

BS ISO 11783-6:2010



BSI Standards Publication

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

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

raising standards worldwide™

Full document details: BS ISO 11783-6:2010



National foreword

This British Standard is the UK implementation of ISO 11783-6:2010. It supersedes BS ISO 11783-6:2004 which is withdrawn.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.

© BSI 2010

ISBN 978 0 580 62543 5

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 30 November 2010.

Amendments issued since publication

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

INTERNATIONAL STANDARD

BS ISO 11783-6:2010
ISO
11783-6

Second edition
2010-10-15

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

Part 6: Virtual terminal

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

Partie 6: Terminal virtuel



Reference number
ISO 11783-6:2010(E)
Documents
Document Sharing Hub
© ISO 2010

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



COPYRIGHT PROTECTED DOCUMENT

© ISO 2010

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing 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	v
Introduction.....	vi
1 Scope	1
2 Normative references	1
3 Terms, definitions and abbreviated terms	1
4 Technical requirements	4
4.1 Overview.....	4
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 Data Mask	9
4.5.2 Soft Key Mask area and Soft Key designators	9
4.6 Behaviour	12
4.6.1 Object pools	12
4.6.2 Working Sets.....	13
4.6.3 Language, formats and measurement units selection	14
4.6.4 Initialization.....	14
4.6.5 Working Set object and active masks	16
4.6.6 Connection management.....	19
4.6.7 Updating the operator interface	21
4.6.8 Special objects	21
4.6.9 Relative X,Y positions	23
4.6.10 Overlaid objects.....	24
4.6.11 Alarm handling	25
4.6.12 Clipping	26
4.6.13 Scaling	27
4.6.14 Operator input.....	27
4.6.15 Soft key and button activation	30
4.6.16 Font rendering	31
4.6.17 Filling output shape objects.....	38
4.6.18 Events	40
4.6.19 Touch screens and pointing devices	40
4.6.20 Proprietary Means	41
4.6.21 VT Number	41
4.7 Displaying data from Multiple Working Sets on one Mask (Version 4 VT and later).....	41
4.7.1 General	41
4.7.2 User-Layout Data Mask.....	42
4.7.3 Window Mask object	42
4.7.4 Window Mask content.....	43
4.7.5 Window Cell Size and Borders.....	44
4.7.6 Window Mask scaling	44
4.7.7 Using Window Masks outside User-Layout Data Masks.....	45
4.7.8 User-Layout Soft Key Mask.....	45
4.7.9 Key Group Objects	45
4.7.10 Key Cell Size and Borders	46
4.7.11 Key Group Scaling	47
4.7.12 Using Key Group Objects outside User-Layout Soft Key Masks	47
4.7.13 Operator Inputs.....	47
4.7.14 Refreshing on-screen data	47

4.7.15 User-Layout Data Mask look and feel	48
4.7.16 Uploading New Window Mask and Key Group objects	49
Annex A (normative) Object, event, colour and command codes	50
Annex B (normative) Object definitions	63
Annex C (normative) Object transport protocol	149
Annex D (normative) Technical data messages	154
Annex E (normative) Non-volatile memory operations commands	162
Annex F (normative) Command and Macro messages	166
Annex G (normative) Status Messages	198
Annex H (normative) Activation messages	200
Annex I (normative) Other messages	210
Annex J (normative) Auxiliary control	211
Annex K (normative) Extended transport protocol	245
Annex L (normative) Character sets	250
Bibliography	258

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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.

ISO 11783-6 was prepared by Technical Committee ISO/TC 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics*.

This second edition cancels and replaces the first edition (ISO 11783-6:2004), which has been technically revised. It also incorporates the Technical Corrigendum ISO 11783-6:2004/Cor.1:2005. It introduces requirements for Version 4 VTs and Working Sets, while retaining the requirements of the first edition for Version 3 VTs and Working Sets. The specific requirements of Annex G, as well as Annex J and the revised Table L.2, are also applicable to Version 3.

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 CAN 2.0 B^[1] protocol. SAE J 1939^[2] 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^[3]. 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 part of ISO 11783 that refer explicitly to a software term for an object or a command have the first letter of each object or command word capitalized (e.g. Linear Bar Graph object, Change Numeric Value command). This aids in the recognition of each of these terms as being a specific item having a specific definition in the 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 and mounted, semi-mounted, towed or self-propelled implements. Its purpose is to standardize the method and format of transfer of data between sensors, actuators, control elements and information storage and display units, whether mounted on, or part of, the tractor or implement. This part of ISO 11783 describes a universal virtual terminal (VT) that can be used by both tractors and implements.

It is applicable to both Version 3 and Version 4 VTs and Working Sets.

2 Normative references

The following referenced documents are indispensable for the application of this document. 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, definitions and abbreviated terms

For the purposes of this document, the terms, definitions and abbreviated terms given in ISO 11783-1 and the following terms and definitions 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 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 references a specific object within an object pool

3.4

attribute ID

AID

numeric value which references a specific object's attribute

NOTE 1 AID values range from 0 to FE_{16} (254_{10}), with 255 as the `NULL_AID`.

NOTE 2 This field appears in the attribute and record format tables. AIDs that are explicitly defined with square brackets, [], are read-only attributes using the Get Attribute Value message. AIDs that are explicitly defined without square brackets are writable with the Change Attribute command.

3.5

char

single character where the size is 1 byte

NOTE Commonly used for ISO 8859 characters (e.g. 41_{16} in ISO/IEC 8859-1 represents "A"), see Annex L.

3.6

character

single text grapheme or symbol, as in an alphabet

NOTE Size is variable, based on the encoding scheme (see char and WideChar).

3.7

code plane

group of 65 536 possible character codes

NOTE Unicode/ISO 10646 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 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 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 UTF-16 encoding of characters in code plane 1 to 16 (see 4.6.16.6).

NOTE 2 The UTF-16 character encoding scheme is defined by ISO 10646.

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".

NOTE 1 See Annex L.

NOTE 2 Two WideChars can be combined to indicate character codes exceeding 16 bits (see 4.6.16.6).

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 String length is variable.

3.14

VT number

identification number that is derived from the function instance of the VT

NOTE VTs can then be referenced as VT Number 1, VT Number 2, etc.

3.15

User-Layout Data Mask

special Data Mask controlled by the VT but laid out by the operator

NOTE See 4.1 and 4.7.

3.16

window cell

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

NOTE See 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 See 4.7.

3.18

User-Layout Soft Key Mask

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

NOTE See 4.7.

3.19

Key Cell

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

NOTE See 4.7.

3.20

Key Group Object

area of one or more Key Cells and containing a grouping of one or more Key Objects

NOTE See 4.7.

3.21

non-VT screen

VT-capable display on which no Data Masks are visible

NOTE See 4.7.

3.22

non-VT area

area outside the normal Data Mask and Soft Key Mask visible at the same time as a Data Mask and Soft Key Mask are displayed

NOTE See 4.7.

3.23

range

value

signifying that each object has an attributes and record format table

NOTE Each parameter has a range or value for the specific attribute. Where there are ranges and values that are explicitly defined with square brackets, [], they are applicable to VT Version 3 and prior, whereas the adjacent unbracketed range or value applies to VT Version 4 or later.

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 represented either 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 and control keys of the VT to allow the operator to send signals back to the Working Set.

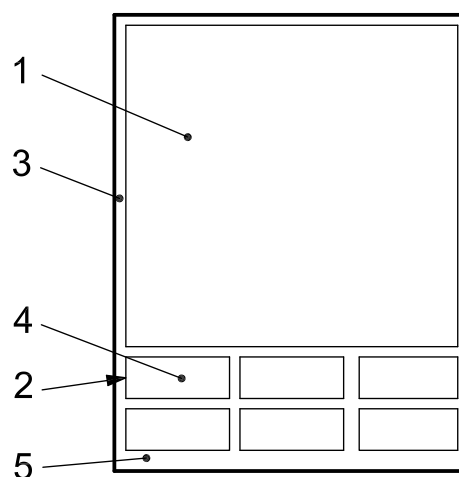
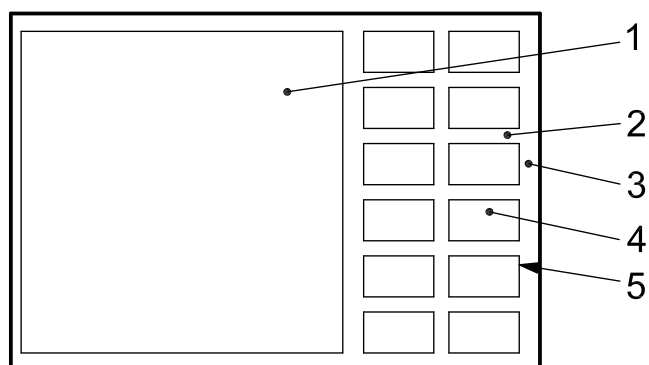
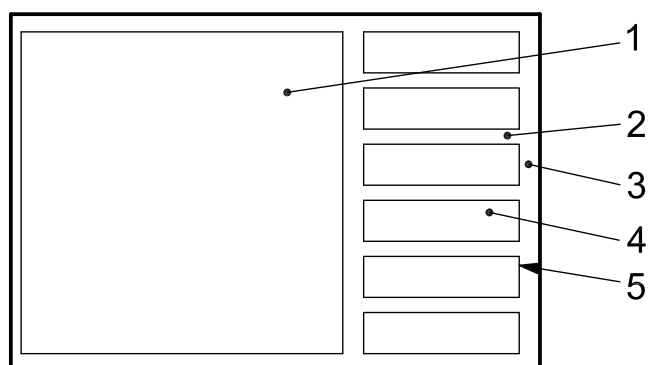
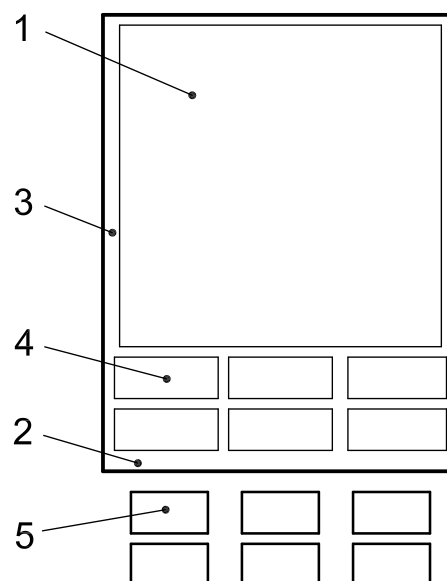
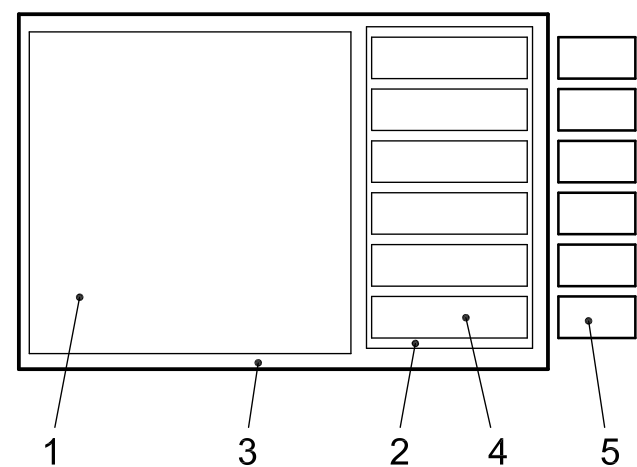
This part of ISO 11783 describes the VT with the detail and clarity required for VTs built by different manufacturers to be interchangeable with any implement Working Set that uses the VT services. The interface protocol of this part of ISO 11783 also reduces the runtime ISO 11783 communication bus traffic 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 manufacturers' 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 shows examples of some possible VT designs and orientations.



Key

- 1 Data Mask Area
- 2 Soft Key Mask area
- 3 physical screen
- 4 Soft Key designator
- 5 physical Soft Key

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.

a) Soft

This 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

This 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

This 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

This 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 ESC means either may be a permanent key or may only be available during data entry (see Table 4). 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 input

This is a means of data input for an Input Number object and Input List object where the object has focus and 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 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 data input, 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 data input shall meet the operator controls requirement specified in ISO 15077.

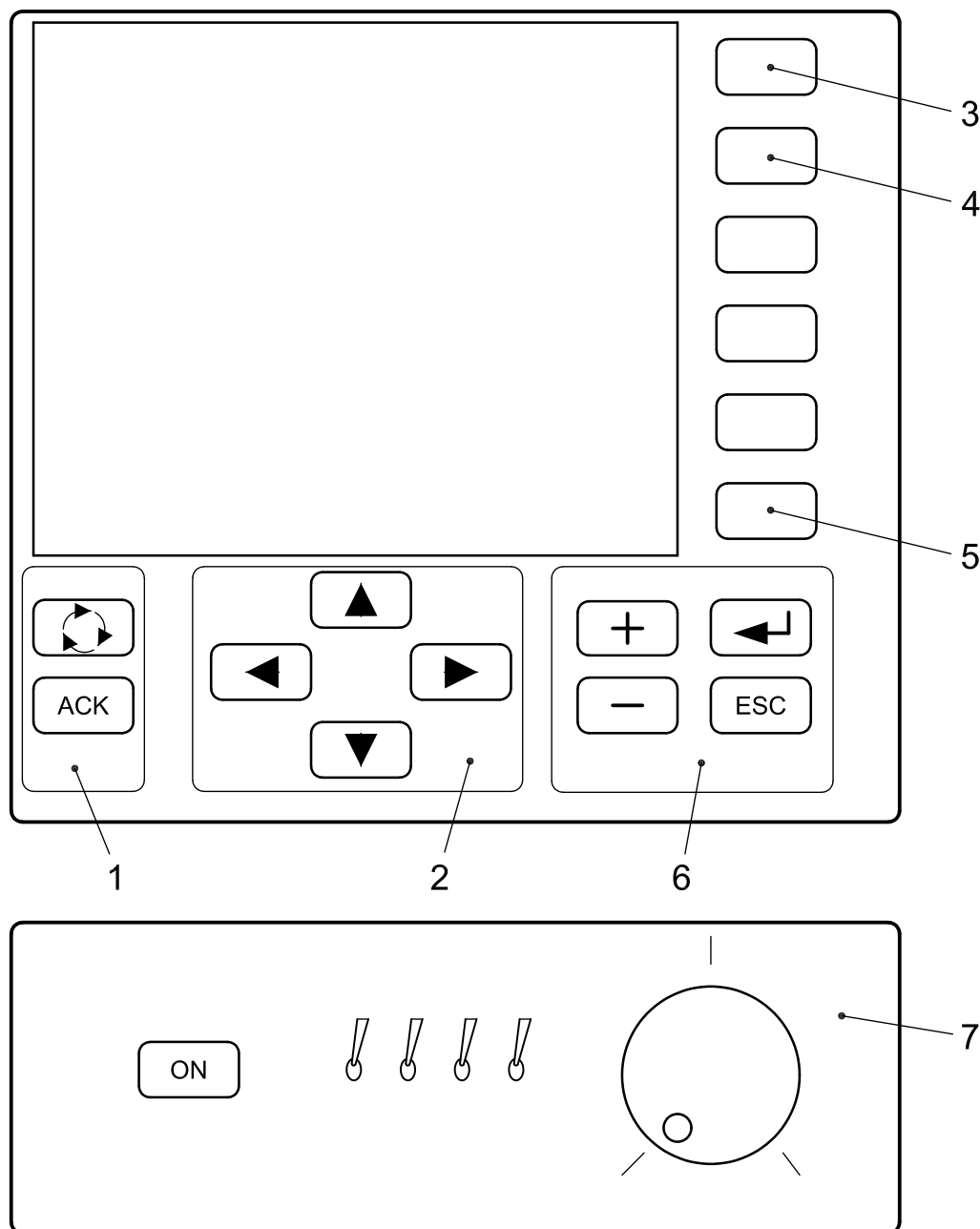
d) Control

This is a means of selecting between Working Sets whenever a Data Mask is visible, and a means of 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

This 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 | 5 Soft Key 6 |
| 2 navigation | 6 data input |
| 3 Soft Key 1 | 7 auxiliary input |
| 4 Soft Key 2 | |

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 variable frequency and audio level.

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 corner of the parent object.

4.5 Display areas

The following defines standard Data Mask and Soft Key Mask areas of the display. Alternative usage of these areas supports displaying data from multiple working sets (see 4.7).

4.5.1 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). 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 $\times$ 200 pixels. This requirement does not limit the physical resolution or size of the display, only the usable 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 $\times$ 200,
- 240 $\times$ 240,
- 320 $\times$ 320, and
- 480 $\times$ 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.

4.5.2 Soft Key Mask area and Soft Key designators

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). Each Soft Key shall have a reserved display area, called a Soft Key designator, for displaying a label (see Figure 1). The minimum size of the designator field is 60 pixels wide $\times$ 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). It is recommended that this visible separation be drawn outside the Soft Key designator area.

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 $\leq$ 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 six Physical Soft Keys.

- c) VT Version 3 and prior shall support a maximum of 64 virtual Soft Keys per Soft Key Mask (see 4.5.2.2).
- d) VT Version 4 and later shall support exactly 64 virtual Soft Keys per Soft Key Mask (see 4.5.2.2).
- 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 Soft Keys using the six physical keys.

4.5.2.1 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.

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.

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.

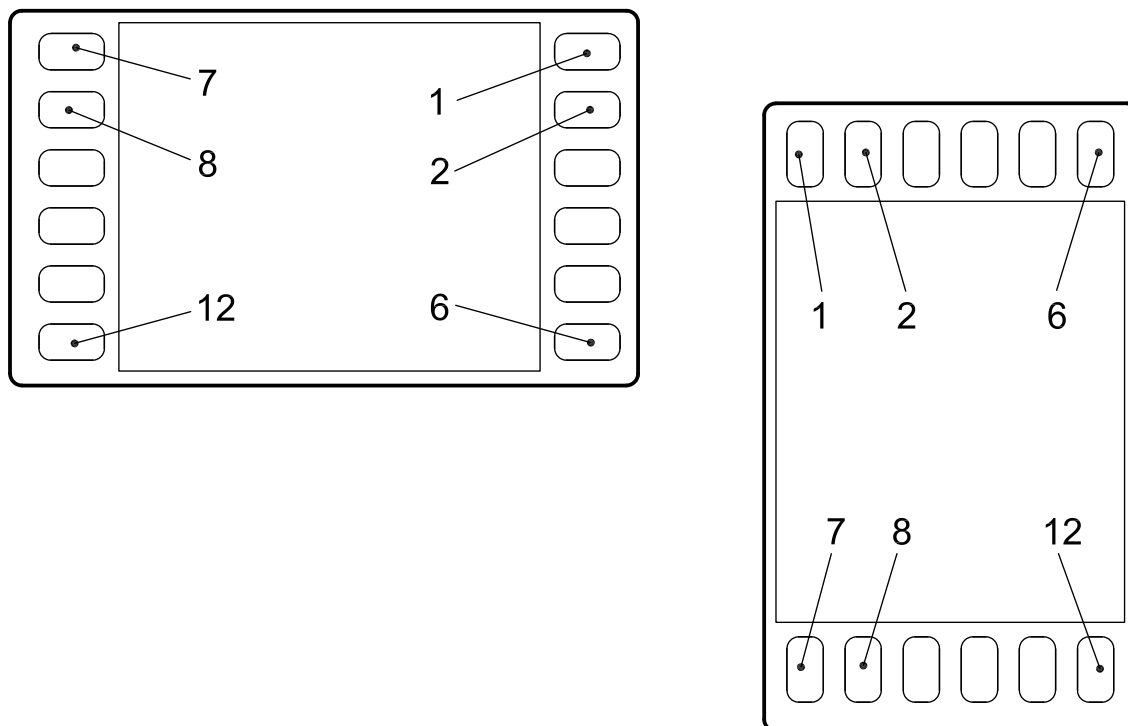


Figure 3 — Physical Soft Key orientation examples showing key locations

4.5.2.2 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 Set's Soft Keys.

4.5.2.3 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 another means of navigation that does not use the physical Soft Keys, this value shall be zero.

4.5.2.4 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 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.

The VT shall not provide navigation to any trailing Soft Keys while those items are Pointers to NULL Object ID.

As shown in Figure 4, a VT is designed with six physical Soft Keys, 64 virtual Soft Keys, and one [a] in Figure 4] navigation Soft Key. The Working Set provides 18 Soft Keys to the VT; however, there are three which are Pointers to 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 five Soft Keys [a] in Figure 4], 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 4] shows a similar example with two navigation Soft Keys. Another example [c] in Figure 4] 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].

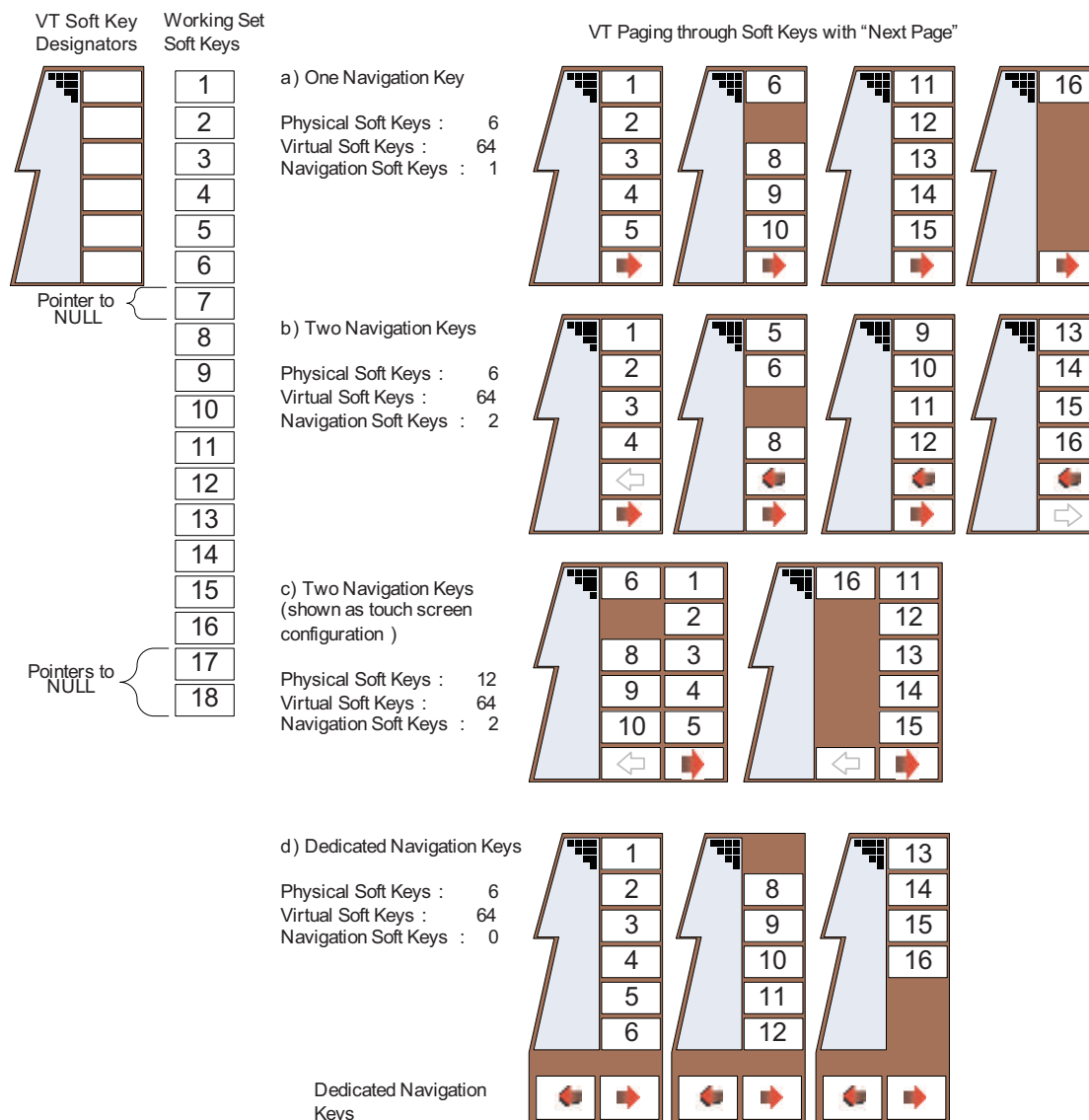


Figure 4 — VT virtual Soft Key paging

4.6 Behaviour

4.6.1 Object pools

4.6.1.1 General

The operator interface definition for a device or 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 the following subclauses. 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 transport protocol described in ISO 11783-3, and the extended transport protocol specified in Annex K. The procedure is described in more detail in Annex C. The VT is intended to be capable of storing the object pools in a modifiable memory area. The size and number of object pools are limited only by the VT's available memory and software design, but only one object pool per Working Set exists. All objects shall be fully described before they are made active in a mask on the display.

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.

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 ISO 11783-5 NAME 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 runtime.

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 is never considered to be a Working Set and therefore 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 ISO 11783-1 in respect of responses being directed only to the Working Set Master. See Table 1.

Table 1 — VT Response message behaviour

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 1 to 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.

In configuration 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 Widechar 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.

4.6.3 Language, formats and measurement units selection

The following is applicable to VTs.

- The VT shall send the standard language, format and measurement units messages defined in ISO 11783-7, hereafter referred to as “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.
- The VT 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 — and to the selected units — e.g. by changing offsets and scales. If the Working Set does not support the specified language, formats or units, it shall use a proprietary method to select an appropriate setting, and can therefore choose default units as specified by the Working Set designers.
- The VT shall store the standard setups in non-volatile storage and restore the values during initialization.
- The VT shall respond to ISO 11783-7 “Language Command” requests sent to the global address (GA).
- The VT shall respond to ISO 11783-7 “Language Command” requests directed to the VT.

The following is applicable to Working Sets.

- Working sets 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).
- Working Sets shall use their default language if the selected language is not supported.

4.6.4 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.

a) 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 at least 3 s 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.

b) 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 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 ISO 11783-7.

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

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.

If the VT had previously detected a shutdown and was transmitting the NACK in response to the Working Set Maintenance message (see 4.6.6), 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.
 - 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 a 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 transport protocol (see ISO 11783-3), extended transport protocol (see Annex K) and the messages given in Annex C.

c) 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, and
- v) 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.

4.6.5 Working Set object and active masks

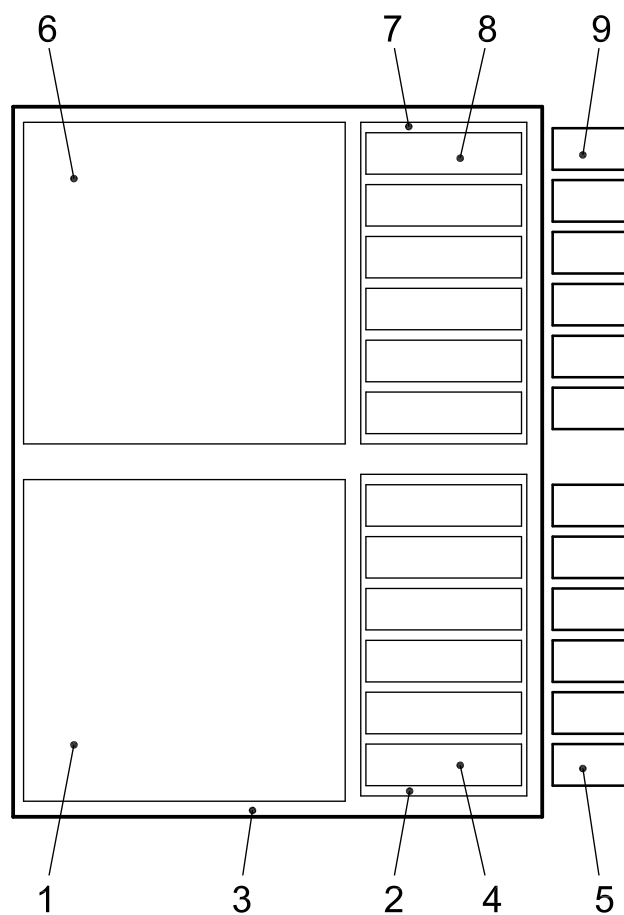
In the initial object pool definition, each Working Set Master shall provide one, and only one, Working Set object in order to define a descriptor, active mask and supported languages for the Working Set. The descriptor may be graphical, text or both, but shall fit inside the area defined by the VT for a Soft Key designator. 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 could 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.

For 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). 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 4.6.7).

For Working Set state changes, see Table 2.



Key

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

Figure 5 — VT displaying active and inactive Working Sets simultaneously — Example

Table 2 — Working Set state changes

VT behaviour		
Working set state from/to	VT displays only active Working Set	VT displays active and inactive Working Sets simultaneously
Active to Inactive	Hide the Working Set Hide the associated Soft Key Mask Send the VT Status message to the global address (255) to inform Working Sets of the change^a	Hide the Working Set, if it is not visible in its inactive state Hide the associated Soft Key Mask, if it is not visible in its inactive state Remove visual indication that the Working Set is the active Working Set Send the VT Status message to the global address (255) to inform Working Sets of the change^a Send the On User-Layout Hide/Show message to the Working Set when it is now inactive but visible
Inactive to Active	Display the Working Set's currently active Data Mask or Alarm Mask Display the associated Soft Key Mask Visually indicate to the operator that the Working Set is the active Working Set Send the VT Status message to the global address (255) to inform Working Sets of the change^a	Send the On User-Layout Hide/Show message when the Working Set was inactive but visible Display the Working Set's currently active Data Mask or Alarm Mask Display the associated Soft Key Mask Visually indicate to the operator that the Working Set is the active Working Set Send the VT Status message to the global address (255) to inform Working Sets of the change^a
Hidden to Active	Display the Working Set's currently active Data Mask or Alarm Mask Display the associated Soft Key Mask Visually indicate to the operator that the Working Set is the active Working Set Send the VT Status message to the global address (255) to inform Working Sets of the change^a	Send the VT Status message to the global address (255) to inform Working Sets of the change^a
Active to Hidden	Hide the Working Set Hide the associated Soft Key Mask Send the VT Status message to the global address (255) to inform Working Sets of the change^a	Send the VT Status message to the global address (255) to inform Working Sets of the change^a
Inactive, but visible, to Hidden	(not applicable)	Send the On User-Layout Hide/Show message when the Working Set is inactive but visible
^a 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 a 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.

4.6.6 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, defined as the state in which all functions dependent 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.

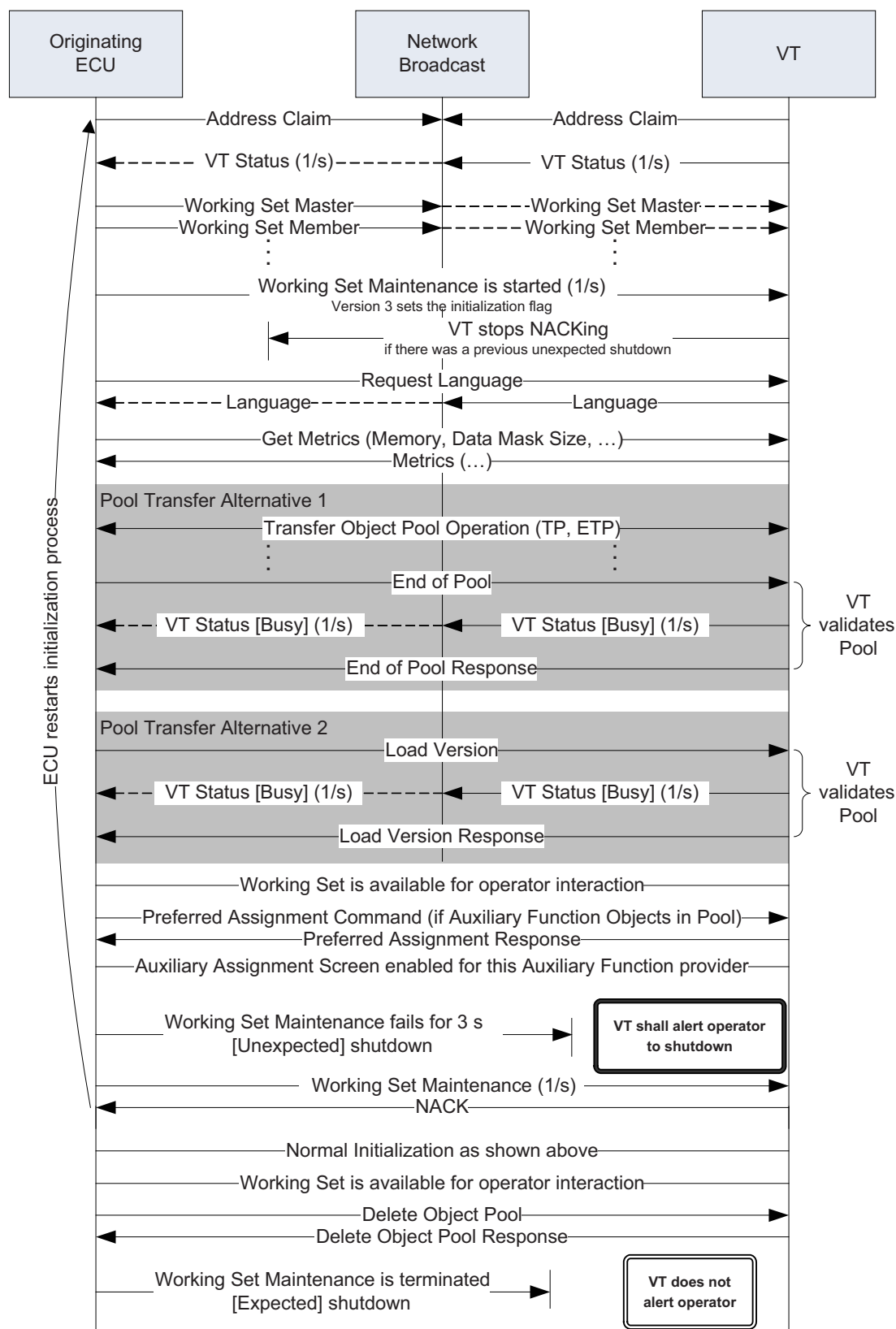
Each Working Set Master sends the Working Set Maintenance message once per second. The VT uses this message to ensure that each Working Set is still present. If the VT does not receive this message for a period of 3 s it is determined to be an unexpected shutdown of the Working Set Master (see Figure 6) 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 — not an unexpected shutdown situation, and which 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 — not considered an unexpected shutdown;
- 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 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 has control of the VT, 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 exist auxiliary assignments mapped to this Working Set, the VT shall remove them. When a Working Set's object pool has been deleted and the VT receives a Working Set Maintenance message from the missing Working Set, it shall NACK the message (see ISO 11783-3). The NACK message is sent to the Working Set Master's source address (see 4.6.4).



NOTE Solid arrows indicate destination-specific messaging, dashed arrows indicate global message.

Figure 6 — Initialization, unexpected shutdown and expected shutdown

4.6.7 Updating the operator interface

4.6.7.1 General

CAN has finite bandwidth available in support of all services (ISO 11783-3 to ISO 11783-14). 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, 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), and
- Inactive Working Sets, which have no active Data Mask and Soft Key Mask displayed reduce the frequency of, or eliminate, updates to the VT.

4.6.7.2 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. Changeable 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.

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.

4.6.7.3 Changing, adding and deleting objects

Objects can be completely redefined at runtime and 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.

4.6.8 Special objects

4.6.8.1 Container objects

A Container object is a special object used to

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

Figure 7 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.

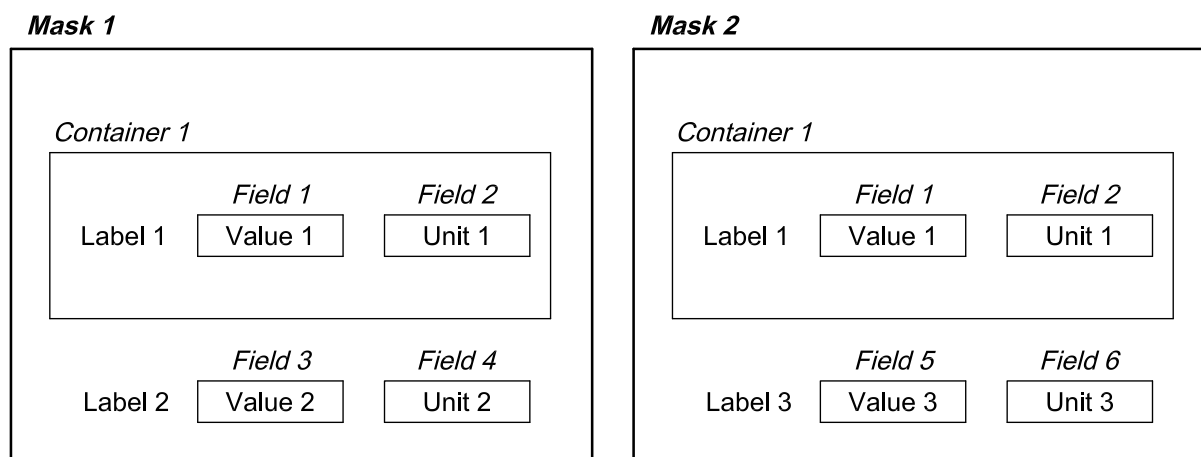


Figure 7 — Container reuse

Figure 8 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, whereas 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 runtime, the Working Set determines whether or not the particular feature is available. If not, the Working Set hides the container, as shown in Figure 8 b).

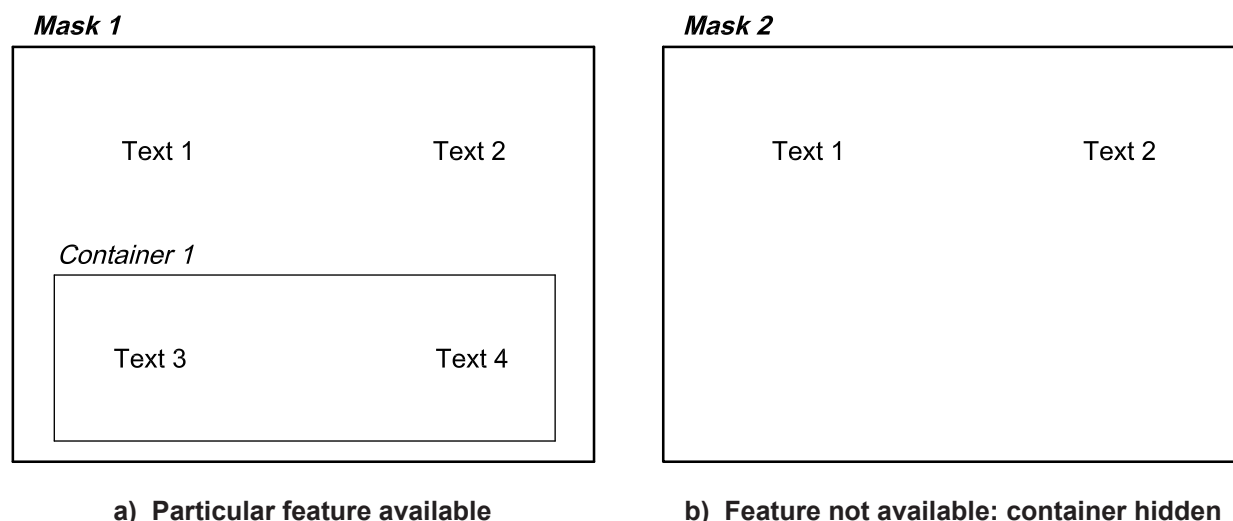


Figure 8 — Container used to hide objects — Examples

4.6.8.2 Attribute objects

There are five types of attribute object: 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.

4.6.8.3 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.

4.6.8.4 Macros

Macro objects are used to improve the performance of the operator interface, as follows.

- 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.
- 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 can be susceptible to a race condition and should be evaluated for predictable behaviour.

EXAMPLE 1 An Input String has a Macro to clear the Input String — On Input Field Deselection. The “Enter” means on an Input String of length greater than three 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 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, such as those created when a Macro triggers an event that references the same Macro, are not permitted, since they create infinite Macro loops inside the VT and could render the VT inoperable for all Working Sets.

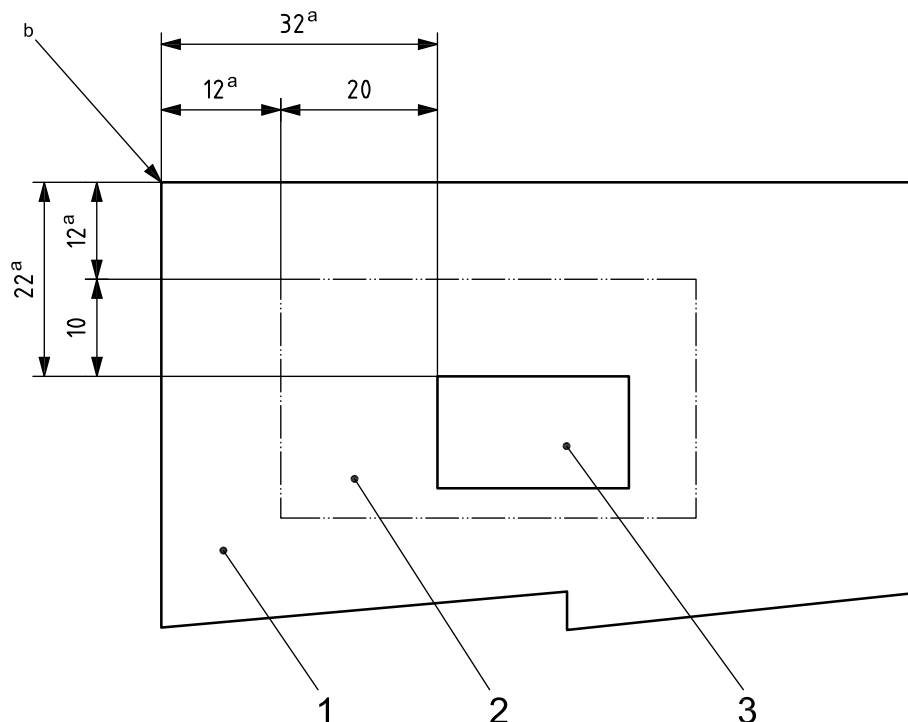
4.6.8.5 Object pointer

The Object Pointer object allows runtime modification of included objects. By changing the value of the 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.

When an Object Pointer is modified at runtime, 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 “Change Active Mask response” (see F.35) or “VT Change Active Mask” message (see H.14) to inform the Working Set, and may delete the object pool.

4.6.9 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 9 shows an example Data Mask with the relative locations of several objects.



Key

- 1 Data Mask
- 2 container
- 3 input field

^a Absolute.

^b Point of origin, 0, 0, absolute.

Figure 9 — Relative and absolute location of objects

4.6.10 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 if the object is currently visible.

EXAMPLE Figure 10 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 Figure 10 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 Figure 10 c). The VT may implement this refresh in a manner where the intermediate display is never seen by the operator.

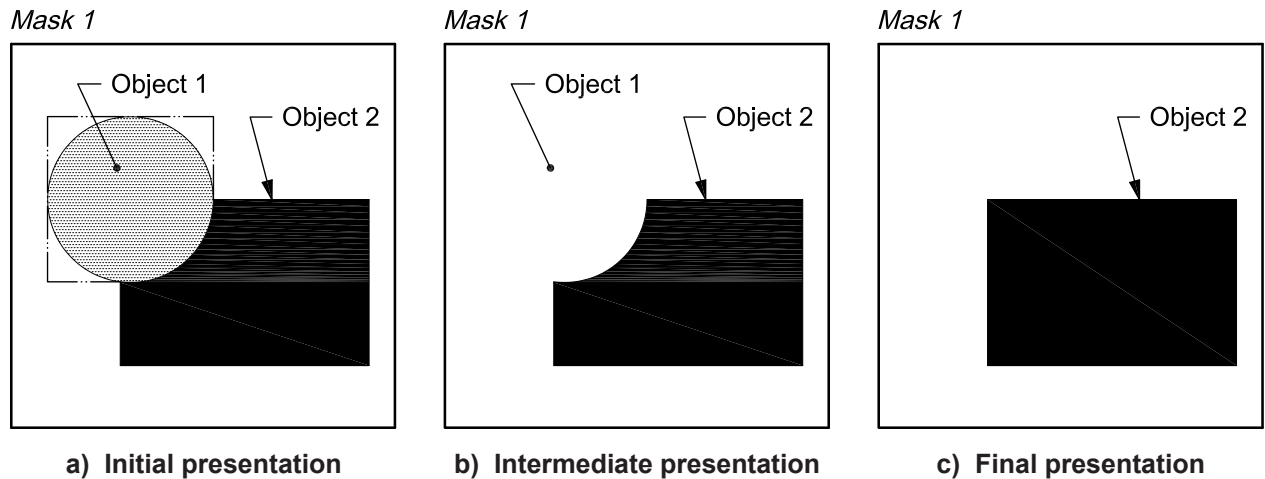


Figure 10 — Object changed or hidden — Display update

4.6.11 Alarm handling

Alarms allow a Working Set to display alarm information in the Data Mask area of the VT at any time. The Alarm Mask covers the entire Data Mask area. 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 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 at the same priority, 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 Mask or Data Mask, the next-highest priority Alarm Mask is processed in turn.

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 mask can be active per Working Set.
- b) The VT responds with an 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 acoustic signal shall be played. 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-version 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).
- d) The VT notifies the Working Set with a VT Status message sent to the global address.
- e) At some point, the operator acknowledges 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).
- f) The Working Set may choose to ignore the key press if the ACK is not allowed or may change the active mask to either an Alarm Mask or a Data Mask by using the Change Active Mask command on the Working Set.
- g) The VT responds with an Active Mask response.
- h) The VT displays a new mask (see Table 3).

Table 3 — VT behaviour on mask transition

Working Set's active mask attribute from/to	Is requester current active Working Set?	VT behaviour
Data to data	Yes	Hide current Data Mask, show new Data Mask.
Data to data	No	No visual change.
Data to alarm	Yes	Hide Data Mask, show Alarm Mask.
Data to alarm	No	If this is the highest priority alarm, deactivate the current Working Set and activate this Working Set.
Alarm to alarm	Yes	If this is the highest priority alarm, hide current Alarm Mask, show new Alarm Mask. Otherwise, deactivate this Working Set and activate the Working Set of the highest priority alarm.
Alarm to alarm	No	If this is the highest priority alarm, deactivate the current Working Set and activate this Working Set.
Alarm to data	Yes	If an alarm exists in another Working Set, deactivate this Working Set and activate the Working Set with the highest priority alarm. Otherwise, if this Working Set had the last visible Data Mask, hide the Alarm Mask and show the Data Mask. Otherwise, deactivate this Working Set and activate the Working Set which had the last visible Data Mask.
Alarm to data	No	No visual change.

4.6.12 Clipping

Most objects defined in this part of ISO 11783 have a given or implied size. The VT shall clip anything drawn outside the defined size of the object. Clipping is always done on a graphical (i.e. pixel) basis.

These clipping rules also apply to text objects. When the text does not completely fit inside the defined object area, in both wrapping and non-wrapping cases, the graphical clipping rules apply and the text is clipped on a graphical (i.e. pixel) basis (see Figure 11).

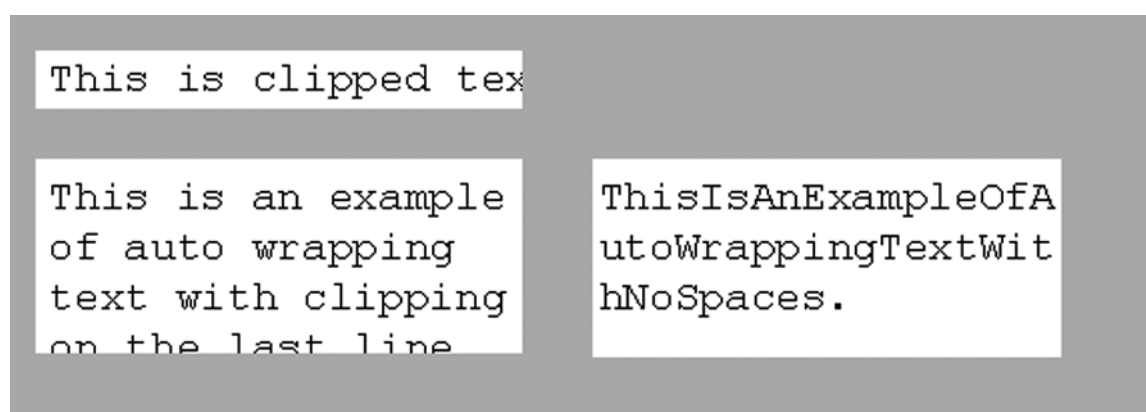


Figure 11 — Clipping examples

4.6.13 Scaling

4.6.13.1 General

The Working Set shall determine the size of the VT's Data Mask area and Soft Key designator and make appropriate adjustments to its object definitions. These adjustments may be applied either before or after transmission of its object pool to the VT, as long as the transmitted pool is not invalid for the VT (e.g. cannot transmit colour objects to a black-and-white VT and then change object to black and white). This gives complete control of the appearance of the masks to the Working Set.

4.6.13.2 Positions and sizes

The Working Set shall scale positions and object sizes to adapt to the VT's Data Mask area and Soft Key designator(s). Dot size (number of pixels) should also be adjusted.

4.6.13.3 Fonts

Font scaling is not provided by the VT. The Working Set shall apply a best-fit algorithm to determine and select the best font for the defined area. The Working Set shall also ensure that the VT supports the selected fonts and font styles. The smallest font size is 6×8 . If the scaled height and width of the original font is less than 6×8 , the 6×8 font shall be used. This could result in clipping, if the text is near the edge of the field object, or in text overlap, if two occurrences of text are too close together. Working Set designers shall be aware of this limitation and shall take the necessary steps.

4.6.13.4 Picture graphic objects

Bitmap graphics are automatically scaled by the VT according to the width attribute in the Picture Graphic object.

4.6.14 Operator input

Whenever the VT displays a Data Mask that contains one or more enabled visible input objects or buttons, it can be in one of the following states:

- a) Navigating;
- b) Data input.

NOTE When determining the visibility of an input object, the object shall be considered visible even under the following conditions:

- the object's width or height equals zero;
- the object is covered in its entirety by another object;
- the object is wholly outside the Data Mask;
- the object is wholly outside the visible area of a container, as defined by the container's width and height attributes.

Objects with width or height of zero, objects outside the Data Mask and objects outside a container's visible area are discouraged, as activation may not be possible (e.g. touch screen).

When a new active Data Mask is selected, the state is reset to "Navigating". If an Alarm Mask is selected, then the VT may remember the state and if the Working Set returns to the same Data Mask after showing the alarms then the state can be restored. If this approach is implemented, and the ECU returns to a different Data Mask after showing alarms, the VT shall consider it as a normal Data Mask change and send the appropriate messages (see Table 4).

The initial focus point and the tab order for navigating to the various input fields or buttons is VT proprietary, but the ECU shall be aware that the tab order may be defined by the definition order of the input objects in the parent object.

The VT may choose not to send the VT Select Input Object message for every object that the operator passes through while navigating. For example, if the navigation means is a rotary control, the focus will change rapidly while the operator is spinning the control. In such a case, the VT may choose only to send the VT Select Input Object message to the input object that loses focus and that which eventually gets focus.

In the “Navigating” state, the VT shall indicate to the operator which (if any) input is selected (has focus). The method of indication is proprietary to the VT. The VT may open an input object for data input as soon as the object gets focus, and it may remove focus when input is done. This is common, but not required, behaviour for touch screens.

Examples of focus indicators include (but are not limited to) adding a frame around the input field, changing the background colour of the input field, or momentarily highlighting the input field on a touch screen VT.

The VT designer should be aware that the use of the Select Input Object command by the Working Set can be a means by which the Working Set designer intends to focus the operator attention to an input field (e.g. a setup wizard where the recommended action is highlighted).

The VT shall indicate the disabled input objects. The means to represent disabled input objects is VT proprietary. Visual changes to disabled objects to indicate the disabled state shall not extend beyond the width/height of the object and the object shall remain legible.

Working Sets may apply a frame around, or distinct background colour to, input objects as an aid to the operator in identifying input fields.

In the “data input” state, the VT behaviour is proprietary, and the VT may cover part or all of the Data Mask while the object is open for data input. Changes to the attributes of an input object during the data input process shall not affect the value currently being input (e.g. changes to an Input Number Scale shall not alter the apparent value being input).

The VT reacts to navigation-related events (see Table 4).

Table 4 — VT reaction to navigation-related events

Current state	Command/Event	New state	Response frames ^a
Navigating	Select Input Object command (byte 4 = FF ₁₆)	Navigating	Select Input Object response
Navigating	Enable/Disable Object command (to disable the object which has focus)	Navigating	(The object becomes disabled and loses focus — VT may move focus to the next input object) Enable/Disable Object response VT Select Input Object message (on object which loses focus) VT Select Input Object message (on object which gets focus — if any)
Navigating	Change Active Mask command (only if the new mask is a new Data Mask, or if the new mask is an Alarm Mask and the VT does not store the state of the input object)	Navigating	VT Select Input Object message (on object which loses focus — if any) Active Mask response VT Status message (with new Data Mask) VT Select Input Object message (on object which gets focus — if any)
Navigating	ESC command	Navigating	ESC response (with error code indicating that no input is open for input)
Navigating	Operator activates a selected Button Object	Navigating	Button Activation message
Navigating	Operator navigates to a new object	Navigating	VT Select Input Object message (on object which loses focus — if any) VT Select Input Object message (on object which gets focus — if any)
Navigating	Operator opens the object for data input	Data input	VT Select Input Object message (on object which has/gets focus)
Navigating	Select Input Object command (byte 4 = 00)	Data input	Select Input Object response
Data input	Select Input Object command	Data input	Select Input Object response (with error code indicating that another input field is currently being entered)
Data input	Enable/Disable Object command (to disable the object which has focus)	Data input	(Object stays enabled and maintains focus) Enable/Disable Object response (with error code indicating that operator input is active)
Data input	Change Numeric Value command	Data input	Change Numeric Value response (with error code indicating that the object is in use)
Data input	Change String Value command (transport protocol)	Data input	Change String Value response (with error code indicating that the object is in use)
Data input	Change Active Mask command (only if the new mask is a new Data Mask, or if the new mask is an Alarm Mask and the VT does not store the state of the input object)	Navigating	VT ESC message VT Select Input Object message (on object which loses focus — if any) Active Mask response VT Status message (with new Data Mask) VT Select Input Object message (on object which gets focus — if any)

Table 4 (continued)

Current state	Command/Event	New state	Response frames ^a
Data input	Change Attribute command	Data input	Change Attribute response (with error code indicating that the object is in use)
Data input	Change List Item command	Data input	Change List Item response (with error code indicating that the object is in use)
Data input	Pool Update alters this object	Navigating	VT ESC message VT Select Input Object message (on object which loses focus — if any) VT Select Input Object message (on object which gets focus — if any)
Data input	Parent Container is hidden	Navigating	VT ESC message VT Select Input Object message (on object which loses focus — if any) VT Select Input Object message (on object which gets focus — if any)
Data input	Pointer to this object is changed to not reference this object	Navigating	VT ESC message VT Select Input Object message (on object which loses focus — if any) VT Select Input Object message (on object which gets focus — if any)
Data input	ESC command	Navigating	ESC response VT Select Input Object message (on object which has/loses focus)
Data input	Operator activates the ESC means	Navigating	VT ESC message VT Select Input Object message (on object which has/loses focus)
Data input	Operator activates the ENTER means	Navigating	VT Change Numeric Value message or Input String Value (even if the new value is the same as the old) VT Select Input Object message (on object which has/loses focus)

^a They shall be sent in the indicated sequence.

4.6.15 Soft key and button activation

Whenever a Key Object or Button Object or ACK key is pressed, released, or latched, the VT sends a Soft Key Activation message or Button Activation message to the working Set Master. If a Macro is associated with the key press, the VT executes it. Performance of the operator interface can be improved by associating a Macro with the key press event to cause another event, such as activating a different mask. See Tables B.11 and B.13.

If a Key Object or non-latchable Button Object is erased from the screen (e.g. due to a Change Active Mask command, Soft Key Mask, Hide/Show Object command, etc.) while activated, the VT shall send a Soft Key Activation message or Button Activation message indicating *released* to the erased object on its parent Data Mask. The VT shall then ignore the physical key until the operator has released it.

For example, when changing the visible Data Mask, the VT shall send the Soft Key Activation message indicating released for the activated object for the previous mask. It shall not send a Soft Key Activation message for the new mask.

When the “VT supports simultaneous activation of all combinations of Physical Soft Keys” bit (see D.9) is zero, the VT shall only support the activation of a single Soft Key at a time and only in the prescribed sequence of <no keys pressed> : <key pressed> : [<key held>] : <key released> : <no keys pressed>. If a second key is pressed while a first key has already been detected as pressed/held, it shall be ignored.

When the “VT supports simultaneous activation of all combinations of Buttons” bit (see D.9) is zero, the VT shall only support the activation of a single Button at a time and only in the prescribed sequence of <no Buttons pressed> : <Button pressed> : [<Button held>] : <Button released> : <no Button pressed>. If a second Button is pressed while a first Button has already been detected as pressed/held, it shall be ignored.

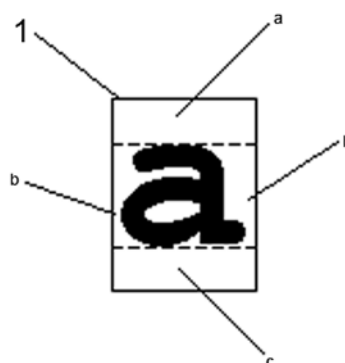
If a Key or Button is found to be pressed at power on, it shall not be reported as held; however, this can be cause for VT to report a diagnostic message to the operator.

4.6.16 Font rendering

4.6.16.1 Text justification

Text-based objects have a justification attribute. Field justification indicates how a text string is positioned horizontally and vertically within the field defined by the width and height attributes. In Version 3 and prior VTs, text justification was done on a character basis, but was not precisely defined. In Version 4 and later VTs, text justification is always done on a graphical (i.e. pixel) basis.

The extents of a text character (see Figure 12) shall be graphically justified as described in the following sections, but some white space is permissible if the VT's font-rendering engine reserves space for ascenders and descenders or for the character itself.



Key

- 1 graphical extents of the character
- a Ascender area can create white space depending on VT design.
- b Character rendering itself can create white space depending on VT design.
- c Descender area can create white space depending on VT design.

Figure 12 — Graphical extents of a character

During data input of a text-based object, the VT designer may choose to suppress justification until the field is closed after data input.

These modifications shall be done for visual justification — the stored value is not modified.

4.6.16.1.1 Horizontal left justification

When left-justified, the VT shall not remove any leading spaces; the first character in the string is positioned and visible at the left side of the text field area. If auto-wrapping is enabled, the rules of auto-wrapping overrule, and leading spaces on subsequent lines are trimmed before justification. If the string does not fit in the defined text field area, and auto-wrapping is not enabled, the string shall be graphically clipped on the right side. If auto-wrapping, justification rules apply to each new line. Blank lines are not removed.

In the examples below, the box represents the extents of the defined text field area.

EXAMPLE 1 Left justification, no auto-wrap, no leading spaces.

Left Justified

EXAMPLE 2 Left justification, no auto-wrap, one leading space.

Left Justified

EXAMPLE 3 Left justification, no auto-wrap, clipping on the right.

Left Justified Text

EXAMPLE 4 Left justification, auto-wrap, no leading spaces.

**This is left
justified,
wrapped text**

EXAMPLE 5 Left justification, auto-wrap, one leading space.

**This is left
justified,
wrapped text**

4.6.16.1.2 Horizontal middle justification

When middle-justified, the VT shall remove all leading and trailing spaces before justifying the string. The string shall be centred in the text field on a pixel basis. If the string does not fit in the defined text field area, and auto-wrapping is not enabled, the string shall be graphically clipped on the left and right sides. If auto-wrapping, justification rules apply to each new line. Blank lines are not removed.

In the examples below, the box represents the extents of the defined text field area.

EXAMPLE 1 Middle justification, no auto-wrap, no leading or trailing spaces.

Middle Justified

EXAMPLE 2 Middle justification, no auto-wrap, one leading and one trailing space (space is removed).

Middle Justified

EXAMPLE 3 Middle justification, no auto-wrap, clipping on the left and right.

iddle Justified

EXAMPLE 4 Middle justification, auto-wrap, no leading or trailing spaces.

**This is middle
justified,
wrapped text!**

EXAMPLE 5 Middle justification, auto-wrap, one leading and one trailing space (space is removed).

**This is middle
justified,
wrapped text!**

4.6.16.1.3 Horizontal right justification

When right-justified, the VT shall remove any trailing spaces before justification and the last character in the string is positioned and visible at the right side of the text field area. If the string does not fit in the defined text field area, and auto-wrapping is not enabled, the string shall be graphically clipped on the left side. If auto-wrapping, justification rules apply to each new line. Blank lines are not removed.

In the examples below, the box represents the extents of the defined text field area.

EXAMPLE 1 Right justification, no auto-wrap, no trailing spaces.

Right Justified

EXAMPLE 2 Right justification, no auto-wrap, one trailing space (space is removed).

Right Justified

EXAMPLE 3 Right justification, no auto-wrap, clipping on the left.

ght Justified Text

EXAMPLE 4 Right justification, auto-wrap, no trailing spaces.

**This is right
justified,
wrapped text**

EXAMPLE 5 Right justification, auto-wrap, one trailing space (space is removed).

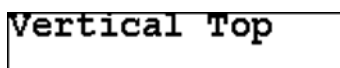
**This is right
justified,
wrapped text**

4.6.16.1.4 Vertical top justification

When vertical top justification is enabled, the VT shall display the text string starting at the extreme top of the defined text field area. This rule applies regardless of the auto-wrapping attribute.

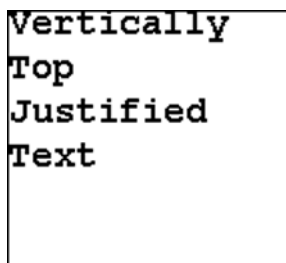
In the examples below, the box represents the extents of the defined text field area and the text is also left-justified.

EXAMPLE 1 Top justification, no auto-wrap.



Vertical Top

EXAMPLE 2 Top justification, auto-wrap.



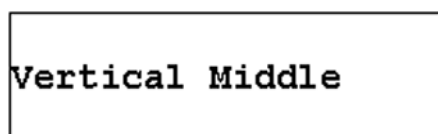
Vertically
Top
Justified
Text

4.6.16.1.5 Vertical middle justification

When vertical middle justification is enabled, the VT shall display the text string graphically centred vertically in the defined text field area. This rule applies regardless of the auto-wrapping attribute.

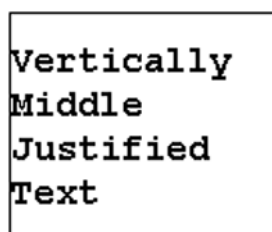
In the examples below, the box represents the extents of the defined text field area and the text is also left-justified.

EXAMPLE 1 Middle justification, no auto-wrap.



Vertical Middle

EXAMPLE 2 Middle justification, auto-wrap.



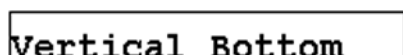
Vertically
Middle
Justified
Text

4.6.16.1.6 Vertical bottom justification

When vertical bottom justification is enabled, the VT shall display the text string with the bottom edge of the text block along the bottom edge of the defined text field area. This rule applies regardless of the auto-wrapping attribute. Blank lines are not removed.

In the examples below, the box represents the extents of the defined text field area and the text is also left-justified.

EXAMPLE 1 Bottom justification, no auto-wrap.



Vertical Bottom

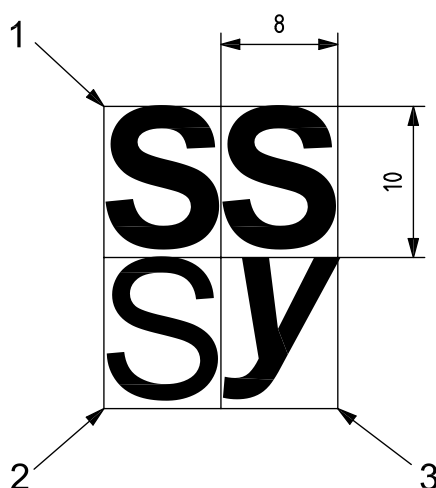
EXAMPLE 2 Bottom justification, auto-wrap.

Vertically
Bottom
Justified
Text

4.6.16.2 Non-proportional fonts

The VT shall support non-proportional block fonts. Sizes are always given in X–Y pairs. For example, 8 × 10 indicates a character size of 8 pixels wide by 10 pixels high. Characters shall not exceed the size of the box defined by the font size, regardless of style. An 8 × 10 font set to bold and italics, for example, shall still fit inside the 8 × 10 pixel area. It is suggested that space be left on the bottom and right sides on the inside of the box to accommodate characters placed side by side or row by row (see Figure 13).

Dimensions in pixels



Key

- 1 bold
- 2 normal (upright)
- 3 bold italic

Figure 13 — 8 × 10 fonts — Example

The VT designer may choose the font sizes and styles to be made available but the default 6 × 8 font, normal (upright) style, is a minimum requirement. The Get Text Font Data message can be used by a Working Set to determine the VT's font capabilities. Characters can be rendered transparent (background shows through) or opaque (solid background colour), depending on the options attribute of the applicable object.

4.6.16.3 Proportional fonts

In Version 4 and later VTs as an option, the VT design may allow for proportional font rendering. In this case, the width of each character is variable and the height attribute is fully scalable in the range from 8 pixels up to and including the largest supported font height as identified in D.7. As a result, only the height attribute of the font size applies and the width attribute is ignored. Therefore, the rendered characters shall not exceed the height of the chosen font size.

For example, if the VT responds to a Get Text Font Data message with Byte 6 = 10₁₆, and Byte 7 = 02₁₆, then the largest report non-proportional font size is 48 pixels wide by 64 pixels in height. If the VT has indicated support for proportional font rendering (Byte 8 bit 7 = 1), then the VT supports a proportional font height from 8

up to and including the value 64, in one-pixel resolution. Due to characteristics of the implemented font rendering engine, and the shape of the characters, one-pixel resolution may not be detectable for every character, even though it is supported.

Normal clipping rules still apply and Working Set designers have to be aware of the variable nature of the font width and therefore assign sufficient width to the text field. Implements can query the VT's font capabilities, including the support of proportional font rendering, via the Get Text Font Data message, and may choose the proportional rendering option in the font style attribute of the Font Attributes object. If the implement requests a proportional font rendering and the VT does not support this option, the VT shall respond in one of two ways:

- if a Font Attributes object indicates proportional font during the validation of an object pool, the object pool shall be rejected;
- if a Change Font Attributes command is received with invalid size and/or font type, a Change Font Attributes response shall be sent, indicating an error in the size and/or type.

The following general rule applies for the support of proportional font rendering:

- A VT that indicates support for proportional fonts shall support the full height scaling range of 8 to the largest supported font size as identified in Get Text Font Data response. Character width is recognized to be variable.

Before using proportional fonts, Working Sets shall query the VT to determine if proportional font rendering is supported by the VT. If not supported, only non-proportional fonts shall be used in the pool upload and subsequent Change Font Attributes commands.

4.6.16.4 Auto-wrap

If the amount of text to display is longer than the width of the text object, and auto-wrap is enabled, then regardless of non-proportional or proportional rendering, the VT shall format and wrap the text into the next line(s). The text shall not exceed the boundaries defined by the width and height of the text object. Wrapping shall occur under these conditions.

- Space (20_{16}) between words.
- Soft Hyphen (AD_{16}). When wrapping occurs on the soft hyphen, the soft hyphen shall be shown before the line break; otherwise the soft hyphen is not shown.
- Hyphen ($2D_{16}$). Wrapping can occur between a hyphen ($2D_{16}$) and the following character, when the Wrap on Hyphen option bit in text objects is TRUE.
- If there is no Space (20_{16}) in the line, then wrap on the last wholly visible character in the line (see Figure 11).
- At line end (see 4.6.16.5).

Leading space characters on the next line shall be suppressed and not considered in additional auto-wrap decisions.

4.6.16.5 Non-printing characters in strings

Version 3 VTs and prior allowed CR, LF, BS and 0x00 control characters in strings. The rendered presentation was not precisely defined.

The following clarification applies to Version 4 VTs and later:

- BS (Back Space) is ignored, as are other characters denoted with scissors in Annex L;
- Single CR (Carriage Return) is interpreted as a line end;
- Single LF (Line Feed) is interpreted as a line end;
- Sequence CRLF is interpreted as a line end;
- 00₁₆ or 0000₁₆ (WideChar) shall terminate the presentation, even if this occurs before the defined string length.

All characters following the terminating zero shall be ignored, both for data input and presentation. Editing may increase the apparent string length up to the defined string length, but shall not limit the editing of strings, which is controlled by the length attribute.

Based on these definitions, Sequence LFCR is interpreted as two line ends.

Other non-displayable characters shall not advance the cursor during drawing or alignment decision-making. The VT can remove these characters from the string to facilitate more efficient processing.

String data	VT presentation		
	Left alignment	Middle alignment	Right alignment
String 1: ABCDEF _{CR} GHIJKL or ABCDEF _{LF} GHIJKL or ABCDEF _{CRLF} GHIJKL	ABCDEF GHIJKL	ABCDEF GHIJKL	ABCDEF GHIJKL
String 4: MNOPQR _{LF} CRSTUVWX	MNOPQR ¶ ^a STUVWX	MNOPQR ¶ ^a STUVWX	MNOPQR ¶ ^a STUVWX
^a “¶” is shown for examples and would not be visible on the VT.			

Figure 14 — CR and LF application to test strings

4.6.16.6 String encoding

Text strings can be encoded with either 8 bit characters (chars) or Unicode/ISO 10646 characters (WideChars).

VT Version 3 and prior supported font types are shown in Tables L.1 and L.2 .

VT Version 4 and later supported font types are shown in Tables L.1 to L.7.

The character set is indicated by the font type attribute of the Font Attributes object.

WideStrings shall be encoded according to UTF-16 (Unicode Transformation Format 16 bit) and therefore always start with the Byte Order Mark (BOM) character FEFF₁₆. BOM is not a displayable character and is not considered part of the text string.

UTF-16 allows both big and little endian encoding, but ISO 11783-6 WideStrings shall always be encoded as little endian.

BOM is used to distinguish between 8 bit strings and WideStrings.

If the first two bytes are FF_{16} , FE_{16} , it is a WideString; otherwise, it is an 8 bit string.

The length attribute of an object or message always indicates the number of bytes in the text string; therefore, the number of characters in a WideString shall be found as

$$\text{number of WideChar} = \text{length}/2 - 1$$

e.g. $FF, FE, 41, 00, 42, 00, 43, 00 \Rightarrow$ "ABC", [length = 8, number of WideChar = 3]

This is only correct if all characters are from Code Plane 0. The characters above $FFFF_{16}$ require 4 bytes per character (see "surrogate pair" below).

If the length attribute does not indicate an even number of bytes, the last byte is ignored.

The VT may support any character defined by Unicode/ISO 10646, but as a minimum it shall support the characters listed in Table L.7.

The font type attribute of the Font Attributes object is ignored for WideStrings.

The VT shall display WideStrings even if the WideStrings contain characters which are not supported by the VT.

The VT shall substitute unsupported characters by a displayable character. The displayable character may be VT proprietary (e.g. "□").

The encoding of Input String values shall not be changed by the VT, i.e. if an Input String object contains (or references) a WideString, the VT Change String Value message sent by the VT shall also contain a WideString.

Characters above $FFFF_{16}$ are represented by a 32 bit "surrogate pair", which consists of a high surrogate followed by a low surrogate.

The surrogate pair is constructed as follows:

- $S = \text{character} - 10000_{16}$;
- High surrogate = $D800_{16} + (S \text{ shifted } 10 \text{ bits to the right})$;
- Low surrogate = $DC00_{16} + (10 \text{ least significant bits of } S)$.

The highest character defined by Unicode and ISO 10646 is $10FFFF_{16}$, corresponding to the surrogate pair $DBFF_{16}$, $DFFF_{16}$.

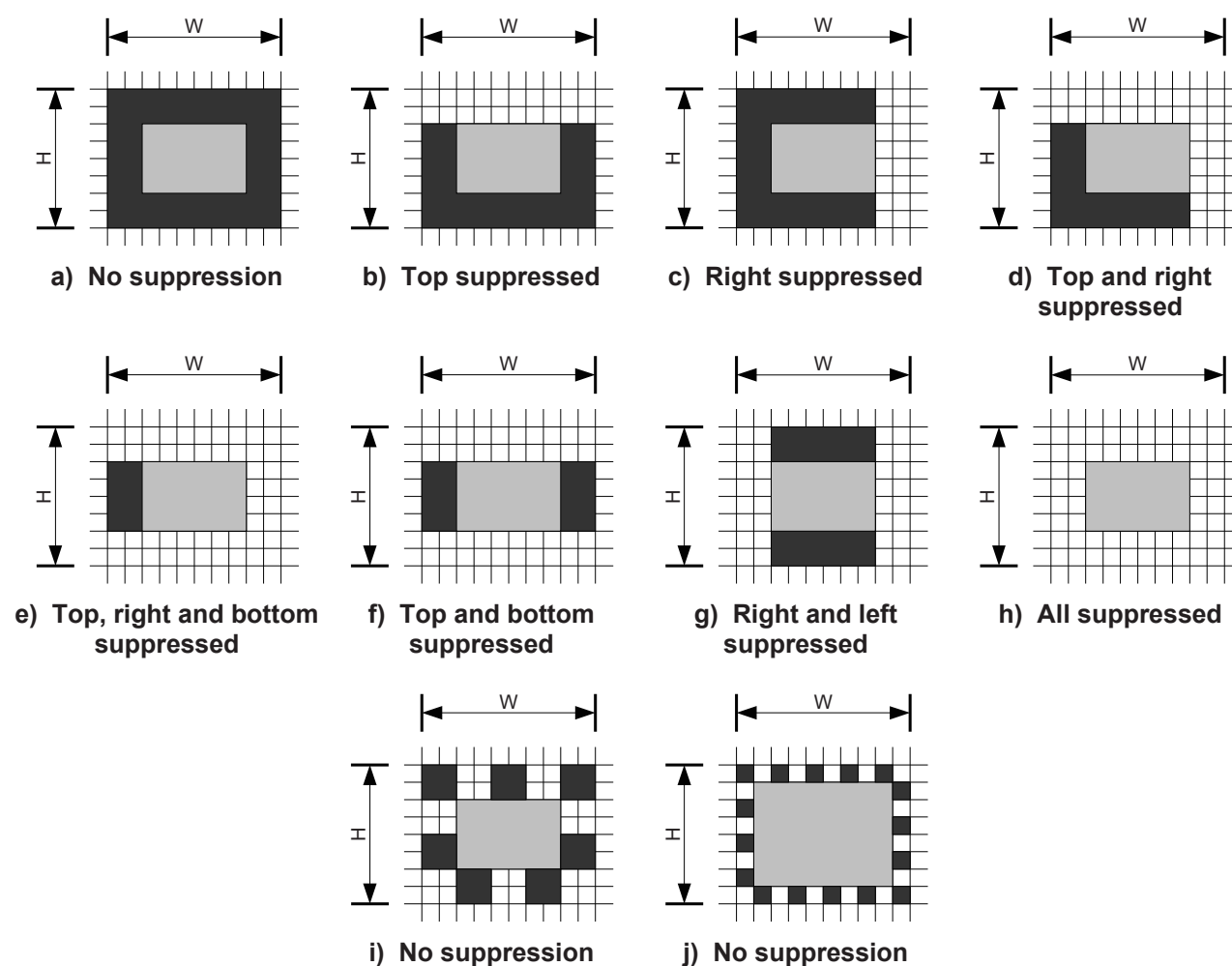
4.6.17 Filling output shape objects

When solid-filling output shape objects on the VT, flood-fill or boundary-fill type algorithms are not suitable, since objects can be overlaid and interrupt the fill. There are also performance problems with this type of approach. Scan-line type fills shall be implemented. In addition, only the interior area of the object, not including any pixel that is/would be part of the border, shall be included in the fill. Incorrect filling would be particularly visible when line art is used on the border or when the border is completely or partially suppressed (i.e. rectangles).

The following rules apply to Pattern fills made from Output Shape objects.

- The pattern shall be a Picture Graphic object whose width is an integer divisible by 8. The raw data is used for the pattern and the object is not scaled regardless of the attributes of the object.
- The upper left corner of the pattern buffer shall be anchored to the upper left corner of the VT's physical Data Mask Area and repeats across and also down. This ensures that the pattern matches between objects in the Data Mask Area and that the pattern fill looks the same on all VT designs.
- Transparency and flashing option attributes shall be ignored for fill patterns.

Filling and line suppression examples are shown in Figures 15 to 17.



Rectangles a) to h): width = 10, height = 8, line width = 2, fill type = solid grey.

Rectangle i): width = 10, height = 8, line width = 2, fill type = solid grey, line art 1010.

Rectangle j): width = 10, height = 8, line width = 1, fill type = solid grey, line art 1010.

Figure 15 — Rectangle line suppression and filling examples

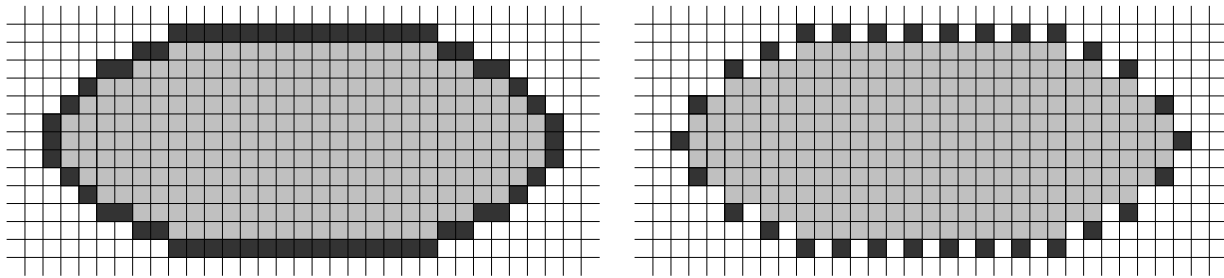


Figure 16 — Ellipse filling examples (without and with border line art)

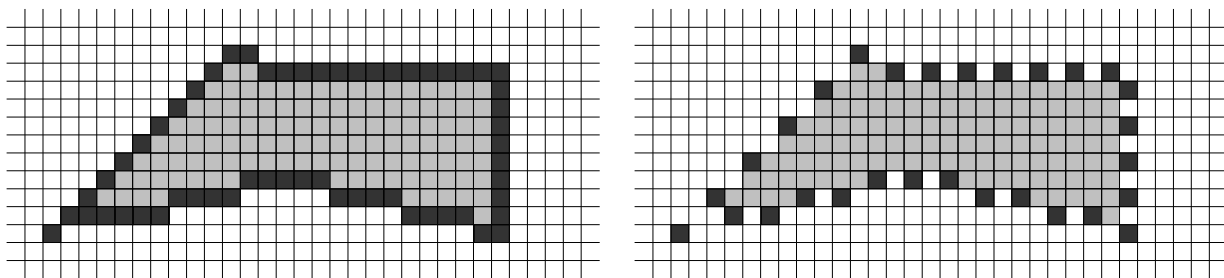


Figure 17 — Polygon filling examples (without and with border line art)

4.6.18 Events

Manipulation of objects in an object pool by a Working Set or by the VT causes certain events to occur. Events can be caused by commands or by VT actions in response to a command or by other events. Many objects defined by this part of ISO 11783 have an optional list of event and Macro groupings. If the occurring event has one or more Macros associated with it, the Macro or set of Macros is executed by the VT when the command has been accepted by the VT as a valid command. Macros are executed in the order they are encountered in the event/Macro list. Using events and Macros can make the VT operator interface more responsive, since the Working Set Master does not need to be directly involved in responding to the event.

EXAMPLE 1 A Soft Key press event could cause an appropriate Data Mask to be made active.

EXAMPLE 2 A Soft Key press event could cause two separate Macros to execute — one to make a new Data Mask active and the other to control the audio signal.

EXAMPLE 3 A Change Numeric Value with an invalid value causes rejection of the command and an associated Macro is not executed. A Change Numeric Value with a valid value, even if the value is the same as already in the value field, will cause the associated Macro to execute.

EXAMPLE 4 A container with an On Hide Macro is already hidden. A Hide Show command with the parameter set to hide the container is valid and will cause the associated Macro to execute.

NOTE The same event ID can be listed more than once in the event/Macro list of any given object.

4.6.19 Touch screens and pointing devices

The VT design can optionally support a touch screen or a pointing method such as a mouse or joystick. The Working Set can determine the VT's capabilities in this regard by using a Get Hardware message and then make necessary adjustments to its object pool. A Button Object is defined to allow touchable or clickable buttons to be included in a Data Mask. A Pointing Event message is defined to allow the VT to notify the active Working Set that an area of the Data Mask *not associated* with a button or other input object has been touched or clicked on.

4.6.20 Proprietary Means

There are various proprietary means supported by the VT (Proprietary objects, Proprietary events, Proprietary colours, Proprietary commands, and Proprietary fonts). Use of these proprietary means where the Working Set is provided by a different manufacturer than that which provides the VT is not recommended, in order to provide maximum ISO compatibility. As these items are proprietary, the Working Set or VT manufacturer may change their proprietary means without disclosure.

Further, and in the context of the Proprietary objects, it is not possible to parse an object for which the definition is not known, and attempting to do so can cause the VT to reject the pool.

4.6.21 VT Number

By default, VTs shall be factory set to function instance zero (0), but retain the function instance as configured by the operator. A mechanism is required in order to conveniently resolve conflicts in the case where there are multiple VTs with the same function instance and in the case where there is no VT with function instance zero. The VT shall be responsible for providing a proprietary means for setting the function instance from the display itself. This means shall ensure that duplicate function instances between VTs are not created. The new function instance shall not be used until a reboot of the VT is performed. The VT with function instance zero (0) is defined as the "primary VT".

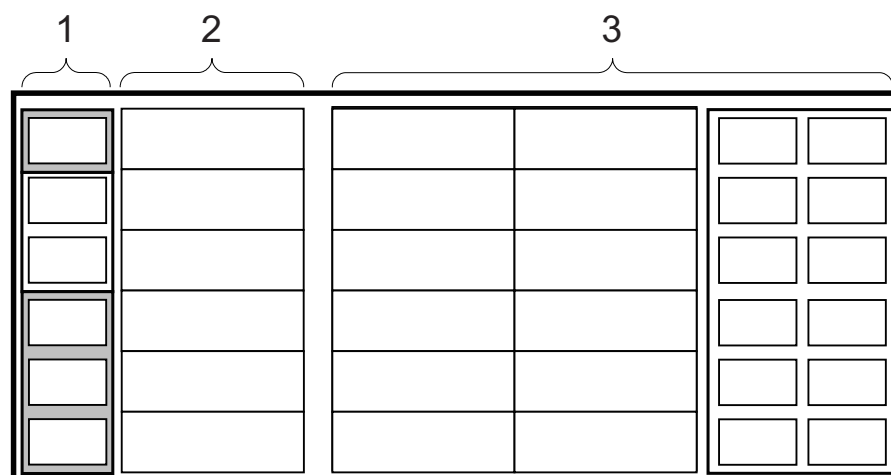
The proprietary means to set the function instance shall represent to the operator a VT Number. In this way, VTs from all manufacturers will present a consistent numbering scheme for the operator to choose the primary (and secondary) VT(s). To facilitate easy VT identification, the Identify VT message may be used.

4.7 Displaying data from Multiple Working Sets on one Mask (Version 4 VT and later)

4.7.1 General

The following subclauses are applicable to Version 4 and later VTs.

The VT may provide a means where data from multiple Working Sets can be made available on one screen. Depending upon the VT design, it is also possible that data from multiple Working Sets can be made available simultaneous to the VT's standard Data Mask and Soft Key Mask. See example in Figure 18.



Key

- 1 Key Group Objects in non-VT area (shaded regions indicate independent Key Groups)
- 2 Window Mask Objects in non-VT area
- 3 VT area (presenting standard VT screen or User-Layout Data Mask and User-Layout Soft Key Mask)

Figure 18 — Displaying data from multiple Working Sets — Example

4.7.1.1 Minimum requirements

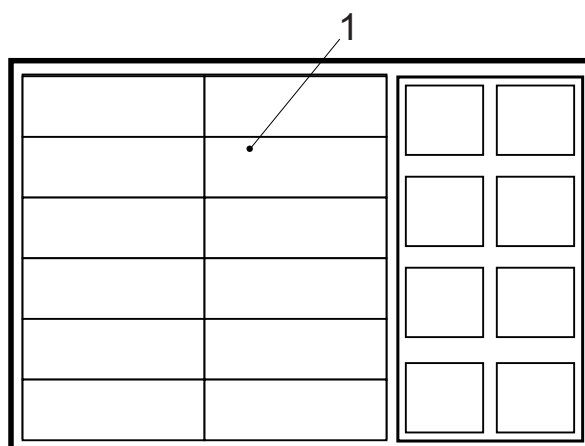
This is an optional feature. However, as a minimum, the VT shall be required to parse the Window Mask and Key Group objects even if not supported. This allows Working Sets to upload these objects to all Version 4 and later VTs without modification of the object pool. If this feature is supported, the VT shall support the “free form” (type 0) window mask type of the Window Mask object as a minimum. Any number of the other non-zero window mask types may also be supported as desired. Working Set designs could desire the use of any of the window mask types, so implementation by the VT design of all window mask types is recommended. If the Working Set uploads a Window Mask object with an unsupported window mask type, the VT shall parse, but ignore, this object and no errors shall be raised. Unsupported window mask types would not be presented to an operator for selection.

If the Working Set chooses to participate in this feature, it shall upload Window Mask and Key Group objects as part of the object pool. Since the VT shall parse while ignoring any Window Mask objects with unsupported window mask type, no modification of the object pool is necessary.

4.7.2 User-Layout Data Mask

The VT may support any number of User-Layout Data Masks. User-Layout Data Masks are special Data Mask objects that are owned by the VT. The VT shall provide a mechanism to allow the operator to access the available User-Layout Data Masks.

Each User-Layout Data Mask is the same size as a standard Data Mask object but is divided into a grid of window cells with exactly two columns and six rows (see Figure 19).



Key

- 1 User-Layout Data Mask showing 12 window cells

Figure 19 — User-Layout Data Mask

4.7.3 Window Mask object

Window Mask objects, optionally supplied in the object pools of Working Sets, are placed into a specific User-Layout Data Mask grid as desired by the operator of the VT. An individual Window Mask object may take up more than one Window Cell, and may take up the entire 2×6 grid. 1×1 Window Masks are recommended, where possible, to allow the operator to select from a wide range of Window Mask objects to display in each User-Layout Data Mask (see Figure 20).

NOTE Window Mask objects are not limited to the sizes shown in Figure 20, which is an example only.