



Training: ISOBUS Virtual Terminal (VT)

Michael Schmidt

Technical Leader, Embedded Software

AGCO Advanced Technology Solutions

Objectives

- Explain the purpose of a Virtual Terminal (VT)
- Provide helpful information for developing a VT Client application for an ISOBUS implement

Topics

- What is an ISOBUS Virtual Terminal (VT)?
- Advantages of developing a VT Client
- Minimum requirements for a VT Client
- VT Client Startup and Shutdown
- Maintaining the Connection – “Alive” Messages
- Message Transport Methods
- VT Object Pools
- Commands
- User Input
- Auxiliary (AUX) Controls
- VT Versions

What is an ISOBUS Virtual Terminal (VT)?

- Graphical display that follows the ISO 11783-6 Specification
- Displays data to the user
- Receives input from the user

Advantages of developing a VT Client

- Less clutter in the cab
- Ease of use
 - Single place for the user to make settings, get diagnostic information, set controls, etc.
- Shorten development time
 - No need to develop a separate user input / feedback device for each implement
- Help to standardize the user interface
 - Each terminal has a standard way to enter numbers, key in text, select from lists, push buttons, etc.
- Reduce operator training

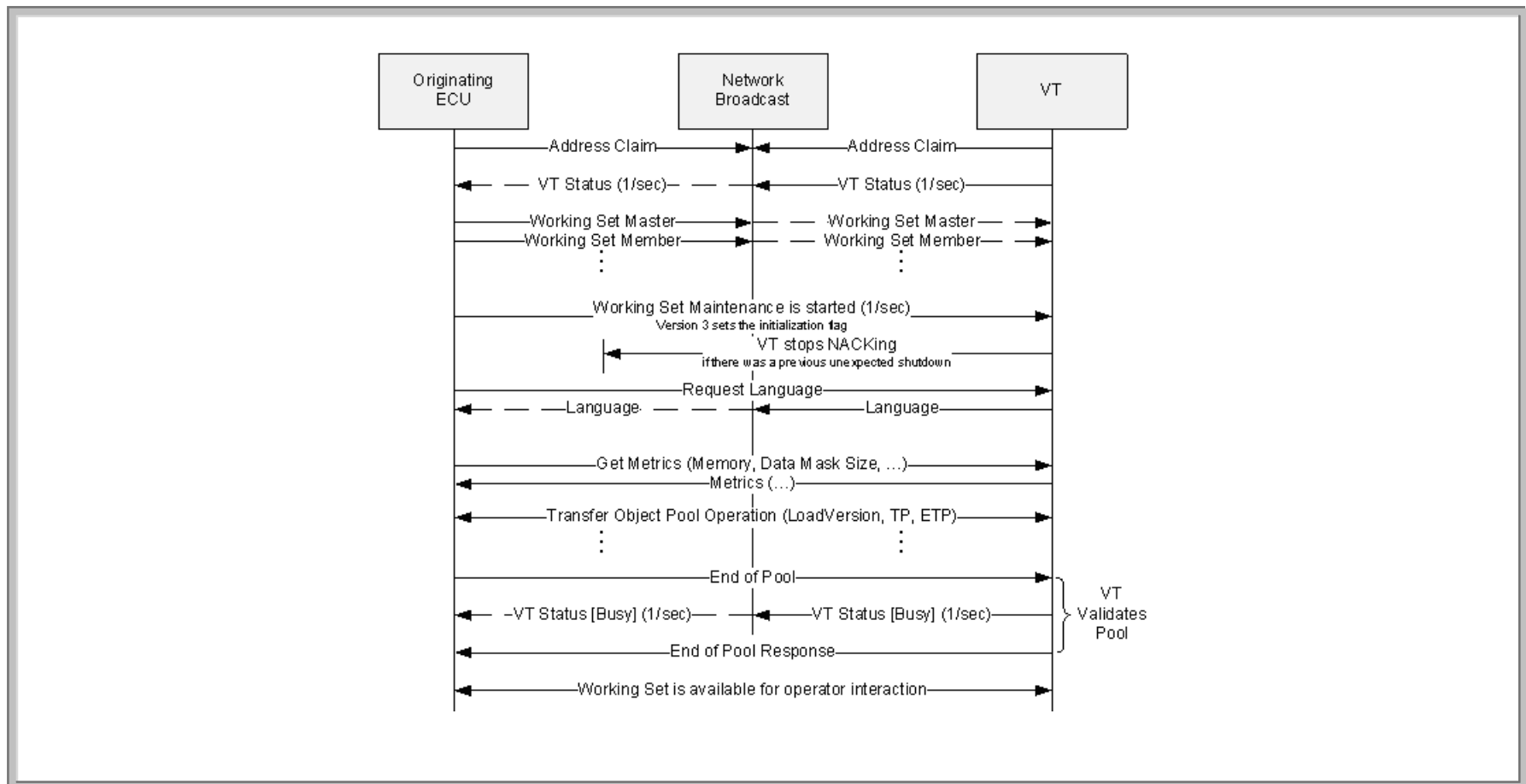
Minimum Requirements for a VT Client

- Must follow the Connection Management procedure to make a connection to the VT
- Must upload an object pool with at minimum:
 - Exactly one Working Set object
 - This must contain at least one other object
 - Reference to an Active Mask
 - List of supported languages
 - At least one Data Mask object
- Must send the Working Set Maintenance “alive” message once per second

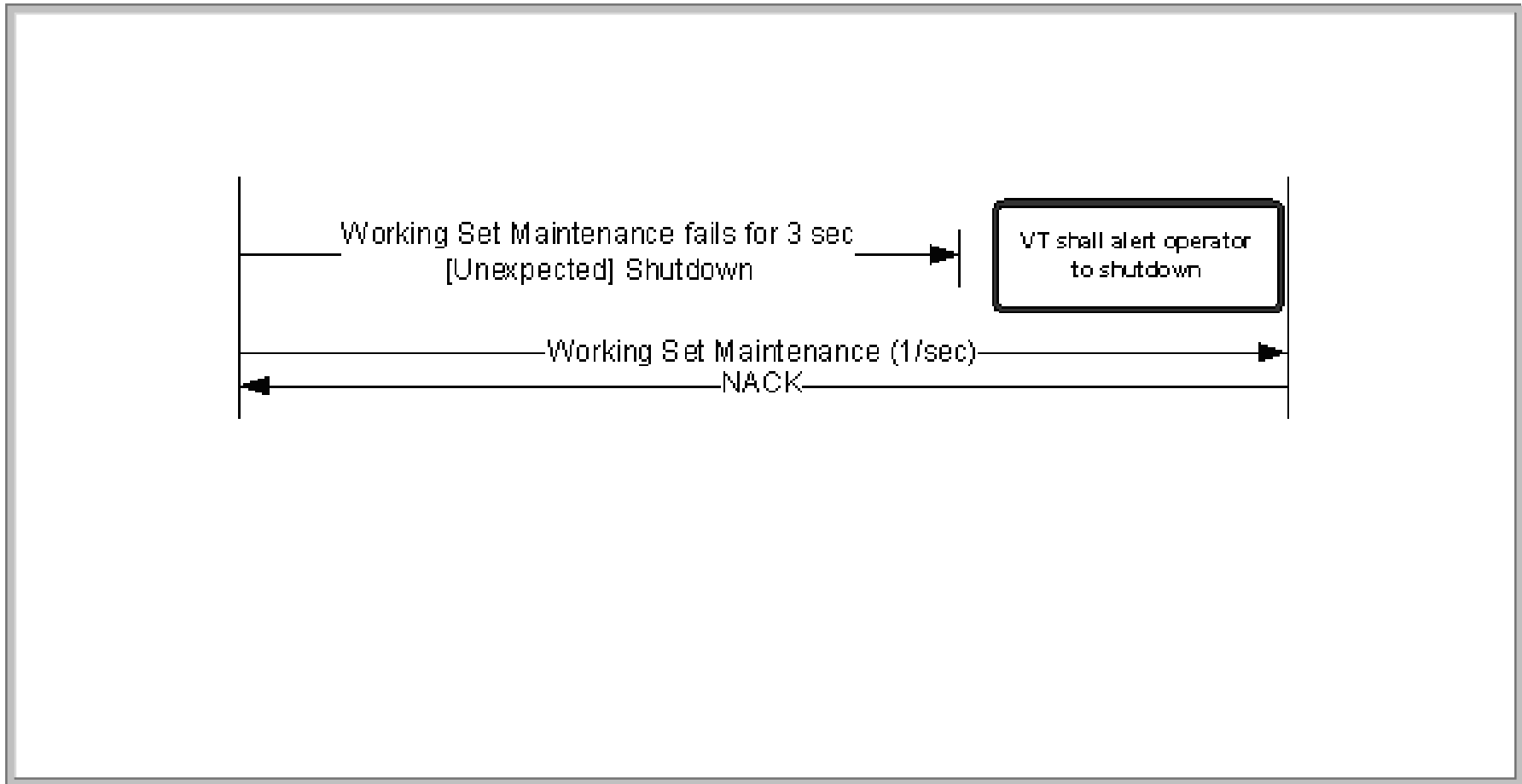
VT Client Startup and Shutdown

- Described in full detail in section 4.6.4, but generally...
- At startup, the VT Client must do the following:
 - Claim an Address on the bus
 - Listen for a VT
 - Declare itself and any “partner” ECUs to the VT
 - Begin sending the 1-second periodic “alive” message
 - Ask the VT what language and units that the operator wants it to display
 - Start the sequence to load the user interface (VT object pool)
- At shutdown, the VT Client should:
 - Tell the VT that it is shutting down gracefully by sending the Delete Object Pool message
 - Stop sending the “alive” message

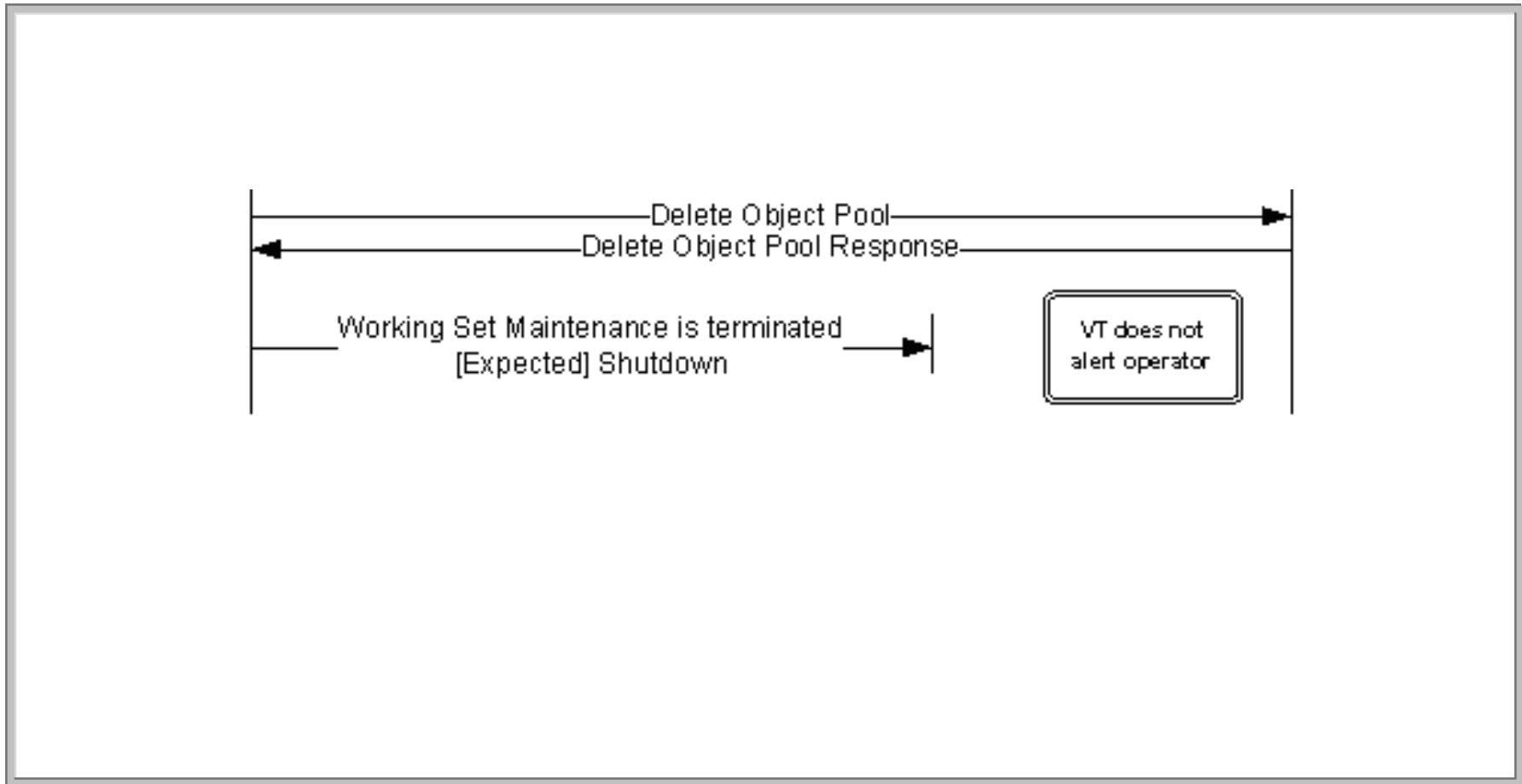
VT Client Startup



VT Client Unexpected Shutdown



VT Client Normal Shutdown



Maintaining the Connection – “Alive” Messages

- VT->ECU: VT Status Message
- ECU->VT: Working Set Maintenance Message

Message Transport Methods

- Single CAN frame (0 - 8 bytes)
- Transport Protocol (TP) (9 - 1,785 bytes)
- Extended Transport Protocol (ETP) (1786 - 117,440,512 bytes)

VT Object Pools

- The complete lists are given in the standard, but a few examples are given below
- Objects
 - Input String
 - Output Number
 - Bar Graph
 - Line Attribute
- Events
 - On Activate
 - On Change Size
- Macros

Commands

- The complete list of commands is given in the standard, but a few examples are given below
- VT->ECU
 - Activation Messages
 - VT change numeric value
- ECU->VT
 - Technical data messages
 - Get number of soft keys
 - Non-volatile memory operations commands
 - Store version
 - Object / Attribute manipulation commands
 - Hide/show object
 - Change child location
 - Set audio volume

User Input

- Input objects
 - Input Numbers
 - Input Strings
 - Input Lists
 - Input Boolean
- Button type objects
 - Buttons
 - Soft Keys
- Mouse / Touch Screen Input
 - Pointing Events
- Auxiliary Inputs

Auxiliary (AUX) Controls

- Auxiliary Inputs
- Auxiliary Functions

VT Versions

- Currently there are two VT Versions
 - Version 2
 - Version 3
 - Will not support the AUX controls from Version 2
 - Contains many clarifications which also apply to Version 2
 - Contains some new objects
 - Allows for multiple VTs to be on the same CAN bus
 - Allows multiple VT Clients to be displayed simultaneously in different view ports
 - Support for Unicode characters
 - Many other new features and enhancements