Record Format (Tabelle B.76)

Die folgende Tabelle beschreibt den Aufbau des Scaled Graphic Objekts im Objektpool.

AID	Name	Typ	Größe (Bytes)	Bereich / Wert	Record Byte	Beschreibung
-	Object ID	Integer	2	0 – 65534	1 – 2	Eindeutige ID im Objektpool.
[0]	Type	Integer	1	48	3	Objekttyp = Scaled Graphic.
[1]	Width	Integer	2	0 – 65535	4 – 5	Zielbreite in Pixeln.
[2]	Height	Integer	2	0 – 65535	6 – 7	Zielhöhe in Pixeln.
[3]	ScaleType	Integer	1	0 – 127	8	Skalierungsmodus und Justierung (siehe unten).
[4]	Options	Bitmask	1	0 – 1	9	Bit 0: Flashing (0=Normal, 1=Blindend).
[5]	Value	Integer	2	0 – 65535	10 – 11	Objekt-ID des anzuzeigenden Grafikobjekts (Graphic Data ID 46 oder Picture Graphic ID 20) oder Pointer.
-	Number of macros to follow	Integer	1	0 – 255	12	Anzahl der folgenden Makro-Referenzen.

-	Repeat: {Event ID}	Integer	1	0 – 255	var.	Event ID, die das Makro auslöst.
-	{Macro ID}	Integer	1	0 – 255	var.	Makro ID des auszuführenden Makros.

20.49.0.2 Skalierungstypen (Bits 0-2 von ScaleType)

- **0:** Keine Skalierung (Originalgröße aus den Rohdaten verwenden).
- **1:** Auf Breite skalieren (Seitenverhältnis beibehalten).
- **2:** Auf Höhe skalieren (Seitenverhältnis beibehalten).
- **3:** Auf Breite und Höhe skalieren (Verzerrung möglich).
- **4:** In Bereich einpassen (Best Fit, Seitenverhältnis beibehalten, Grafik wird so groß wie möglich).

20.49.0.3 Justierung (Bits 3-6 von ScaleType)

Definiert die Position innerhalb des durch Width und Height definierten Bereichs:

- **Horizontal (Bits 3-4):** 0=Links, 1=Mitte, 2=Rechts.
- **Vertikal (Bits 5-6):** 0=Oben, 1=Mitte, 2=Unten.

20.49.1 Ereignisse (Events - Tabelle B.75)

Das Scaled Graphic Objekt reagiert auf folgende Ereignisse:

- **On Refresh:** Wird ausgelöst bei Masken-Refresh oder Optionsänderung.
- **On Change Attribute:** Reaktion auf generelle Attributänderungen.
- **On Change Value:** Wird ausgelöst, wenn das referenzierte Grafikobjekt (Value) geändert wird. Das VT lädt und skaliert das neue Bild.

Hinweis: Für detaillierte Spezifikationen wird auf die offizielle ISO 11783-6:2018, B.28 verwiesen.

20.50 ISOBUS VT Objekte

Diese Sektion beschreibt die verschiedenen Objekttypen des ISOBUS Virtuellen Terminals gemäß ISO 11783-6.

20.50.1 Liste der Objekte

- ID 0 - Working Set
- ID 1 - Data Mask
- ID 2 - Alarm Mask
- ID 3 - Container
- ID 4 - Soft Key Mask
- ID 5 - Key (Soft Key)
- ID 6 - Button
- ID 7 - Input Boolean
- ID 8 - Input String
- ID 9 - Input Number
- ID 10 - Input List
- ID 11 - Output String
- ID 12 - Output Number
- ID 13 - Output Line
- ID 14 - Output Rectangle
- ID 15 - Output Ellipse
- ID 16 - Output Polygon
- ID 17 - Output Meter
- ID 18 - Output Linear Bar Graph
- ID 19 - Output Arched Bar Graph

-
- ID 20 - Picture Graphic
 - ID 21 - Number Variable
 - ID 22 - String Variable
 - ID 23 - Font Attributes
 - ID 24 - Line Attributes
 - ID 25 - Fill Attributes
 - ID 26 - Input Attributes
 - ID 27 - Object Pointer
 - ID 28 - Macro
 - ID 29 - Auxiliary Function Type 1
 - ID 30 - Auxiliary Input Type 1
 - ID 31 - Auxiliary Function Type 2
 - ID 32 - Auxiliary Input Type 2
 - ID 33 - Auxiliary Control Designator Type 2 Object Pointer
 - ID 34 - Window Mask
 - ID 35 - Key Group
 - ID 36 - Graphics Context Object
 - ID 37 - Output List
 - ID 38 - Extended Input Attributes
 - ID 39 - Colour Map
 - ID 40 - Object Label Reference List
 - ID 41 - External Object Definition
 - ID 42 - External reference NAME
 - ID 43 - External Object Pointer
 - ID 44 - Animation
 - ID 45 - Colour Palette
 - ID 46 - Graphic Data (PNG)
 - ID 47 - Working Set Special Controls
 - ID 48 - Scaled Graphic