

BS ISO 11783-6:2014



BSI Standards Publication

Tractors and machinery for agriculture and forestry — Serial control and communications data network Part 6: Virtual terminal

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This British Standard is the UK implementation of ISO 11783-6:2014. It supersedes BS ISO 11783-6:2010 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee AGE/6, Agricultural tractors and forestry machinery.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2014. Published by BSI Standards Limited 2014

ISBN 978 0 580 78555 9

ICS 35.240.99; 65.060.01

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2014.

Amendments issued since publication

Date	Text affected
------	---------------

INTERNATIONAL STANDARD

BS ISO 11783-6:2014

ISO
11783-6

Third edition
2014-07-01

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 6: Virtual terminal

*Tracteurs et machines agricoles et forestiers — Réseaux de commande
et de communication de données en série —*

Partie 6: Terminal virtuel



Reference number
ISO 11783-6:2014(E)

AVA Documents
Document Sharing Hub
© ISO 2014



COPYRIGHT PROTECTED DOCUMENT

© ISO 2014

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	xiii
Introduction.....	xv
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Technical requirements	5
4.1 Overview.....	5
4.2 Operator input and control	7
4.3 Acoustic alarm	9
4.4 Coordinate system	9
4.5 Display areas	9
4.5.1 General	9
4.5.2 Data Mask	9
4.5.3 Soft Key Mask area and Soft Key designators	9
4.6 Behaviour	13
4.6.1 Object pools	13
4.6.2 Working Sets.....	13
4.6.3 Multiple Visually Similar Working Sets	15
4.6.4 Displayed Working Set number	16
4.6.5 Language, formats and measurement units selection	16
4.6.6 Initialization	17
4.6.7 System Shutdown	18
4.6.8 Working Set object and active masks	20
4.6.9 Connection management.....	22
4.6.10 Updating the operator interface	25
4.6.11 Special objects	25
4.6.12 Relative X/Y positions	30
4.6.13 Overlaid objects.....	31
4.6.14 Alarm handling	32
4.6.15 Clipping	33
4.6.16 Scaling	34
4.6.17 Operator input.....	34
4.6.18 Soft Key and Button activation	37
4.6.19 Font rendering	38
4.6.20 Object Rendering Accuracy, Quality and VT Developer Freedom	47
4.6.21 Filling output shape objects.....	48
4.6.22 Events.....	49
4.6.23 Touch screens and pointing devices	50
4.6.24 Proprietary Means	51
4.6.25 VT Number	51
4.6.26 Packet Padding.....	51
4.7 Displaying Data from Multiple Working Sets on One Mask	51
4.7.1 General	51
4.7.2 User-Layout Data Mask.....	52
4.7.3 Window Mask object	53
4.7.4 Window Mask content.....	53
4.7.5 Window Cell Size and Borders.....	55
4.7.6 Window Mask Scaling.....	55
4.7.7 Using Window Masks Outside of User-Layout Data Masks.....	56

4.7.8	User-Layout Soft Key Mask	56
4.7.9	Key Group Objects	57
4.7.10	Key Cell Size and Borders	58
4.7.11	Key Group Scaling	58
4.7.12	Using Key Group Objects outside of User-Layout Soft Key Masks	58
4.7.13	Operator Inputs	59
4.7.14	Refreshing On Screen Data	59
4.7.15	Look and Feel	60
4.7.16	Uploading New Window Mask and Key Group objects	61
Annex A	(normative) Object, event, colour and command codes	63
A.1	Object types	63
A.1.1	General	63
A.1.2	Nomenclature	65
A.1.3	Object relationships	66
A.2	Event types	68
A.3	VT standard colour palette	70
A.4	Command/parameter code summary	72
Annex B	(normative) Object definitions	78
B.1	Working Set object	78
B.2	Data Mask object	81
B.3	Alarm Mask object	83
B.4	Container object	86
B.5	Soft Key Mask object	88
B.6	Key object	89
B.7	Button object	91
B.8	Input field objects	95
B.8.1	General	95
B.8.2	Input Boolean object	97
B.8.3	Input String object	98
B.8.4	Input Number object	101
B.8.5	Input List object	104
B.9	Output field objects	108
B.9.1	General	108
B.9.2	Output String object	109
B.9.3	Output Number object	110
B.9.4	Output List object	113
B.10	Output shape objects	115
B.10.1	General	115
B.10.2	Output Line object	115
B.10.3	Output Rectangle object	118
B.10.4	Output Ellipse object	120
B.10.5	Output Polygon object	123
B.11	Output graphic objects	125
B.11.1	General	125
B.11.2	Output Meter object	125
B.11.3	Output Linear Bar Graph object	129
B.11.4	Output Arched Bar Graph object	133
B.12	Picture Graphic object	137
B.12.1	General	137
B.12.2	Picture Graphic object raw data format and compression	139
B.13	Variable objects	139
B.13.1	General	139
B.13.2	Number Variable object	140
B.13.3	String Variable object	140
B.14	Attribute objects	141
B.14.1	General	141
B.14.2	Font Attributes object	141
B.14.3	Line Attributes object	143

B.14.4	Fill Attributes object	145
B.14.5	Input Attributes object	147
B.14.6	Extended Input Attributes object	148
B.15	Object Pointer object	151
B.16	Macro object	151
B.17	Colour Map object	152
B.18	Graphics Context object	154
B.19	Window Mask object	158
B.19.1	General	158
B.19.2	Window Mask Window Types	163
B.20	Key Group object	182
B.21	Object Label Reference List object	184
B.22	External Object Definition object	185
B.23	External Reference NAME object	186
B.24	External Object Pointer object	187
B.25	Animation object	188
Annex C	(normative) Object transport protocol	192
C.1	Virtual terminal messages and object transfer	192
C.2	Building object pools	192
C.2.1	General	192
C.2.2	Object pool transfer procedure	193
C.2.3	Object pool transfer message	194
C.2.4	End of Object Pool message	194
C.2.5	End of Object Pool response	195
C.2.6	Updating pools at runtime	195
Annex D	(normative) Technical data messages	197
D.1	General	197
D.2	Get Memory message	197
D.3	Get Memory response	198
D.4	Get Number of Soft Keys message	199
D.5	Get Number of Soft Keys response	199
D.6	Get Text Font Data message	199
D.7	Get Text Font Data response	200
D.8	Get Hardware message	200
D.9	Get Hardware response	201
D.10	Get Supported Widechars message	201
D.11	Get Supported WideChars response	202
D.12	Get Window Mask Data message	203
D.13	Get Window Mask Data response	203
D.14	Get Supported Objects message	203
D.15	Get Supported Objects response	204
Annex E	(normative) Non-volatile memory operations commands	205
E.1	General	205
E.1.1	Introduction	205
E.1.2	Version Management – VT version 4 and prior	206
E.1.3	Version Management – VT version 5 and later	206
E.2	Get Versions message	206
E.3	Get Versions response	206
E.4	Store Version command	206
E.5	Store Version response	207
E.6	Load Version command	207
E.7	Load Version response	207
E.8	Delete Version command	208
E.9	Delete Version response	208
E.10	Extended Get Versions message	208
E.11	Extended Get Versions response	209
E.12	Extended Store Version command	209

E.13	Extended Store Version response	209
E.14	Extended Load Version command	210
E.15	Extended Load Version response	210
E.16	Extended Delete Version command	211
E.17	Extended Delete Version response	211
Annex F (normative) Command and Macro messages		212
F.1	General	212
F.2	Hide/Show Object command	212
F.3	Hide/Show Object response	212
F.4	Enable/Disable Object command	213
F.5	Enable/Disable Object response	213
F.6	Select Input Object command	213
F.7	Select Input Object response	214
F.8	ESC command	215
F.9	ESC response	215
F.10	Control Audio Signal command	215
F.11	Control Audio Signal response	217
F.12	Set Audio Volume command	217
F.13	Set Audio Volume response	218
F.14	Change Child Location command	218
F.15	Change Child Location response	219
F.16	Change Child Position command	219
F.17	Change Child Position response	219
F.18	Change Size command	220
F.19	Change Size response	220
F.20	Change Background Colour command	220
F.21	Change Background Colour response	221
F.22	Change Numeric Value command	221
F.23	Change Numeric Value response	222
F.24	Change String Value command	223
F.25	Change String Value response	224
F.26	Change End Point command	224
F.27	Change End Point response	224
F.28	Change Font Attributes command	225
F.29	Change Font Attributes response	225
F.30	Change Line Attributes command	225
F.31	Change Line Attributes response	226
F.32	Change Fill Attributes command	226
F.33	Change Fill Attributes response	227
F.34	Change Active Mask command	227
F.35	Change Active Mask response	227
F.36	Change Soft Key Mask command	228
F.37	Change Soft Key Mask response	228
F.38	Change Attribute command	228
F.39	Change Attribute response	229
F.40	Change Priority command	229
F.41	Change Priority response	230
F.42	Change List Item command	230
F.43	Change List Item response	230
F.44	Delete Object Pool command	231
F.45	Delete Object Pool response	231
F.46	Lock/Unlock Mask command	231
F.47	Lock/Unlock Mask response	233
F.48	Execute Macro command	233
F.49	Execute Macro response	233
F.50	Change Object Label command	234
F.51	Change Object Label response	234
F.52	Change Polygon Point command	235

F.53	Change Polygon Point response	235
F.54	Change Polygon Scale command.....	236
F.55	Change Polygon Scale response.....	236
F.56	Graphics Context command	237
F.57	Graphics Context response	241
F.58	Get Attribute Value message	241
F.59	Get Attribute Value response.....	242
F.60	Select Colour Map command	242
F.61	Select Colour Map response	243
F.62	Identify VT message.....	243
F.63	Identify VT response	244
F.64	Execute Extended Macro command.....	244
F.65	Execute Extended Macro response.....	244
F.66	Unsupported VT Function message.....	245
F.67	VT Unsupported VT Function message	245
Annex G (normative) Status Messages		246
G.1	General	246
G.2	VT Status message.....	246
G.3	Working Set Maintenance message	246
Annex H (normative) Activation messages		248
H.1	General	248
H.2	Soft Key Activation message	248
H.3	Soft Key Activation response	248
H.4	Button Activation message	249
H.5	Button Activation response.....	249
H.6	Pointing Event message.....	250
H.7	Pointing Event response	251
H.8	VT Select Input Object message.....	251
H.9	VT Select Input Object response	252
H.10	VT ESC message	252
H.11	VT ESC response.....	252
H.12	VT Change Numeric Value message	253
H.13	VT Change Numeric Value response.....	253
H.14	VT Change Active Mask message	253
H.15	VT Change Active Mask response.....	254
H.16	VT Change Soft Key Mask message.....	254
H.17	VT Change Soft Key Mask response	255
H.18	VT Change String Value message	255
H.19	VT Change String Value response.....	255
H.20	VT On User-Layout Hide/Show message	256
H.21	VT On User-Layout Hide/Show response	256
H.22	VT Control Audio Signal Termination message	257
Annex I (normative) Other messages.....		258
Annex J (normative) Auxiliary control		259
J.1	General	259
J.2	Auxiliary Inputs.....	259
J.3	Auxiliary controls in multiple VT environments.....	260
J.3.1	General rules.....	260
J.3.2	Primary VT and resolving VT function instance zero	260
J.4	Defining auxiliary inputs and functions	261
J.4.1	General	261
J.4.2	Auxiliary Function Type 1 object	261
J.4.3	Auxiliary Function Type 2 object	262
J.4.4	Auxiliary Input Type 1 object.....	263
J.4.5	Auxiliary Input Type 2 object.....	264
J.4.6	Auxiliary Function Type 2 types	265
J.4.7	Auxiliary Control Designator Type 2 Object Pointer	269

J.5	Automatic Auxiliary Control assignment	274
J.6	Manual Auxiliary Control assignment	276
J.7	Auxiliary control messages	279
J.7.1	General.....	279
J.7.2	Auxiliary Assignment Type 1 command	279
J.7.3	Auxiliary Assignment Type 1 response	279
J.7.4	Auxiliary Input Type 1 status.....	280
J.7.5	Auxiliary Assignment Type 2 command	280
J.7.6	Auxiliary Assignment Type 2 response	283
J.7.7	Preferred Assignment command	283
J.7.8	Preferred Assignment response	286
J.7.9	Auxiliary Input Type 2 Status message	287
J.7.10	Auxiliary Input Type 2 Maintenance message	288
J.7.11	Auxiliary Input Status Type 2 Enable command	289
J.7.12	Auxiliary Input Status Type 2 Enable response	290
J.7.13	Auxiliary Capabilities request	290
J.7.14	Auxiliary Capabilities response	290
J.8	Learn Mode.....	291
Annex K (normative)	Extended transport protocol	293
K.1	General.....	293
Annex L (normative)	Character sets	294
Bibliography		302

Table of Tables

Table 1 — VT Response message behavior.....	15
Table 2 — Working Set state changes (VT Supports only Active Mask)	21
Table 3 — Working Set state changes (VT Supports Multiple Working Sets or Window Masks Visible Simultaneously)	22
Table 4 — VT behaviour on mask transition	33
Table 5 — VT Reaction to navigation and data input events	35
Table 6 — VT Behavior When New Window Mask or Key Group Object is Uploaded	62
Table A.1 — Virtual terminal objects	63
Table A.2 — Allowed hierarchical relationships of objects	67
Table A.3 — Event summary	69
Table A.4 — Standard VT RGB colour palette	70
Table A.5 — Command/parameter summary	73
Table B.1 — Working Set events	78
Table B.2 — Working Set attributes and record format	80
Table B.3 — Data Mask events	81
Table B.4 — Data mask attributes and record format	82
Table B.5 — Alarm Mask events	83
Table B.6 — Alarm Mask attributes and record format	85
Table B.7 — Container events	86
Table B.8 — Container attributes and record format	87
Table B.9 — Soft Key Mask events	88
Table B.10 — Soft Key Mask attributes and record format	88
Table B.11 — Key events	89
Table B.12 — Key attributes and record format	90
Table B.13 — Button events	92
Table B.14 — Button attributes and record format	93
Table B.15 — Input events	96
Table B.16 — Input Boolean attributes and record format	98
Table B.17 — Input String attributes and record format	99
Table B.18 — Input Number attributes and record format	102
Table B.19 — Input List events	105
Table B.20 — Input List attributes and record format	107
Table B.21 — Output field events	108
Table B.22 — Output String attributes and record format	109
Table B.23 — Output Number attributes and record format	111
Table B.24 — Output List events	113
Table B.25 — Output List attributes and record format	113
Table B.26 — Output Line events	116
Table B.27 — Output Line attributes and record format	116
Table B.28 — Output Rectangle Events	118
Table B.29 — Output Rectangle attributes and record format	119
Table B.30 — Output Ellipse events	121
Table B.31 — Output Ellipse attributes and record format	121
Table B.32 — Output Polygon events	124
Table B.33 — Output Polygon attributes and record format	124
Table B.34 — Output Meter events	127
Table B.35 — Output Meter attributes and record format	127
Table B.36 — Output Linear Bar Graph events	131
Table B.37 — Output Linear Bar Graph attributes and record format	131
Table B.38 — Output Arched Bar Graph events	134
Table B.39 — Output Arched Bar Graph attributes and record format	135
Table B.40 — Picture Graphic events	137
Table B.41 — Picture Graphic attributes and record format	137

Table B.42 — Variable events	140
Table B.43 — Number Variable attributes and record format	140
Table B.44 — String Variable attributes and record format	140
Table B.45 — Font Attributes events	141
Table B.46 — Font Attributes attributes and record format	142
Table B.47 — Line Attributes events	144
Table B.48 — Line Attributes attributes and record format	144
Table B.49 — Fill Attributes events	146
Table B.50 — Fill Attributes attributes and record format	146
Table B.51 — Input Attributes events	147
Table B.52 — Input Attributes attributes and record format	148
Table B.53 — Extended Input Attributes attributes and record format	150
Table B.54 — Object Pointer events	151
Table B.55 — Object Pointer attributes and record format	151
Table B.56 — Macro attributes and record format	152
Table B.57 — Colour Map attributes and record format	153
Table B.58 — Graphics Context events	156
Table B.59 — Graphics Context attributes and record format	157
Table B.60 — Window Mask events	159
Table B.61 — Window Mask attributes and record format	160
Table B.62 — Key Group events	182
Table B.63 — Key Group attributes and record format	182
Table B.64 — Object Label Reference List attributes and record format	184
Table B.65 — External Object Definition events	185
Table B.66 — External Object Definition attributes and record format	185
Table B.67 — External Reference NAME events	186
Table B.68 — External Reference NAME attributes and record format	186
Table B.69 — External Object Pointer events	187
Table B.70 — External Object Pointer attributes and record format	187
Table B.71 — Animation events	189
Table B.72 — Animation attributes and record format	190
Table F.1 — Graphic command summary	238
Table J.1 — Auxiliary Function Type 1 attributes and record format	261
Table J.2 — Auxiliary Function Type 2 attributes and record format	262
Table J.3 — Auxiliary Input Type 1 attributes and record format	264
Table J.4 — Auxiliary Input Type 2 attributes and record format	265
Table J.5 — Auxiliary Function Type 2 types	266
Table J.6 — Auxiliary Control Designator Type 2 Object Pointer attributes and record format	271
Table J.7 — Auxiliary Control Designator Type 2 Object Pointer examples	271
Table J.8 — Set Information	291
Table L.1 — ISO 8859-1 (Latin 1) character set	294
Table L.2 — ISO 8859-15 (Latin 9) character set	295
Table L.3 — ISO 8859-2 (Latin 2) character set	296
Table L.4 — ISO 8859-4 (Latin 4) character set	297
Table L.5 — ISO 8859-5 (Cyrillic) character set	298
Table L.6 — ISO 8859-7 (Greek) character set	299
Table L.7 — WideString minimum character set	300

Table of Figures

Figure 1 — Virtual terminal — examples.....	6
Figure 2 — Operator input and control means – example	8
Figure 3 — Physical Soft Key Orientation Examples showing Key Locations	11
Figure 4 — VT virtual Soft Key paging	12
Figure 5 — Example VT which displays an active and an inactive Working Set simultaneously	21
Figure 6 — Initialization, unexpected shutdown, and expected shutdown	24
Figure 7 — Container reuse	26
Figure 8 — Container used to hide objects — Example	26
Figure 9 — External Object References — VT Example	29
Figure 10 — External Object References — Relationship Example	30
Figure 11 — Relative and absolute location of objects	31
Figure 12 — Object changed or hidden — Display update	32
Figure 13 — Clipping examples	34
Figure 14 — Graphical Extents of a Character	39
Figure 15 — 8 × 10 fonts — Example	44
Figure 16 — CR and LF application to test strings	46
Figure 17 — Rectangle line suppression and filling examples.....	48
Figure 18 — Ellipse filling examples (Without and with border line art).....	49
Figure 19 — Polygon filling examples (Without and with border line art).....	49
Figure 20 — Displaying data from multiple Working Sets - Example.....	52
Figure 21 — User-Layout Data Mask.....	53
Figure 22 — Window Mask objects - Example.....	54
Figure 23 — Window Mask Border - Example	55
Figure 24 — Key Cell layout - Examples.....	56
Figure 25 — User-Layout Data Mask with 6 Key Cells - Example	57
Figure 26 — Key object in a Key Group indicating Working Set - Example	58
Figure 27 — Key Group Objects outside of User-Layout Data Mask - Example	59
Figure A.1 — Bit positions in a bitmask.....	66
Figure B.1 — Button examples with border (Options – Bit 5 = FALSE).....	92
Figure B.2 — Button examples no border (Options – Bit 5 = TRUE).....	92
Figure B.3 — Input Boolean examples.....	97
Figure B.4 — Output Line object showing start and end points using different brush sizes	115
Figure B.5 — Output Rectangle object showing end points using different brush sizes.....	118
Figure B.6 — Output Ellipse object	120
Figure B.7 — Output Ellipse object – correct and incorrect rendering.....	121
Figure B.8 — Output Polygon types.....	123
Figure B.9 — Output Meter object.....	126
Figure B.10 — Output Meter object — examples.....	129
Figure B.11 — Output Linear Bar Graph — examples.....	130
Figure B.12 — Output Arched Bar Graph object — example	134
Figure B.13 — Effect of Line Attribute - example of same line art with different width	145
Figure B.14 — Effect of Line Attribute — example pattern: 1010.....	145
Figure B.15 — Colour Map object reverses colours – example.....	153
Figure B.16 — Example drawing with Graphics Context object.....	155
Figure B.17 — Example application of the Graphics Context object and viewport.....	156
Figure C.1 — Object pool variable length record format.....	193
Figure F.1 — Acoustic signal termination.....	216
Figure F.2 — Acoustic signal with multisound.....	216
Figure J.1 — Quadrature non-latching boolean value representation	269
Figure J.2 — Examples of Auxiliary Function references on Auxiliary Input unit Data Mask	272
Figure J.3 — Example showing expansion of a single assignment designator	272
Figure J.4 — Example showing expansion of a multiple assignment designator	273

Figure J.5 — Example showing expansion of Auxiliary Inputs on an Auxiliary Function Data Mask 273

Figure J.6 — Typical message sequence to make assignment and later remove assignment 278

Figure J.7 — Auxiliary control message flow..... 281

Figure J.8 — Auxiliary assignment screen – example..... 282

Figure J.9 — Permitted remove assignment alternatives..... 283

Figure J.10 — Preferred assignment example 286

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics*.

This third edition cancels and replaces the second edition (ISO 11783-6:2010) which has been technically revised.

ISO 11783 consists of the following parts, under the general title *Tractors and machinery for agriculture and forestry — Serial control and communications data network*:

- *Part 1: General standard for mobile data communication*
- *Part 2: Physical layer*
- *Part 3: Data link layer*
- *Part 4: Network layer*
- *Part 5: Network management*
- *Part 6: Virtual terminal*
- *Part 7: Implement messages application layer*
- *Part 8: Power train messages*
- *Part 9: Tractor ECU*
- *Part 10: Task controller and management information system data interchange*

- *Part 11: Mobile data element dictionary*
- *Part 12: Diagnostics services*
- *Part 13: File server*
- *Part 14: Sequence control*

Introduction

Parts 1 to 14 of ISO 11783 specify a communications system for agricultural equipment based on the ISO 11898 [5] protocol. SAE J 1939 [1] documents, on which parts of ISO 11783 are based, were developed jointly for use in truck and bus applications and for construction and agriculture applications. Joint documents were completed to allow electronic units that meet the truck and bus SAE J 1939 specifications to be used by agricultural and forestry equipment with minimal changes. The specifications for virtual terminals given in this part of ISO 11783 are based on DIN 9684-4 [2]. General information on ISO 11783 is to be found in ISO 11783-1.

The purpose of ISO 11783 is to provide an open, interconnected system for on-board electronic systems. It is intended to enable electronic control units (ECUs) to communicate with each other, providing a standardized system.

All phrases in this document that refer explicitly to a software term for an object or a command shall have the first letter of each object or command word capitalized (e.g. Output Linear Bar Graph object, Change Numeric Value command). This aids in the recognition of these terms as a specific item which has a specific definition in this document.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this part of ISO 11783 may involve the use of a patent concerning the controller area network (CAN) protocol referred to throughout the document.

ISO takes no position concerning the evidence, validity and scope of this patent.

The holder of this patent has assured ISO that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Robert Bosch GmbH
Wernerstrasse 51
Postfach 30 02 20
D-70442 Stuttgart-Feuerbach
Germany

Attention is drawn to the possibility that some of the elements of this part of ISO 11783 may be the subject of patent rights other than that those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 6: Virtual terminal

1 Scope

ISO 11783 as a whole specifies a serial data network for control and communications on forestry or agricultural tractors, mounted, semi-mounted, towed or self propelled implements. Its purpose is to standardize the method and format of transfer of data between sensor, actuators, control elements, information storage and display units whether mounted or part of the tractor, or any implements.

This part of ISO 11783 describes a universal virtual terminal that can be used by both tractors and implements.

Corrections in the second edition were made to Table L.2 — ISO 8859-15 (Latin 9) character set.

Requirements in the second edition were specified for two versions of the VT and Working Sets. Version 3 VTs and Working Sets meet all the requirements of the first edition, the specific requirements for version 3 of Annex G and the requirements of Annex J and Table L.2 — ISO 8859-15 (Latin 9) character set of the second edition. Version 4 VTs and Working Sets meet all the requirements of the second edition.

New requirements in this third edition are specified as version 5 VT.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 11783-1, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 1: General standard for mobile data communication*

ISO 11783-3, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 3: Data link layer*

ISO 11783-5, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 5: Network management*

ISO 11783-7, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 7: Implement messages application layer*

ISO 15077, *Tractors and self-propelled machinery for agriculture — Operator controls — Actuating forces, displacement, location and method of operation*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 11783-1 and the following apply.

3.1 auxiliary input unit

autonomous control function (CF) providing Auxiliary Controls for common use that may also be physically located within an electronic control unit (ECU), or on the virtual terminal (VT)

3.2 object pool

collection of objects that completely define the operator interface for an implement or a single Working Set

Note 1 to entry: The complete VT definition will be made up of one or more object pools — one for each Working Set.

3.3 Object ID

numeric value which identifies a specific object within an object pool

Note 1 to entry: Object ID values range from 0 to $FFFF_{16}$ (65535_{10}), with 65535 as the NULL Object ID.

3.4 attribute ID AID

numeric value which references a specific object's attribute

Note 1 to entry: AID values range from 0 to FF_{16} (255_{10}), with 255 as the NULL_AID.

3.5 char

single character where the size is 1 byte

Note 1 to entry: Commonly used for ISO 8859 characters (e.g. 41_{16} in ISO 8859-1 represents 'A') (See Annex L).

3.6 character

single text grapheme or symbol, as in an alphabet

Note 1 to entry: Size is variable based on the encoding scheme (See char and WideChar).

3.7 code plane

group of 65536 possible character codes

Note 1 to entry: Unicode/ISO10464 organizes the characters in 17 code planes numbered 0 to 16.

EXAMPLE

Code plane 0 covers characters 000000_{16} to $00FFFF_{16}$.

Code plane 1 covers characters 010000_{16} to $01FFFF_{16}$.

...

Code plane 16 covers characters 100000_{16} to $10FFFF_{16}$.

3.8 open input object

state of an input object where the object has focus and it is open for operator input

Note 1 to entry: Open input object is used interchangeably with data input.

3.9 selected input object

state of an input object where the object has focus but it is not open for operator input

Note 1 to entry: Selected input object is used interchangeably with “has focus”.

3.10

surrogate pair

32 bit code for characters composed of a 16 bit high pair and a 16 bit low pair

Note 1 to entry: UTF-16 encoding of characters in code plane 1 to 16 (See Clause 4.6.19.7 String encoding)

Note 2 to entry: UTF-16 Character encoding scheme defined by ISO10646.

3.11

WideChar

single character with a size of 2 bytes encoded in little endian order

EXAMPLE Byte sequence 41_{16} , 00_{16} represents 'A'. (See Annex L).

Note 1 to entry: Two WideChars can be combined to indicate character codes exceeding 16-bit (See Clause 4.6.19.7 String encoding).

3.12

WideString

zero or more characters composed of the primitive type “WideChar” always preceded by the byte order mark $FEFF_{16}$

EXAMPLE Byte sequence FF_{16} , FE_{16} , 41_{16} , 00_{16} , 42_{16} , 00_{16} , 43_{16} , 00_{16} represents “ABC”. This WideString has a Length of 8 bytes with the number of characters in the presentation equal to 3.

3.13

8-bit string

zero or more characters composed of the primitive type “char”

Note 1 to entry: String length is variable.

3.14

VT Number

number that is used to uniquely identify each connected VT to the operator

Note 1 to entry: See Clause F.62.

3.15

User-Layout Data Mask

special Data Masks (see Clause 4.1) that are controlled by the VT but layed out by the operator

Note 1 to entry: See Clause 4.7.

3.16

Window Cell

equally sized cell in a grid on a User-Layout Data Mask

Note 1 to entry: See Clause 4.7.

3.17

Window Mask object

supplied by the Working Set for placement by the operator into the area of one or more window cells but not a partial cell

Note 1 to entry: See Clause 4.7.

3.18

User-Layout Soft Key Mask

Soft Key Masks that are controlled by the VT but layed out by the operator

Note 1 to entry: See Clause 4.7.

3.19

Key Cell

cell that is the size of a Soft Key designator in a User-Layout Key Mask

Note 1 to entry: See Clause 4.7.

3.20

Key Group object

area of one or more Key Cells and contains a grouping of one or more Key objects

Note 1 to entry: See Clause 4.7.

3.21

Non-VT Screen

display screen that is not part of the VT application or one in which the layout is controlled by the VT

EXAMPLE A screen that comes from another application within the display. (See Clause 4.7)

3.22

Non-VT Area

visible area outside the normal Data Mask and Soft Key Mask areas

EXAMPLE A display of information related to the vehicle operation. (See Clause 4.7)

3.23

Referenced WS

working set with an object pool containing objects which are shown by another object pool via the External Object Pointer object

Note 1 to entry: See Clause 4.6.11.6.

3.24

Referencing WS

A working set with an object pool which shows object(s) from another object pool via the External Object Pointer object

Note 1 to entry: See Clause 4.6.11.6.

3.25

Functionally Identical WS

Working Set(s) with a NAME that exactly matches other Working Sets, when the Self Configurable, Instance fields, and Identity Number are excluded in the comparison

3.26

Line End

"cursor" or text positioning control intended to locate the following displayable character "font height" pixels downward and at the left-most position in the containing object

Note 1 to entry: See Clause 4.6.19.6.

4 Technical requirements

4.1 Overview

A virtual terminal (VT) is a control function (CF) within an electronic control unit (ECU), consisting of a graphical display and input functions, connected to an ISO 11783 network that provides the capability for a CF, composing an implement or a group of implements to interact with an operator. The VT provides the capability to display information and to retrieve data from an operator. The CF, as an implement or a group of implements represented by a Working Set Master acquires storage for objects within the VT and on demand displays this stored information to an operator. In this part of ISO 11783, the term *Working Set* will be used for a CF, as an implement or a group of implements either represented by a single ECU or a group of ECUs acting as a Working Set. Working Sets on the network can also acquire the use of input methods of the VT to allow the operator to send signals back to the Working Set.

This part of ISO 11783 describes the VT with the detail and clarity required for VTs built by different manufacturers to be interchangeable with any implement Working Set that uses its services. The interface protocol of this part of ISO 11783 also reduces the run-time ISO 11783 communication bus traffic as much as possible. For these reasons, the requirements of this part of ISO 11783 are organized in an object-oriented manner with specific attributes and behaviour of each object clearly and fully defined. The required behaviour of the VT given certain situations is also detailed.

In general, the functions, not the design, of the user interface of the VT are defined in order to avoid restrictions on possible designs. However, certain limitations are imposed in order to meet the goal of interchangeability between various manufacturers. Specifications regarding physical layout, components, processing power and the number of physical elements comprising a VT have been omitted in order to avoid restricting manufacturer's designs.

The VT shall have a pixel-addressable (graphical) display. Information from connected Working Sets is shown to the operator on the graphical display. This information is shown in display areas that are defined by Data Masks, Alarm Masks and Soft Key Masks. The data for these masks is contained in object definitions that are loaded into a VT via the ISO 11783 CAN bus, or from non-volatile memory. When the information defined by a mask is required on the display, the mask can be made visible by a single Change Active Mask command from the Working Set, and therefore does not require significant additional network traffic.

The physical size, resolution, orientation and methods of implementing the graphical display are at the discretion of the designer of the VT. Figure 1 — Virtual terminal — examples shows examples of some possible VT designs and orientations.

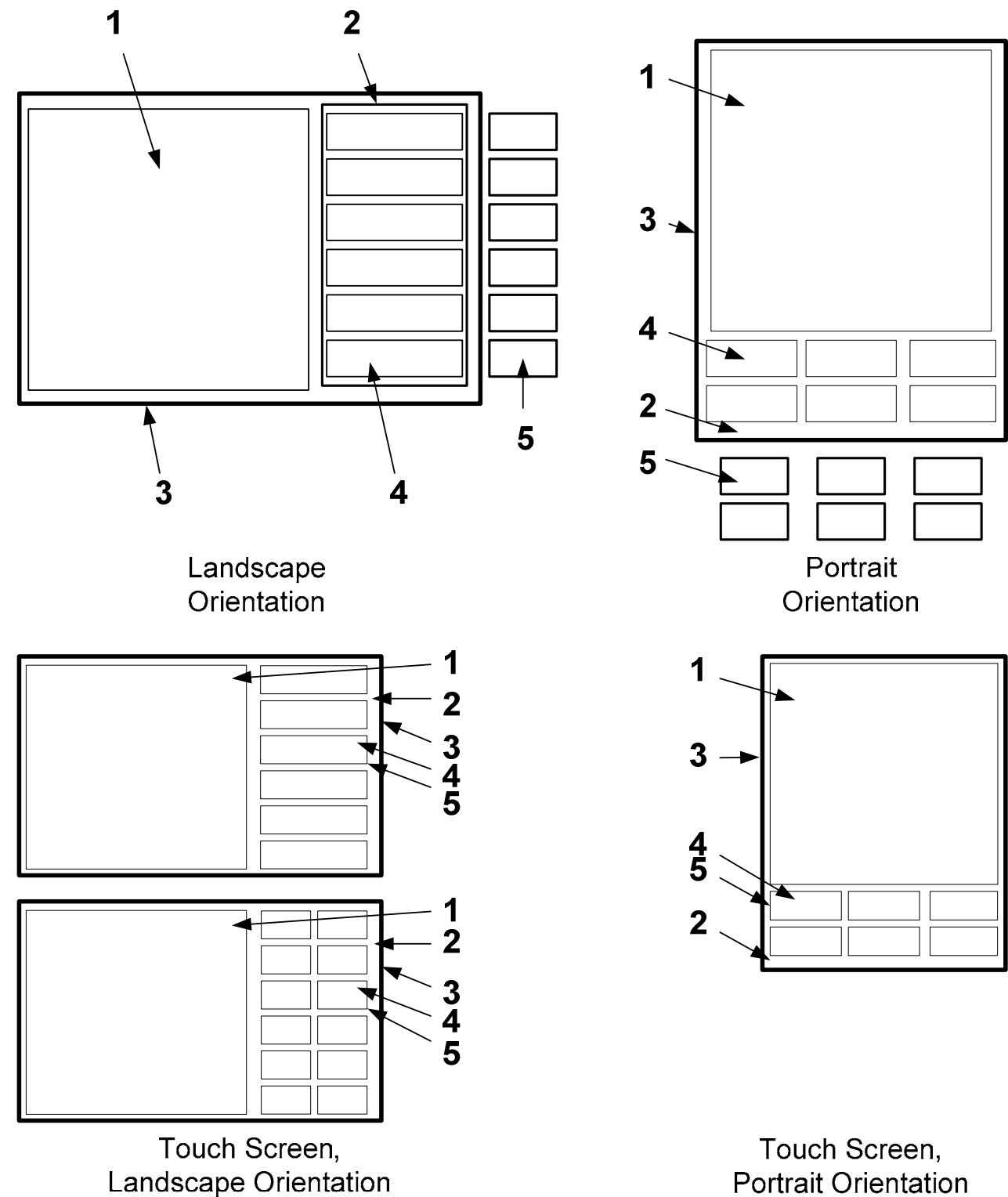


Figure 1 — Virtual terminal — examples

4.2 Operator input and control

The VT shall provide the operator with means for control and input. There are five means associated with a VT that can be used for the input of data, selection of display data, and the control of connected Working Sets.

See Figure 2 — Operator input and control means

a) **Soft**

is a means, most likely keys on the VT, using software-changeable designators (labels). “Soft Keys” have their identity changed depending on which Soft Key Mask is visible. The VT shall make the association between a Soft Key and its designator clearly evident to the operator.

b) **Navigation**

is a means of selecting an input field or Button within the active Data Mask. If keys are used for “Navigation”, they do not send key activation information to the Working Set and are proprietary to the VT.

c) **Data Input**

is a means of entering/editing information in an input field within the active Data Mask. If keys are used for “Data Input”, they do not send key activation information to the Working Set and are proprietary to the VT. A means shall be provided for entering any number or character sequence that is valid for the input field.

During the data input operation, the VT Status message will continue to indicate the active Working Set, and active mask which contains the input object for which the data input operation applies. Data input operation that originates on a User-Layout Data Mask does not affect the VT Status message.

There are two types of Data Input — “editing” and “real time editing”.

1) **Editing**

is a means of data input where the new value being entered is composed by the operator using a proprietary means within the VT. During the composition of the new value, changes to the original value are not communicated to the Working Set. A means shall also be provided for ESC from or ENTER of information into a data field.

The ENTER means shall be provided to indicate to the Working Set the completion of data entry and communication of the new value, and the ESC means shall be provided to indicate that the data entry was aborted. The ENTER and ESC means may either be a permanent key or may only be available during data entry. (See Table 5 — VT Reaction to navigation and data input events) The VT shall send a VT ESC message to a Working Set for an operator-activated ESC means or an ESC response as a response to receiving an ESC command from a Working Set.

2) **Real Time Editing**

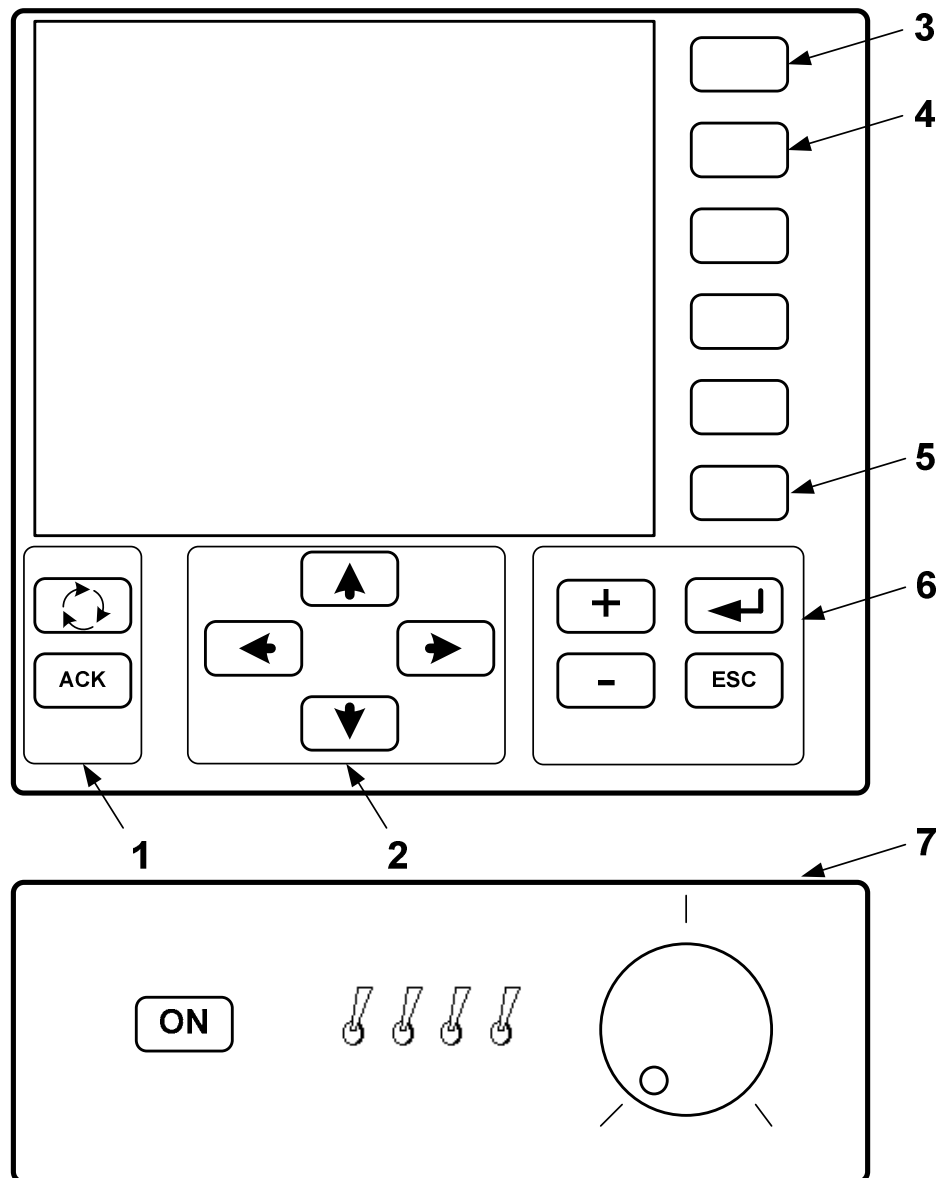
is a means of data input for an Input Number object and Input List object where the object has focus and it is open for operator input and changes by the operator to the value are periodically transmitted to the Working Set while the object is being changed. The VT Change Numeric Value message is limited to a 5 Hz update rate. Each value change sent to the Working Set is considered a complete transaction, as if the ENTER means was activated, and cannot be reverted by the ESC means. The VT is not required to provide steps in uniform increments, however it shall be possible to set any value (e.g. fast scrolling is allowed to span a wide range of values, with fine adjustment for final setting). If the ESC means is activated during real time editing, the VT shall ensure that the on-screen value is equal to the value last sent to the Working Set. The VT may send a final value to the Working Set prior to sending the VT ESC message, or ESC response message to ensure this synchronization. Real time editing shall meet the operator controls requirement specified in ISO 15077.

d) **Control**

is a means of selecting between Working Sets whenever a Data Mask is available and for acknowledging alarms. Both means are required. Since more than one Working Set can use the services of the VT, the VT shall provide a means for the operator of selecting between connected Working Sets. The Working Set selection means should be indicated by three circular arrows or a similar graphic. Only the ACK means sends key activation information to the Working Set.

e) **Auxiliary Input**

is a means available to the operator for communicating input commands to the Working Set(s) using Auxiliary Controls which are assigned to Auxiliary Functions. (See Annex J)



Key

- 1 control
- 2 navigation
- 3 Soft Key 1
- 4 Soft Key 2

- 5 Soft Key 6
- 6 data input
- 7 auxiliary input

Figure 2 — Operator input and control means – example

4.3 Acoustic alarm

The VT shall provide an acoustic alarm. The alarm may be a simple on/off type buzzer or an acoustic component capable of either/or variable frequency and audio level. (See Clause D.9 Get Hardware response)

4.4 Coordinate system

Positions and sizes in this part of ISO 11783 are always given in physical pixels unless otherwise stated. A two-dimensional coordinate plane (x, y) is used, where x is the number of units wide (x increases from left to right) and y is the number of units high (y increases from top to bottom). The coordinates are signed values. The origin (0, 0) for any object's coordinate system is located at the top left-hand corner of the parent object.

4.5 Display areas

4.5.1 General

This section defines standard Data Mask and Soft Key Mask areas of the display. Alternate usage of this area supports displaying data from multiple working sets. (See Clause 4.7)

4.5.2 Data Mask

The VT shall reserve an area of the display for displaying Data Masks and Alarm Masks. This area is called the Data Mask area (See Figure 1 — Virtual terminal — examples). Recognizing that the physical orientation of the VT display could be different, depending on the manufacturer of the VT, a square data mask aspect ratio is chosen to ensure correct display in either landscape or portrait orientation. The minimum Data Mask area shall be 200 pixels × 200 pixels. This requirement does not limit the physical resolution or size of the display, only the useable Data Mask area. Higher resolution mask areas are permitted, but the square aspect ratio shall be strictly enforced. Examples of Data Mask areas that would meet this requirement are:

- 200 × 200,
- 240 × 240,
- 320 × 320, and
- 480 × 480.

Any other square dimensions would be acceptable.

It is suggested that unused areas of the physical display be used for proprietary information such as vehicle data, VT statistics or other data.

4.5.3 Soft Key Mask area and Soft Key designators

4.5.3.1 Soft Key variants and navigation

The VT shall reserve an area of the display for Soft Key labels, separate from the Data Mask area. This area is called the Soft Key Mask area (See Figure 1 — Virtual terminal — examples). Each Soft Key shall have a reserved display area, called a Soft Key designator, for displaying a label (See Figure 1 — Virtual terminal — examples). The minimum size of the designator field is 60 pixels wide × 32 pixels high regardless of screen orientation. The Soft Key designators may contain text, graphics or both. The Soft Key Mask area may be adjacent to, or physically separate from, the Data Mask area, but shall not be part of the Data Mask area.

The VT shall provide a clearly visible separation between the individual Soft Key designators (for example by drawing a one-pixel line). This visible separation shall be drawn outside of the Soft Key designator area. Only if the minimum size of the designator field cannot be fulfilled due to this requirement, the drawing of a one-pixel line on the border of the Soft Key designator area is acceptable.

The presentation of the Soft Keys can be further described in three groups, with a defined relationship:
Navigation Soft Keys < Number of Physical Soft Keys <= Number of Virtual Soft Keys.

- a) VT Version 3 and prior VTs have no requirement on the number of physical Soft Keys.
- b) VT Version 4 and later VTs shall provide at least 6 physical Soft Keys.
- c) VT Version 3 and prior shall support a maximum of 64 virtual Soft Keys per Soft Key Mask and shall support as a minimum the number of reported physical Soft Keys (see Clause 4.5.3.3).
- d) VT Version 4 and later shall support exactly 64 virtual Soft Keys per Soft Key Mask (see Clause 4.5.3.3).
- e) The VT shall provide a means for the operator to navigate and select all defined Soft Keys. For example, if there are six physical keys, some type of paging would be required to allow the operator to navigate to, and select from, any of the 64 virtual Soft Keys using the six physical keys.

4.5.3.2 Physical Soft Keys

Physical Soft Keys is the count of the number of permanently dedicated keys that the VT makes available to active Working Sets. The term “physical Soft Key” does not imply that the VT must provide physical buttons for the Soft Keys. For example on a VT with touch screen, the physical Soft Keys may be located directly on the touch screen as shown in Figure 1 — Virtual terminal — examples.

For VTs with a vertical arrangement of physical Soft Keys, key number 1 shall be on the right and the top-most position. Key number 2 shall be adjacent and below Key 1. Key m shall be at the bottom of the first column. If there are additional physical Soft Keys, the column containing keys m+1 to key n shall be to the left of the first column. Each additional column of physical Soft Keys shall continue to the left.

For VTs with a horizontal arrangement of physical Soft Keys, Key number 1 shall be on the top row and in the left-most position. Key number 2 shall be adjacent and to the right of Key 1. Key m shall be at the far right of the top row. If there are additional physical Soft Keys, the row containing keys m+1 to key n shall be below the first row. Each additional row of physical Soft Keys shall continue below the previous row. Examples of these arrangements are shown in Figure 3 — Physical Soft Key Orientation Examples showing Key Locations.

For VTs without a clear horizontal or vertical arrangement of physical Soft Keys (e.g. physical Soft Keys located in a matrix on the touch screen) the rules for a VT with a vertical arrangement of physical Soft Keys apply.

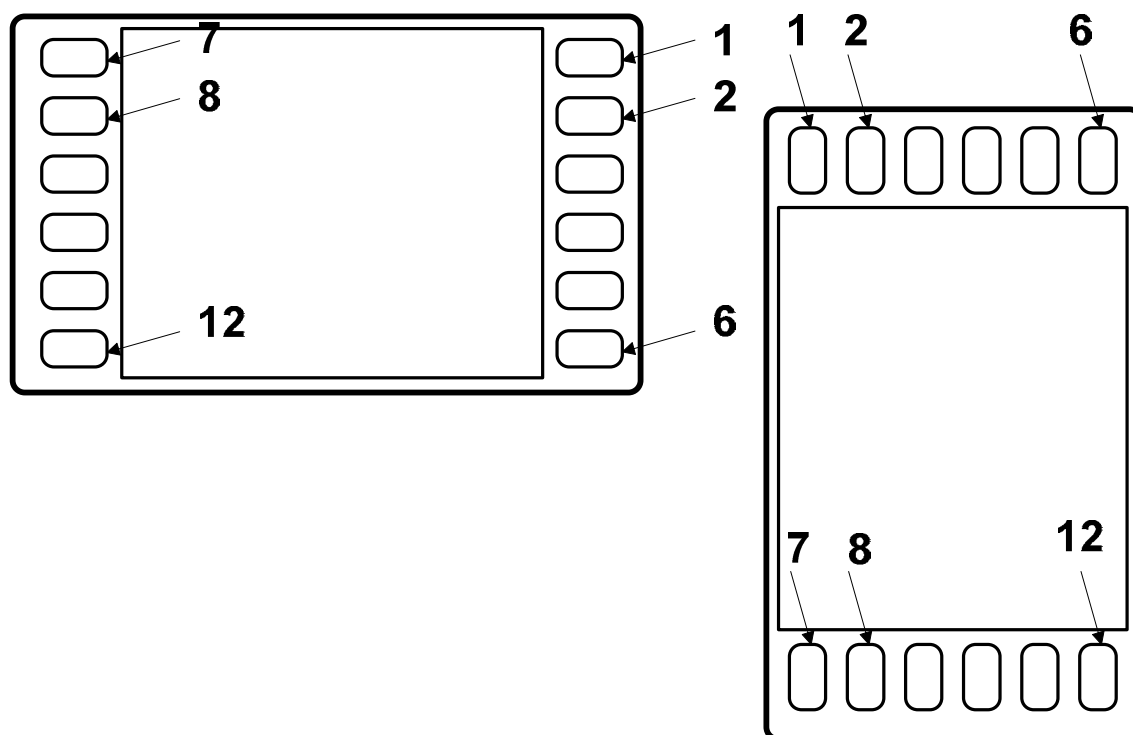


Figure 3 — Physical Soft Key Orientation Examples showing Key Locations

4.5.3.3 Virtual Soft Keys

Virtual Soft Keys is the count of the number of Soft Keys that the VT supports for each active Working Set's Data Mask. If the physical Soft Keys count is less than the virtual Soft Keys count, the VT shall provide a means for navigation to allow the operator to choose from any of the Working Sets Soft Keys.

4.5.3.4 Navigation Soft Keys

Navigation Soft Keys is the count of the number of physical Soft Keys that the VT may allocate for the purpose of navigation among the Soft Keys. The number of navigation Soft Keys shall be less than the number of physical Soft Keys. If the VT provides other means of navigation that does not use the physical Soft Keys, this value shall be zero.

4.5.3.5 Navigation among Soft Keys

If the Working Set provides a number of Soft Keys on a Soft Key Mask equal to or less than the number of physical Soft Keys reported by the VT, then all of the Soft Keys on this Soft Key Mask shall be accessible with the physical Soft Keys. The VT shall not provide any navigation means for this Soft Key Mask.

If the Working Set provides more Soft Keys on a Soft Key Mask than the VT has reported in the number of physical Soft Keys, the VT shall provide navigation for that Soft Key Mask. This navigation among the Soft Keys shall be done by paging through the Soft Keys in groups, not by scrolling. Further, a "group" is defined as the "physical Soft Keys" count minus the "navigation Soft Keys" count. The navigation Soft Keys shall always occupy the same physical Soft Key positions on all pages, although the VT designer may choose to disable (but not remove) the navigation keys on certain pages. The last set of virtual Soft Keys (depending on how many Soft Keys the Working Set provided to the VT) may not completely fill the Soft Key Mask. The remainder of the Soft Key designators shall not be used.

As described in section B.5 Soft Key Mask object and illustrated in Figure 4 — VT virtual Soft Key paging, pointers to the NULL Object ID reserve a Soft Key position. Pointers to NULL Object ID that are at the end of the list of Soft Keys shall not reserve a Soft Key position and shall not be considered for paging or navigation.

EXAMPLE As shown in Figure 4 — VT virtual Soft Key paging (a in figure), a VT is designed with 6 physical Soft Keys, 64 virtual Soft Keys, and 1 (a in figure) navigation Soft Key. The Working Set provides 18 Soft Keys to the VT, however there are 3 which are Pointers to the NULL Object ID. To support navigating among the Soft Keys the VT designer alters Soft Key 6 into a “next Soft Key group” button. A navigation group is calculated as sets of 5 Soft Keys (a in figure), starting with the first Soft Key. When the navigation key is pressed, the VT shows the next group of Soft Keys. Another example (b in figure) shows a similar example with 2 navigation Soft Keys. Another example (c in figure) shows an arrangement with two columns of keys and two navigation keys. If the VT provides dedicated navigation keys, the number of navigation Soft Keys reported shall be zero (d in figure).

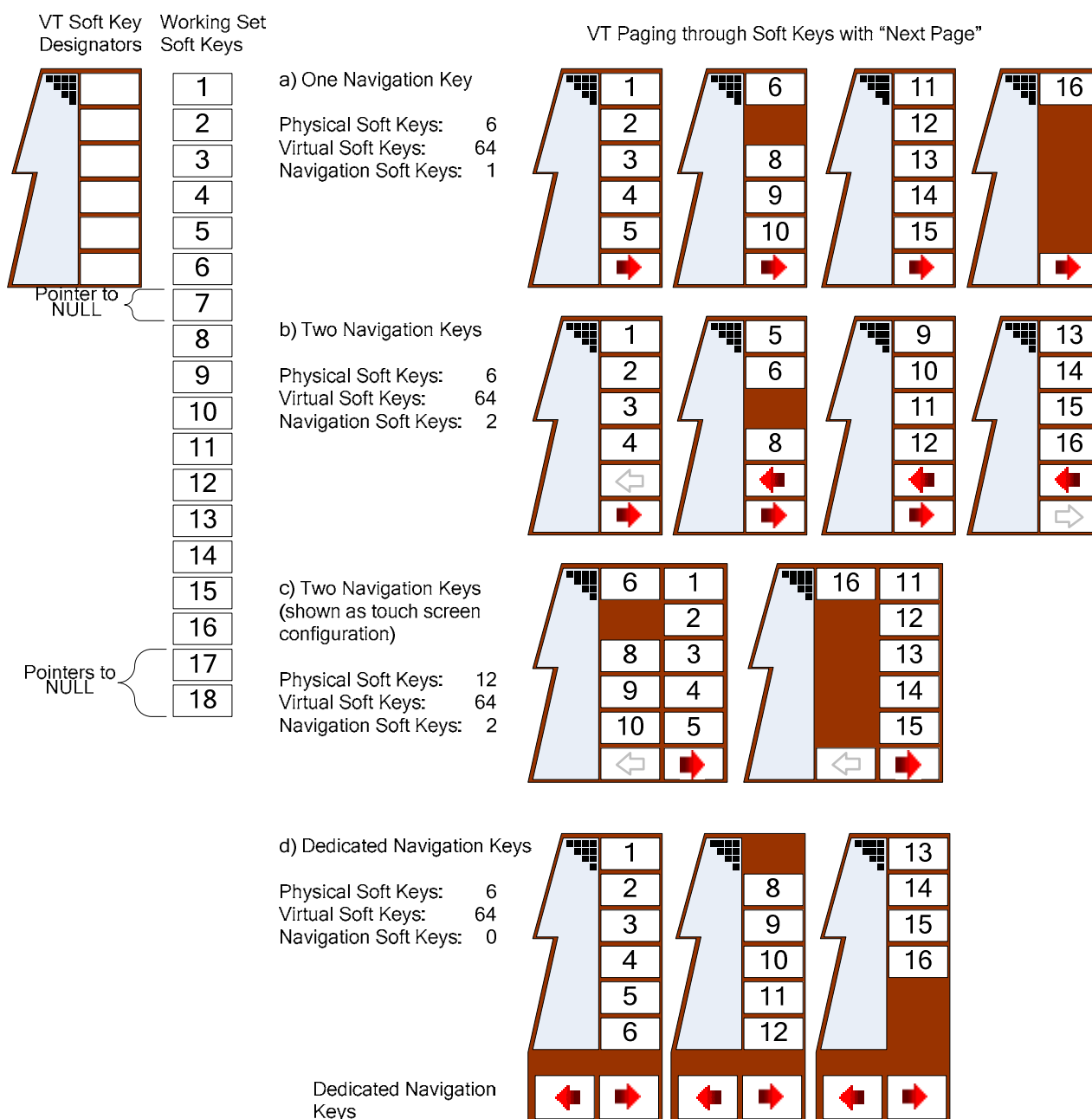


Figure 4 — VT virtual Soft Key paging

4.6 Behaviour

4.6.1 Object pools

4.6.1.1 General

The operator interface definition for a device of one or more implements represented by either a single ECU or a Working Set consists of a set of objects (hereafter referred to as the Working Set's *object pool*). These objects are defined in detail in Annex B and Annex J. Each object contains all necessary attributes and child object references for processing the object to completion. The Working Set assigns a unique Object ID to each object in its object pool so that each object is uniquely addressable. Object IDs shall be unique within a single Working Set's object pool but may not be between different Working Sets.

The object pool is transferred to the VT at initialization by using the procedure described in Annex C. The VT is intended to be capable of storing the object pools in a modifiable memory area. VTs may store multiple pools of a Working Set, in non-volatile memory, if they have unique version labels. For example, multiple pools that differ only by language. All objects shall be fully described before they are made active in a mask on the display.

The behaviour of the VT when no object pools are loaded is proprietary.

4.6.1.2 “NULL” Object ID

Object ID FFFF₁₆ (65535₁₀) is reserved for use as the “NULL” Object ID.

4.6.1.3 Processing objects

Objects listed in parent objects may also list child objects, thereby creating a tree hierarchy in the object pool. Objects are always processed in the order listed in the parent object in a “depth-first” manner. In other words, if a reference is made to an object that references other objects, the child references are processed to completion before returning to the parent to continue processing.

VT version 5 and later VTs shall support a minimum hierarchy depth of 30 objects. For VT version 4 and prior VTs, the requirement is unspecified.

The hierarchy depth is computed starting with the following objects; Data Mask object, Alarm Mask object, Window Mask object, Soft Key Mask object, Key Group Objects, and increments by one to reach a child object. For the child objects that have child objects the depth increments again. This process continues to the last child object. For computing the hierarchy depth, the object to which a pointer object references is counted as a child object.

The relationship from the Working Set object to either a Data Mask object or an Alarm Mask object, and the relationship from those to a Soft Key Mask object are not included in the count.

4.6.2 Working Sets

The Object Pool supplied by a Working Set Master is associated with all members of that Working Set. This allows object information from one CF or all the CFs that make up a Working Set to be collectively presented as a common object pool. One CF shall be designated as the Working Set Master for each Working Set. As coordinator of the communications of a Working Set, the Working Set Master shall secure the use of the VT and provide the object pool definition. It shall also send Working Set messages that provide the NAMES of the members of said Working Set to the VT. This identifies the members of the Working Set and hence those CFs which can communicate to the VT. Appropriate messages for defining a Working Set are given in ISO 11783-7.

Once members of the Working Set have been identified and after the object pool has been loaded into the VT, any member of the Working Set has the ability to provide data for objects and to change attributes in the object pool during run-time.

The Working Set Master shall provide the initial object pool definition. Any data input by the operator into input field objects is always transmitted to the Working Set Master.

The VT shall never have Working Set Members and shall not transmit the Working Set Master or Working Set Member messages (See ISO 11783-7).

The handling of VT Response messages defined herein supersedes part ISO 11783-1 in reference to responses being directed only to the Working Set Master. See Table 1 for — VT Response message behavior to Working Set messaging.

Table 1 — VT Response message behavior

Configuration	Working Set Version ^a	VT Version ^b	Behaviour
1	3 and prior	3 and prior	VT response to any command is directed to the WS Master
2	3 and prior	4 and later	VT response to any command is directed to the WS Master
3	4 and later	3 and prior	VT response to any command is directed to the WS Master
4	4 and later	4 and later	VT response to any command is directed to the originator
^a Working Set Version is reported in the Working Set Maintenance message.			
^b VT Version is reported in the Get Memory response message.			

In configurations types 1 through 3, the Working Set Member has the responsibility to monitor all [destination specific] VT to Working Set Master messages in order to pair its commands with responses. The Working Set Master will receive unsolicited responses from the VT (which were originated by its members), and will not be able to pair these with messages the master originated. The Working Set Members also will not be able to pair the messages correctly when originating from another member or from the master.

In configuration type 4, all responses from the VT are directed to the originating nodes. Responses that are communicated via Transport Protocol are now possible (e.g. Get Supported WideChars response). Further, the Working Set Master no longer receives unsolicited response messages. Working Set Members no longer have an obligation to monitor destination specific messages directed to another address.

In order to maintain backward compatibility, Working Sets shall not send higher version messages to a lower version VT (e.g. a version 4 command sent to a version 2 VT). How a lower version VT would respond in such a case should be considered unpredictable. For example some VT designs might respond with a NACK message, others might ignore the message. In extreme cases this could cause a software crash or reset at the VT.

Conversely, the VT shall not send higher version messages to a lower version Working Set (e.g. a version 4 event sent to a version 2 Working Set). How a lower version Working Set would respond in such a case should be considered unpredictable. For example some Working Set designs might respond with a NACK message, others might ignore the message. In extreme cases this could cause a software crash or reset at the Working Set.

VT version 5 and higher VT's and Working Sets shall support the VT Unsupported VT Function message, and Unsupported VT Function message, respectively. With this message the VT and Working Set respond in a predictable way. VTs and Working Sets, designed for VT version 4 and prior, may implement these messages.

Additional compatibility information is defined in clause 4.6.24.

4.6.3 Multiple Visually Similar Working Sets

When more than one visually similar Working Set from the same manufacturer becomes part of a network, these Working Sets should be uniquely identified to correlate each instance with a location. This shall be accomplished using an Instance field of the NAME (e.g. 2 sprayers from the same manufacturer, or 2 or more visually similar Auxiliary Input units).

For consistent system configuration, the Working Sets should be arranged with the lowest to highest instance from left to right followed by front to rear followed by bottom to top.

The manufacturer shall provide the operator a means to establish a working configuration, using one or more of the following methods or via some other means not specified here.

The operator can correctly locate the Working Set based on its instance with:

- An indication on the label of the Working Set unit identifying its Instance,
- By physical location, via a wire in the harness of the Working Set that automatically sets its instance in increasing values left to right followed by front to rear followed by bottom to top.

The operator can set the Instance based on its location with:

- An operator accessible “Instance” switch on the Working Set unit,
- An operator accessible “Instance” setting, such as on a Data Mask,
- Use of the commanded name message (See ISO11783-5) with a provided service tool.

4.6.4 Displayed Working Set number

When more than one visually similar Working Set exists on a network, the Working Set shall indicate its working set number on its Working Set object. Additionally, the Working Set should indicate its working set number on visible masks.

The displayed Working Set number shall be defined by the manufacturer. The Working Set number should be related to the Function Instance, the Device Class Instance, and/or the ECU Instance, as defined by the manufacturer (See ISO11783-5). Other factors defined by the manufacturer may also be used to ensure the Working Set is uniquely identifiable. All visually similar equipment from the same manufacturer shall apply the same relationship.

Example 1 working set number = working set Function Instance + 1

Example 2 working set number = working set Device Class Instance + 1

Example 3 working set number = working set ECU Instance + 1

4.6.5 Language, formats and measurement units selection

The VT(s):

- Shall send the standard language, format and measurement units messages defined in ISO 11783-7, hereafter “standard setups”. The Working Set object identifies the languages that the Working Set supports. The VT shall provide a method for the operator to view the list of supported languages and to select an item from the list. If no language has been entered by the operator (as would be the case in a factory-new VT), the VT shall attempt to query the default language from the tractor ECU. Once the operator has set the language, The VT’s language message always takes priority over the tractor ECU’s language.
- Shall also provide a method for the operator to select formats (Time, Date, etc.) and measurement units. The VT shall report selected language, formats and measurement units at power up and any time there is a change. These messages allow the Working Set to modify its object pool to the operator-selected language (.e.g. by updating string fields, selecting units of measure, changing offsets and scales, etc.).
- Shall store the standard setups in non-volatile storage and restore the values during initialization.
- Shall respond to ISO11783-7 “Language Command” requests sent to the global address

- Shall respond to ISO11783-7 “Language Command” requests directed to this VT

The Working Set(s):

- Shall configure their standard setups according to the VT to which they are publishing the pool(s). This can cause different standard setups to be published to different VTs. (e.g. auxiliary objects are published to VT with Function instance 0 and the remainder of the pool to other VTs)

Shall use a proprietary method to select an appropriate (or default) setting if the Working Set does not support the selected language, formats or units.

4.6.6 Initialization

Upon power-up or reinitialization, a specific sequence of events shall occur in order to ensure proper initialization of the VT and Working Sets, as follows.

4.6.6.1 VT initialization

- 1) The VT shall complete the address claim procedure in accordance with ISO 11783-5 and shall also send an address claim request to the global destination address (255).
- 2) The VT shall begin transmission of the VT Status message. In the case of a reset or recovery, the VT shall ensure that greater than 3 seconds have elapsed between this initial VT Status message and the previous VT Status message.
- 3) If language selection has not been entered by an operator, the VT shall attempt to request the default language setting from the tractor ECU.
- 4) The VT shall allow Working Sets to initialize and to load their object pools.

4.6.6.2 Working Set initialization with VT

- 1) The Working Set, if equipped with Auxiliary Functions, shall clear any assignments in volatile memory.
- 2) The Working Set Master (and Working Set Members) shall complete the address claim procedure in accordance with ISO 11783-5.
- 3) The Working Set Master shall wait until the VT begins transmission of the VT Status message.
- 4) The Working Set Master shall identify itself and its members to the VT using messages given in ISO11783-7.

NOTE 1 The Working Set Master may send these messages for other purposes (e.g. Task controller initialization).

NOTE 2 If the Working Set Master has a need to reconfigure the list of Working Set Members after the initialization is complete, the Working Set Master shall send the Working Set Master and Working Set Members messages. The Working Set Master may use this to add or remove members from the set. No Working Set initialization is required.

- 5) The Working Set Master shall transmit the Working Set Maintenance message once with the ‘initiating flag’ set to 1 (when designed for version 3 and later VTs).

If the VT had previously detected a shutdown and was transmitting the NACK in response to the Working Set Maintenance message (See Clause 4.6.9 Connection management), there are two cases where the VT shall stop transmitting the NACK:

- I) In the case of a version 3 or later ECU, identified by Working Set Maintenance message; Byte 3 < 255 and the initiating flag Byte 2 Bit 0 = 1.
 - II) In the case of a version 2 and prior ECUs, identified by Working Set Maintenance message; Byte 2 = FF₁₆, and the VT received a Working Set Master message since the prior maintenance message.
- 6) The Working Set Master shall begin transmitting the Working Set Maintenance message with the 'initiating flag' set to 0 (when designed for version 3 and later VTs).
- 7) The Working Set Master may request the language and format messages from the VT (See ISO 11783-7) if it has not already received this message from the VT and the Working Set has presentation that is language or unit specific.
- 8) The Working Set Master may query the VT as necessary to determine its capabilities. Based on the VT's responses, the Working Set Master shall adjust its object pool for scaling, available fonts, supported colours, etc.
- 9) The Working Set Master may query the VT to determine if its object pool already exists in non-volatile memory.
- 10) Object pool transfer shall commence and be completed. This can be done either by asking for the object pool to be transferred from non-volatile memory (See Annex E) or by using the protocols detailed in Annex C.

4.6.6.3 Working Set initialization on networks with multiple VTs

A Working Set Master shall have a means to perform a "Move to another VT" function on networks with multiple VTs. This function shall allow movement of the Working Set to each of the available VTs in sequence. For example, this function could be accomplished with a "Next VT" Soft Key or Button in the user interface and/or in combination with the Identify VT message. The function behaves as follows:

- 1) "Move to another VT" is enabled if the Working Set Master detects more than one VT on the network.
- 2) When "Move to another VT" is activated, the Working Set Master:
 - I) Puts itself in a safe state, or prevents activation of this feature unless it is in a safe state.
 - II) Shall send the Delete Object Pool command to the VT and wait for the response.
 - III) Shall stop sending the Working Set Maintenance message to the VT.
 - IV) Starts the initialization process with another VT on the network.
 - V) The Working Set Master shall save the new VT as the preferred VT for a next power cycle. If the preferred VT is not available within a certain time period after startup, the Working Set Master may initialize connection to any other VT on the network. The Working Set may provide a means for the operator to set the maximum wait time period or it may be obtained from the boot time specification in the "Get Hardware response" message of the preferred VT.

4.6.7 System Shutdown

4.6.7.1 General

In this context "System Shutdown" is defined as the period of time when the Key Switch state indicates the key is off and yet ECU Power remains on. Actuator Power may or may not remain on concurrent with ECU Power (see ISO 11783-7).

When the Key Switch state indicates the key has been turned off, and while ECU Power has not been terminated, it is expected that systems may transition to a shutdown state that is appropriate for that system. In some devices, this may cause immediate termination of all network communications, where other devices may request power to remain on for a more orderly shutdown. Others still may ignore the Key Switch state and continue normal operation until power is interrupted.

The relevant PGNs defined in ISO 11783-7 for determining the Key Switch state is the Wheel-based speed and distance (PGN 65096) and for requesting power maintenance is the Maintain power (PGN 65095).

The following are recommended practices.

4.6.7.2 VT behavior

The VT can expect that applications on the network may terminate communications without warning.

A recommended behavior of the VT is to monitor the Key Switch state and take the following actions as a result of the transition from "Key switch not Off" to "Key switch Off".

- 1) The VT should disable unexpected shutdown detection logic to avoid unnecessary notification to the operator as a result of one application shutting down immediately while another maintains the ECU Power beyond the normal 3 second timeout (see Clause 4.6.9).
- 2) The VT should maintain services while "Key Switch Off" and for a minimum of 2 seconds following the last "Maintain ECU Power" request from those ECUs which have object pools in the VT volatile memory.
- 3) The VT should continue to monitor the Key Switch state and reinitialize if turned from "Key switch Off" to "Key switch not Off", ensuring that if the VT Status message was discontinued, the standard Initialization process is performed (see Clause 4.6.6).

NOTE VT version 3 and prior did not specify shutdown behavior, therefore, these VTs may discontinue all communications with the network, including discontinuing the VT Status message.

4.6.7.3 Working Set behavior

Working Set behavior may vary significantly depending on the design of the specific set.

One variation of a Working Set design may not monitor the Key Switch state and may continue as normal until power is lost.

A recommended behavior of a Working Set is to monitor the Key Switch state and take the following actions as a result of the transition from "Key switch not Off" to "Key switch Off":

- 1) The Working Set may send a "Maintain Power" message (see ISO11783-7) to inform the system of the state of the Working Set, and optionally as the means to request power be maintained.
- 2) The Working Set may monitor the "Maximum time of tractor power" parameter (see Wheel-based speed and distance message in ISO11783-7) and use this information during any power management processes it may execute.
- 3) The Working Set may send a Delete Object Pool command to the VT to eliminate the possibility of an unexpected shutdown indication (see Clause 4.6.9).
- 4) The Working Set should not consider the lack of the VT Status message or other VT to ECU messages as an unexpected shutdown of the VT, and therefore should not attempt a connection to any other VT as may be available.

- 5) The Working Set should continue to monitor the Key Switch state and reinitialize if turned from "Key switch Off" to "Key switch not Off". (See Clause 4.6.6).

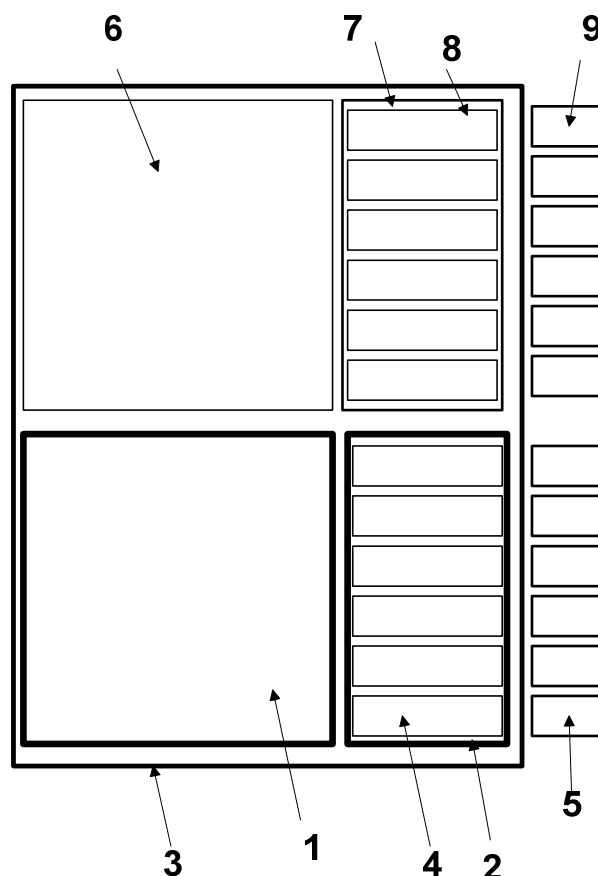
4.6.8 Working Set object and active masks

In the initial object pool definition, each Working Set Master shall provide one, and only one, Working Set object in order to define a descriptor, active mask and supported languages for the Working Set. The descriptor may be graphical; text or both but shall fit inside the area defined by the VT for a Soft Key designator. Any object or part of an object located outside of the Working Set descriptor shall be clipped. The descriptor may be used by the VT any time the Working Set needs to be represented to the operator.

EXAMPLE Communication alarms, Auxiliary Control setup.

When a Working Set is "active", it has exclusive input focus and is displayed on the VT display. When the Working Set is "inactive", it may also be visible on the VT display but does not have input focus. The VT shall provide some means to allow the operator to select the Working Set that is to be active. Only one Working Set is active at any given time. The Working Set cannot force any of its masks to be visible when the Working Set is not visible, and it cannot force its Working Set to be active when another Working Set is active. However, in some cases, setting the active mask to an Alarm mask may make the Working Set visible, but that is not guaranteed.

For VT version 4 and later, a VT may also display one or more Working Sets which are not active in addition to the Active Working set. (See Figure 5 — Example VT which displays an active and an inactive Working Set simultaneously). The VT uses the VT On User-Layout Hide/Show message to inform the inactive Working Set to update its Data Mask and or Soft Key Mask when it is visible. If a Working Set responds with a NACK or with a hidden state for the corresponding Data Mask or Soft Key Mask then the VT knows that the Working Set does not support this feature. If the Working Set does not support this feature, the VT shall inform the operator that the displayed information may not be updated. The VT may still display the inactive Working Set, because the inactive Working Set may update its data. (See Clause 4.6.10 Updating the operator interface)



Key

- | | |
|---|---|
| 1 Data Mask area of active Working Set | 6 Data Mask area of inactive Working Set |
| 2 Soft Key Mask area of active Working Set | 7 Soft Key Mask area of inactive Working Set |
| 3 Physical Screen | 8 Soft Key Designator of inactive Working Set |
| 4 Soft Key Designator of active Working Set | 9 Physical Soft Key of inactive Working Set |
| 5 Physical Soft Key of active Working Set | |

Figure 5 — Example VT which displays an active and an inactive Working Set simultaneously

Table 2 — Working Set state changes (VT Supports only Active Mask)

Working Set state change	VT Behavior
Active to Inactive	1. Hide the Working Set's currently active Data/Alarm Mask and associated Soft Key Mask. 2. Send the VT Status message to the global address (255) to inform Working Sets of the change.¹
Inactive to Active	1. Display the Working Set's currently active Data/Alarm Mask and display the associated Soft Key Mask. 2. Send the VT Status message to the global address (255) to inform Working Sets of the change¹
¹ When the state of a Working Set changes from inactive to active and this causes the state of another Working Set to go from active to inactive, there shall be only one VT Status message (not two), which shall specify the new active Working Set.	

Table 3 — Working Set state changes (VT Supports Multiple Working Sets or Window Masks Visible Simultaneously)

Working Set state change	VT behavior
Active to Inactive and Visible	1. Remove visual indication that the Working Set is the active Working Set. 2. Send the VT Status message to the global address (255) to inform Working Sets of the change.¹ 3. Send the VT On User-Layout Hide/Show message (state: Shown) to the Working Set.
Inactive and Visible to Active	1. Send the VT On User-Layout Hide/Show message (state: Hidden) to the Working Set (state Hidden is a special case here - refer to clause H.20). 2. Display the Working Set's currently active Data/Alarm Mask and display the associated Soft Key Mask. 3. Visually indicate to the operator that the Working Set is the active Working Set. 4. Send the VT Status message to the global address (255) to inform Working Sets of the change.¹
Hidden to Active	1. Display the Working Set's currently active Data/Alarm Mask and display the associated Soft Key Mask. 2. Visually indicate to the operator that the Working Set is the active Working Set. 3. Send the VT Status message to the global address (255) to inform Working Sets of the change.¹
Active to Hidden	1. Hide the Working Set's currently active Data/Alarm Mask and associated Soft Key Mask. 2. Send the VT Status message to the global address (255) to inform Working Sets of the change.¹
Inactive and Visible to Hidden	1. Send the VT On User-Layout Hide/Show message (state: Hidden) to the Working Set.
Hidden to Inactive and Visible	1. Send the VT On User-Layout Hide/Show message (state: Shown) to the Working Set.
¹ When the state of a Working Set changes from inactive to active and this causes the state of another Working Set to go from active to inactive, there shall be only one VT Status message (not two), which shall specify the new active Working Set.	

The Working Set can select different Data Masks or activate Alarm Masks by changing the active mask attribute of the Working Set object with the Change Active Mask command. The Working Set can change the active mask even if the Working Set is inactive. This allows the appropriate mask to be displayed when the Working Set becomes visible. When a Working Set is inactive, its active mask may not be visible, but still remains as the active mask for that Working Set.

4.6.9 Connection management

The VT transmits the VT Status message once per second. The Working Set uses the message to ensure the VT is present and to determine the current status of the VT. If a Working Set does not receive this message for a period of 3 s it is determined to be a shutdown of the VT. When this happens the Working Set shall enter a safe state. The safe state is defined as the state in which all functions dependant on the VT operator interface are put into a known state that will not put the operator or machine at risk. The Working Set may re-establish connection to the VT by restarting the initialization procedure.

Each Working Set Master sends the Working Set Maintenance message once per second. The VT uses this message to ensure that each Working Set is still present. If the VT does not receive this message for a period of 3 s it is determined to be an unexpected shutdown of the Working Set Master (See Figure 6 — Initialization, unexpected shutdown, and expected shutdown) and the following rules apply.

The VT shall not alert the operator -

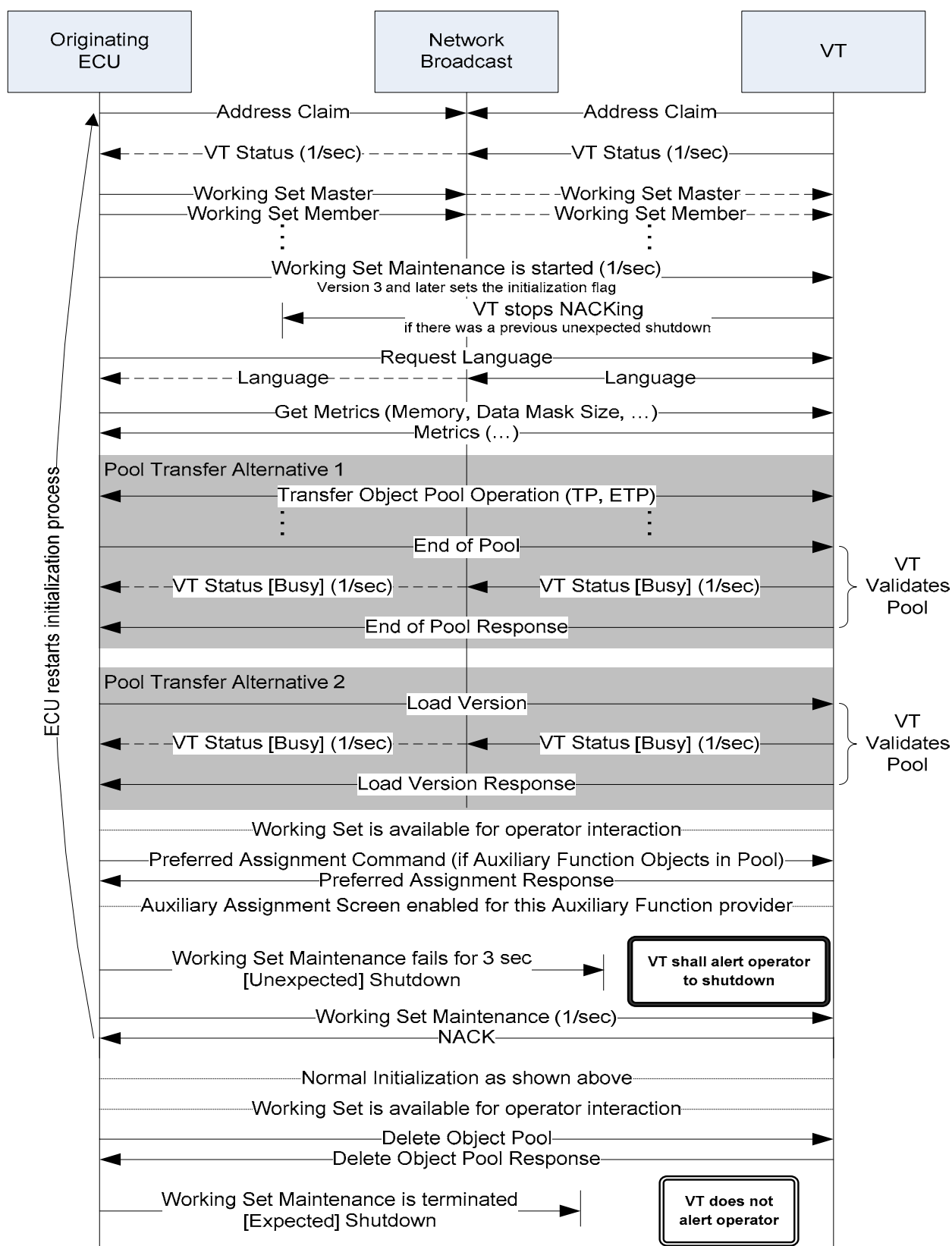
- If the Working Set has commanded the VT to delete the object pool, and the Working Set then stops sending Working Set Maintenance messages. This allows the Working Set to silently remove itself from the VT.
- If the VT can detect the ignition key state and the ignition key is reported as off.
- If there is no pool loaded by the Working Set into the VT volatile memory.

The VT shall alert the operator -

- If the pool has not been commanded to be deleted and the ignition key is not detected as off and the Working Set's object pool is present in the VT. This is detected as an unexpected shutdown of the Working Set and the VT shall alert the operator to this condition after which the VT shall delete the Working Set's object pool from volatile memory to free the memory for other uses. The means to alert the operator is proprietary to the VT. If the Working Set is visible to the operator, the display is cleared and the VT may give control to another connected Working Set and send the VT Status message to the global address. If there is an active alarm for the failed Working Set the VT deselects the Alarm Mask automatically.

When a Working Set's object pool has been deleted and there exists auxiliary assignments mapped to this Working Set, the VT shall remove them.

When the VT receives a Working Set Maintenance message from a Working Set which has unexpectedly shutdown, it shall NACK the message (See ISO 11783-3). The NACK message is sent to the Working Set Master. The Working Set may re-establish connection to the VT by restarting the initialization procedure (See Clause 4.6.6 Initialization).



Key
Solid arrows indicate destination specific messaging, dashed arrows indicate global message.

Figure 6 — Initialization, unexpected shutdown, and expected shutdown

4.6.10 Updating the operator interface

4.6.10.1 General

CAN has finite bandwidth available in support of all services (e.g. ISO 11783 part 3 through part 14). Further, the VT has finite bandwidth that is shared by all Working Sets using its services. In order to best manage the system bandwidth, it is recommended that:

- Active Working Sets, or those that are inactive but visible, should issue commands to the VT only when the data has changed in a way which is visible to the operator (e.g. only update objects actively displayed).
- Inactive Working Sets, which have no active Data Mask and Soft Key Mask displayed should reduce the frequency of, or eliminate, updates to the VT.

4.6.10.2 Changing attributes and values

Attributes of objects can be changed during operation by Working Set Masters and Working Set Members using the defined change attribute messages. Certain attributes in each object are assigned an attribute ID. The Change Attribute command allows any attribute with an AID to be changed if not designated as a read-only attribute. In addition, attributes are sometimes grouped together into a single “change” command for efficiency purposes. (e.g. Change Font Attributes command F.28)

Even when the associated Data Mask is not visible, the Working Set may continue to change the attributes (including value) so that when the mask is made active and visible, the necessary output data is current and ready to display.

4.6.10.3 Changing, adding and deleting objects

Working Sets can replace objects at run-time; however the replaced objects shall be of the same type. New objects can be added by initiating a transport protocol session to send one or more objects to the VT. When the VT receives an object with an existing Object ID, the existing object is replaced (the VT can determine the owner from the source address of the message). Resizing objects is permitted but can cause the VT to run out of memory. (See Annex C)

The entire object pool can be deleted from the volatile memory in the VT by the Working Set sending a Delete Object Pool command.

4.6.11 Special objects

4.6.11.1 Container objects

A Container object is a special object used to

- logically group objects in order to identify and reuse the container, or
- hide and show objects.

Figure 7 — Container reuse shows an example of container reuse. Mask 1 and Mask 2 both need the information displayed in Container 1. The Working Set first creates the container, and then inserts the container into Mask 1 and Mask 2 using the Object ID of the container.

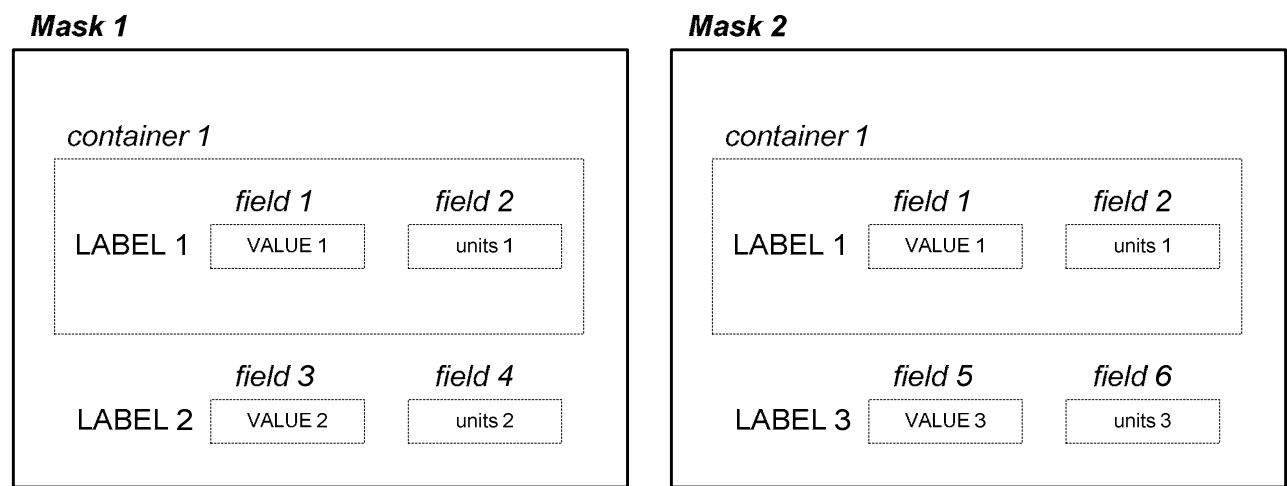


Figure 7 — Container reuse

Figure 8 — Container used to hide objects — Example a) shows an example of using a container to hide a group of objects. In the example, Text 1 and Text 2 should be visible at all times. Text 3 and Text 4 should be visible only if a particular feature of an implement is available. Therefore, the Working Set creates a container containing Text 3 and Text 4 to be inserted in the mask. At run-time, the Working Set determines whether or not the particular feature is available. If not, the Working Set hides the container (See b in Figure 8 — Container used to hide objects — Example).

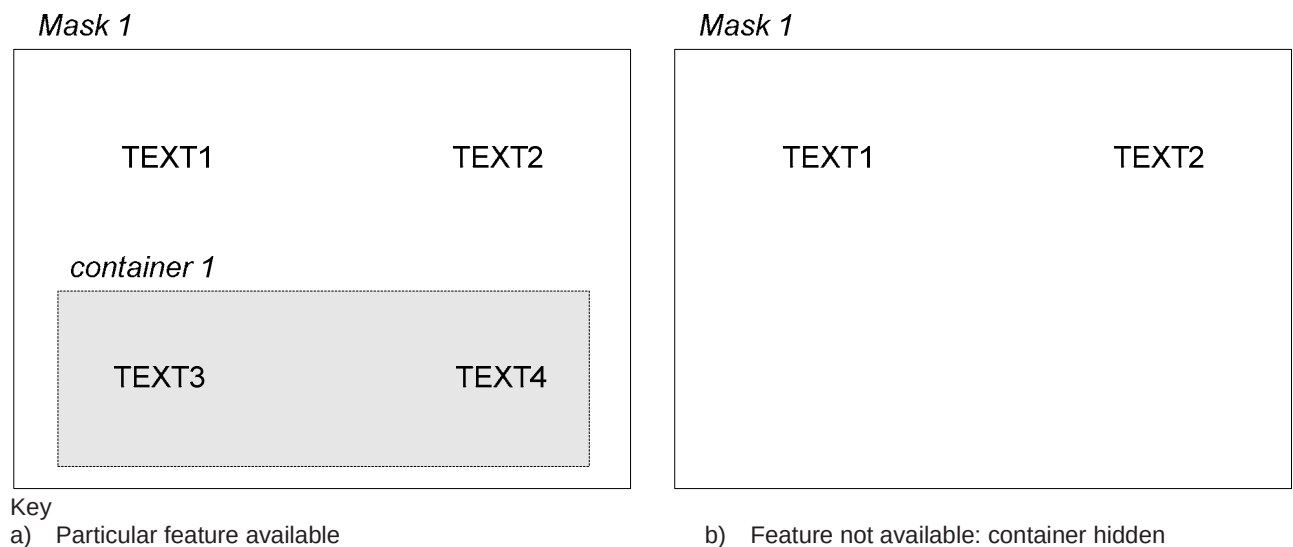


Figure 8 — Container used to hide objects — Example

4.6.11.2 Attribute objects

There are five types of attribute objects: font, line, fill, input, and extended input. Attribute objects are referenced by other objects. This allows one set of attributes to be shared with many objects to create and maintain a common look across those objects. All objects using a given attribute object are updated when the attribute object is changed.

4.6.11.3 Variable objects

Variable objects can be used to share data between two or more other objects. Changing the value in only one object can reduce bus traffic. For example, it could be desirable to draw a Meter object and also to show its current value as a numeric under the meter. In this case, a Number Variable object could be referenced by

both the Meter object and the output field object. By doing this, a change in the value of the variable will be reflected in both the meter and output field at the same time and the change will require only a single Change Numeric Value command.

4.6.11.4 Macros

Macro objects are used to improve the performance of the operator interface. Macro objects have the following properties:

- a) Macros can only contain the commands listed in Annex F.
- b) If a Macro triggers an event that causes another Macro, the current Macro is completed first before another is started.
- c) Macros are executed in the order they were triggered.
- d) Macros triggered by the execution of a command shall be completed before the next bus command is started.
- e) Macro Object IDs shall be in the range 0 to 255 for VT version 4 and prior. Macro Object IDs may be in the range of 0 to 65534 for VT version 5 and later.
- f) Macros do not trigger response messages. When executing the command messages in a macro, the VT shall not send response messages on the CAN bus for the messages contained in the macro. Therefore macros reduce CAN traffic. For example, a Macro that executes a Change Active Mask command will trigger a VT Status message, but will not trigger a Change Active Mask response.

CAUTION: Objects that are altered by both a Working Set and a Macro may be susceptible to a race condition and should be evaluated for predictable behavior.

EXAMPLE 1 An

Input String object has a Macro to clear the

Input String object - On Input Field Deselection. The "Enter" means on an

Input String object of length greater than 3 characters causes the VT to send the data using Transport Protocol (TP) to the Working Set (WS). The operator quickly navigates away from the

Input String object. "On Input Field Deselection" causes the VT to notify the WS with a VT Select Input Object message (Deselect). An On Input Field Deselection Macro alters the

Input String object value. The WS receives the TP data asynchronous to the VT Select Input Object message (Deselect). The outcome is based on the asynchronous processes in the VT, the message processing methods in the VT, including TP processing and potential latencies, and the WS implementation.

EXAMPLE 2 An On Key Press Macro associated with a Button object changes the active Data Mask. An On Show Macro attached to the new Data Mask changes the numeric value in an Input List object on this new Data Mask to initialize it to a known setting. The operator presses the Button, which causes a Button Activation message to be sent to the WS. The WS sends a Change Numeric Value command to the Input List object. The Input List object may end up with either value, due to the asynchronous processes.

Circular references are not permitted since they create infinite Macro loops inside the VT and could render the VT inoperable for all Working Sets.

EXAMPLE When a Macro triggers an event that references the same Macro, a circular reference is created.

4.6.11.5 Object pointer

The Object Pointer object allows run-time modification of included objects. By changing the value of the Object Pointer object, a different object can be drawn at the same location. The type of object that the Object Pointer is allowed to reference is limited and depends on the parent object. Refer to valid parent objects of the Object Pointer for a list of objects that can be referenced. An Object Pointer can always point to another Object Pointer. An Object Pointer can point to the NULL Object ID and in this case nothing shall be drawn.

When an Object Pointer is modified at run-time, resulting in an invalid object reference, the VT is not required to detect this object pool error immediately but may delay error detection until the Data Mask object or Alarm Mask object which contains the Object Pointer is activated. At activation of a data or Alarm Mask containing the invalid object reference the VT sends the F.35 "Change Active Mask response" or H.14 "VT Change Active Mask message" to inform the Working Set and may delete the object pool.

The VT may also detect the error immediately upon Object Pointer value modification, and in this case shall send an F.23 Change Numeric Value response with "invalid value" indicated in the error codes. The VT may also delete the object pool.

4.6.11.6 External Object Pointer

The External Object Pointer object allows a WS to display objects from an object pool of another WS.

To ensure that the external references are valid even after software updates of the participating WS, the referenced WS and the referencing WS shall exchange information about the referenced objects before enabling the external references. The information exchange shall be repeated every time either the referencing WS or the referenced WS is restarted. As a minimum the object identifiers shall be transferred from the referenced WS to the referencing WS.

Note: Some devices (e.g. the SCC/SCM – ISO11783-14) have a standardized method for exchanging object information. If a standardized method does not exist, then a proprietary method shall be agreed between the manufacturers of the referencing and the referenced WS.

To protect against unsolicited references the referenced WS shall use the External Object Definition object to list the objects which are allowed to be referenced by another WS. The External Object Definition object is assigned to one and only one referencing WS. If a referenced WS will allow multiple WS to reference objects it shall have multiple External Object Definition objects.

The referencing WS shall use the External Reference NAME object to identify the WS it plans to reference.

The attribute values of the External Object Definition object, the External Object Pointer object and the External Reference NAME object shall all be valid before the external object can be displayed. Some of the information in the objects depends on the exchange of object information and therefore the objects shall be in the reset state until the object information is complete.

Reset state is defined by:

- External Object Pointer object : External Object Id attribute is the NULL object id
- External Object Definition object : Enable bit in the Options attribute is cleared
- External Reference NAME object : Enable bit in the Options attribute is cleared

When the object pool is loaded from non-volatile storage in the VT there is no guarantee that the attribute values are valid, and therefore the VT shall set all occurrences of the External Object Pointer, the External Object Definition and the External Reference NAME to the reset state.

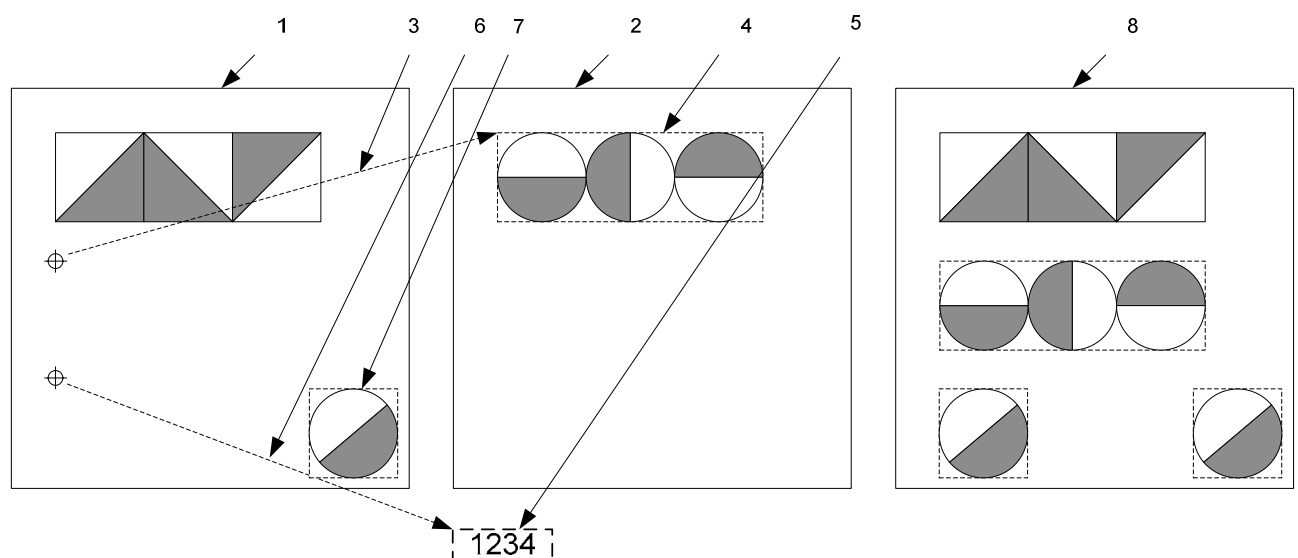
When the object pool is uploaded from the WS master, the VT shall not set the objects to the reset state. If the information exchange between the referenced and the referencing WS is not completed before the object pool is uploaded, the WS master shall set the objects to the reset state before uploading the pool.

Before displaying an external object the VT shall check the validity of the external reference. The external reference is considered to be valid when all of the following are fulfilled:

- The External Reference NAME ID attribute of the External Object Pointer object identifies an enabled External Reference NAME object.
- The referenced WS has an object pool in volatile memory on the VT (referenced object pool).
- The referenced object pool contains an enabled External Object Definition object where the NAME attributes identifies the referencing WS.
- External Object Id is listed in the object list in the above mentioned External Object Definition object.

If the referenced object is NULL or not valid the VT shall draw the object identified in the Default Object ID attribute of the External Object Pointer object.

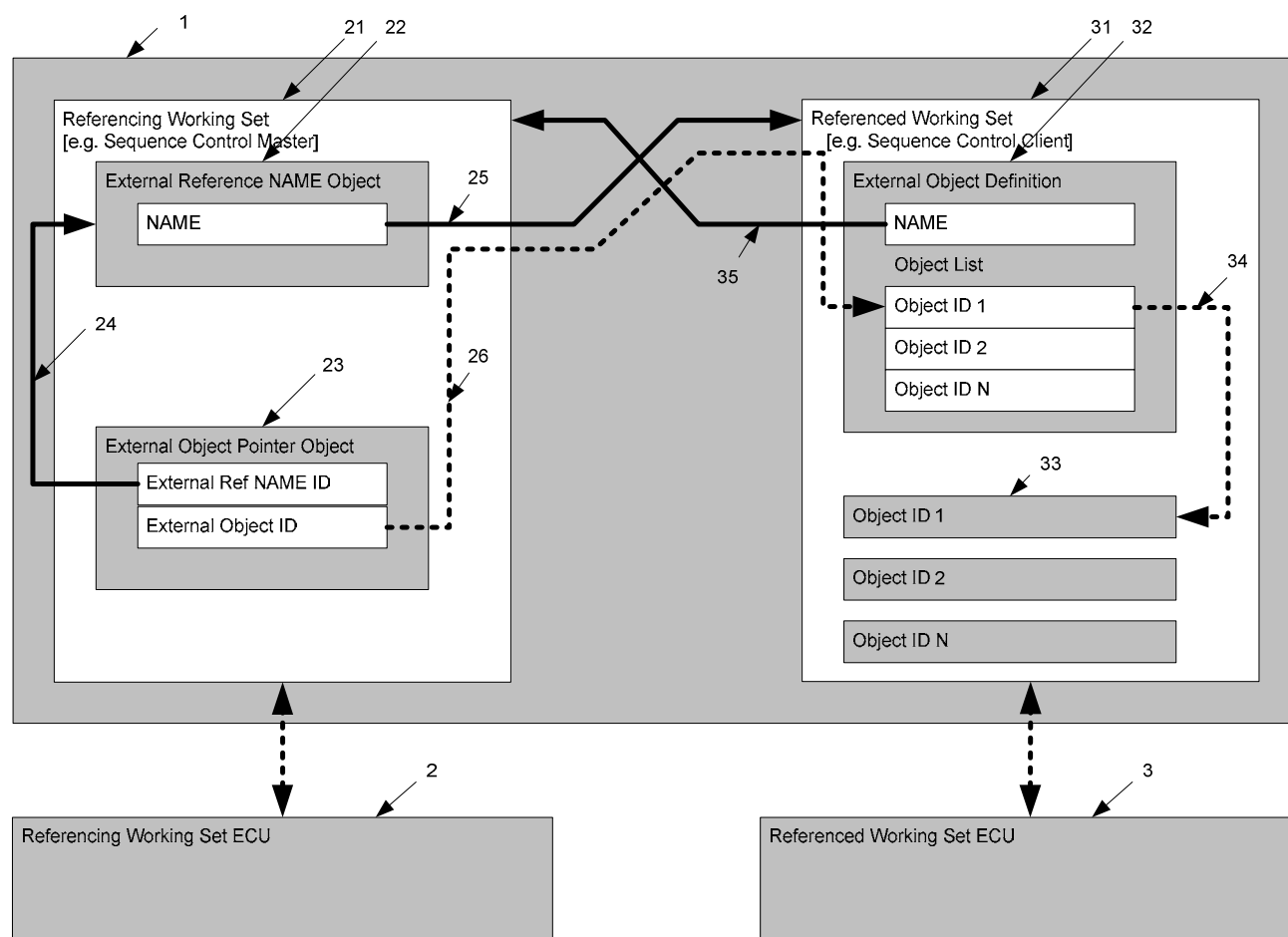
Note There can be multiple instances of the External Object Definition object in an object pool. If multiple External Object Definition objects are assigned to the same NAME, then an external reference shall be considered valid if made valid by one or more External Object Definition objects.



Key

- 1 VT presentation from Working Set 1
- 2 VT presentation from Working Set 2
- 3 External Object Pointer with reference to object in Working Set 2 Object Pool
- 4 Referenced object in Working Set 2 (container that contains several objects), all of which are valid
- 5 Number Variable on Working Set 2 object pool
- 6 External Object Pointer with reference to object in Working Set 2 Object Pool
- 7 Default Object to be shown when External Objects are invalid
- 8 VT presentation from Working Set 1 after references are established, enabled and resolved

Figure 9 — External Object References — VT Example



Key

- | | |
|---|---|
| 1 VT Object Pool volatile memory | 3 Referenced Working Set ECU (ECU 3) |
| 2 Referencing Working Set ECU (ECU 2) | 31 Referenced Working Set in VT memory |
| 21 Referencing Working Set in VT memory | 32 External Object Definition object |
| 22 External Reference NAME Object, including NAME attribute that references ECU 2s | 33 Object Pool Object that can be referenced from an external Working Set |
| 23 External Object Pointer Object | 34 Reference to the Object Pool object that can be referenced. |
| 24 Reference to the External Reference NAME object | 35 Virtual reference established that allows ECU 2 to reference ECU 3 objects |
| 25 Virtual reference established by External Object Pointer object that informs VT to the Working Set containing the referenced objects | |
| 26 Virtual reference established to the External Object of interest. | |

Figure 10 — External Object References — Relationship Example

Example 1 Two External Object Definition objects have the same value in the NAME attributes. Both objects are enabled, only one of the objects includes 1234 in the Object ID list. Object Id 1234 can be referenced.

Example 2 Two External Object Definition objects have the same value in the NAME attributes. Both objects include 1234 in the Object ID list, only one of the objects is enabled. Object Id 1234 can be referenced.

4.6.12 Relative X/Y positions

The X, Y position attribute determines where an object is drawn on the display. This position is always relative to the upper left corner of the parent object. The X,Y position is always found in the parent object. Figure 11 — Relative and absolute location of objects shows an example Data Mask with the relative locations of several objects.

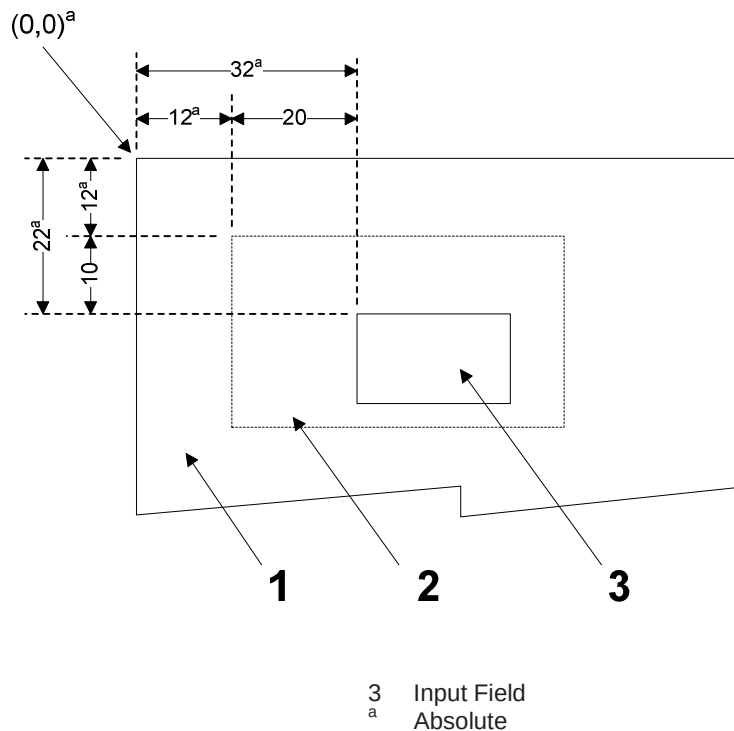


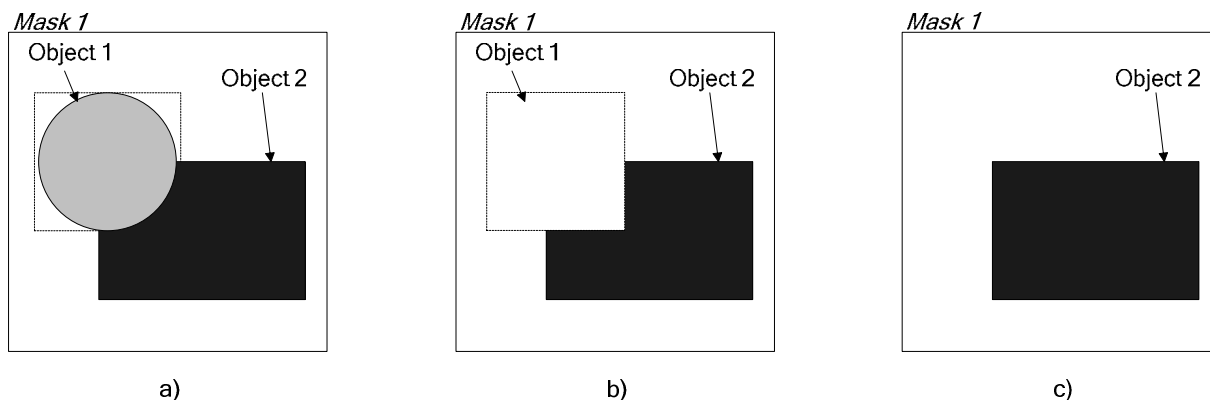
Figure 11 — Relative and absolute location of objects

4.6.13 Overlaid objects

A mask can be built such that two objects will occupy the same or overlapping space on the display. This can require extra processing in the VT, but could be necessary in some cases and shall be supported by the VT. For this reason, the hierarchy or layering of objects shall be understood. Some objects have a list of contained objects. Objects listed first are considered to be lower in the hierarchy than objects listed later. In this case, the objects shall be drawn such that objects listed later appear to overlay objects listed earlier, so that the entire image of the object listed last is visible, while only those areas of the object listed earlier that do not coincide with areas of the object listed last will be visible. When an object is changed, all objects that overlay it and which have been corrupted shall be redrawn (this is called a refresh event). All objects defined by this part of ISO 11783 have a rectangular size either defined or implied to simplify the VT's task of finding overlaid objects.

When an object is changed or hidden, the VT shall update the display accordingly.

EXAMPLE Referring to Figure 12 — Object changed or hidden — Display update, where (a) shows the initial presentation before any command is received. The Working Set then commands object 1 to be hidden. As an intermediate step shown in (b), the VT deletes object 1 and all child objects by filling the object's area with the background colour of the parent mask. The VT then redraws the object along with all child objects. The VT then refreshes any other object (e.g. object 2) that could have been visually altered in the deletion process as shown in (c). The VT may implement this refresh in a manner where the intermediate display is never seen by the operator.



- Key
- a) Initial presentation
 - b) Intermediate operation
 - c) Final presentation

Figure 12 — Object changed or hidden — Display update

4.6.14 Alarm handling

Alarms allow a Working Set to display alarm information at any time. If several Working Sets have an Alarm Mask activated, the VT shall display the masks in order of priority. Priority is determined, first, by the priority attribute defined in the Alarm Mask object and, second, by chronological order of activation. The highest priority alarm is always displayed until the owner Working Set changes the active mask. When more than one Working Set has asserted an Alarm Mask with the same priority attribute, the first of these, as processed by the VT, shall become the active mask. When the active Working Set changes to a lower priority alarm or Data Mask, the next in turn highest priority Alarm Mask is processed.

Operator input disturbed by the activation of an Alarm Mask may be left intact for resumption once all Alarm Masks have been acknowledged. A Soft Key Mask is associated with the Alarm Mask via an attribute in the Alarm Mask object. Whenever the alarm is displayed, the associated Soft Key Mask is also displayed. The following describes the protocol requirements between the VT and the Working Set raising the alarm.

- a) The Working Set Master activates an Alarm Mask by using the Change Active Mask command on the Working Set object. Only one Alarm Mask can be active per Working Set.
- b) The VT responds with a Change Active Mask response.
- c) Based on priorities, at some point, the VT displays the Alarm Mask and the Soft Key Mask associated with the Alarm Mask. When a mask change occurs that causes an Alarm Mask to appear or reappear, the acoustic signal associated with the Alarm Mask shall be activated. The VT shall terminate any acoustic signal from a lower priority Alarm Mask that can be in process. The VT shall terminate any acoustic signal from a control audio command that can be in process (version 4 and later VTs shall send the VT Control Audio Signal Termination message to indicate the termination). If the Alarm Mask acoustic signal completes, or if it is set to none for silent signal, then control audio commands from ECUs shall be accepted as defined in the Control Audio Signal command. (See Table 2 — Working Set state changes (VT Supports only Active Mask) and Table 3 — Working Set state changes (VT Supports Multiple Working Sets or Window Masks Visible Simultaneously))
- d) The VT notifies the Working Set with a VT Status message sent to the global address.
- e) At some point the operator may acknowledge the alarm with the proprietary ACK means on the VT. The VT sends a Soft Key Activation message to the Working Set with key code set to zero (0) based on the