VLCB Versatille Layout Control Bus

Event Teach Service Specification
Service# 4 Teach

Version 1.3, March 2025, for Service version 2

Compatible with CBUS ® 4.0 Rev 8j

VLCB Event Teach Service Specification

0.1 Table of Contents

0.2 Document History

Date	Changed by	Summary of changes	Service version
22nd December 2022	Ian Hogg M.5144	Initial document	1
14 April 2023	Ian Hogg M.5144	Changed name to VLCB	1
1 Feb 2024	Ian Hogg M.5144	General tidy up and added clarifications	1
10 Feb 2024	Ian Hogg M.5144	Updated the list of GRSP response errors	1
29 June 2024	Ian Hogg M.5144	Changed name from Event Teaching to Event Teach. Removed ESD bytes that were duplicates of PARAMs. Opcodes for Event Indexes no longer supported.	1
12 February 2025	Ian Hogg M.5144	Added clarification regarding dependencies between NVs and NVs with EVs.	1
10 March 2025	Ian Hogg M.5144	REVAL and REQEV for EV#0 follow up responses now dependent upon not being in FCU-compatibilty mode	2

1 VLCB Services

This document describes the service related to teaching events following the CBUS™ mechanism. This is an optional service, which may be added to the MNS.

This service is likely to be used in combination with one or both of the Event Producer Service and the Event Consumer Service.

While the above are reported as separate services, practically these can be implemented as a single library, since they have very similar needs, with facilities to characterise the events as producer, consumer, or both. In addition, Learning is intimately related to the definition, storage, and search functions included with Events. While Short and Long events (covered in section 2.1) can be considered as separate, again, it is practical to implement them together.

This service differs from the Indexed Event Teach service (#7) as it uses node number and event number eventId as the key into the event data instead of an event index. Modules should select either the Indexed Event Teach service or the Event Teach Service, it is not expected that modules would need to implement both of these event teaching services.

The Event Teach Service is not dependent on any other service but it is expected to be used in combination with the Event Consumer and Event Producer Services. It should also be compatible with the Consume Own Event Service and the Event Acknowledge service.

2 Introduction

Please refer to the "VLCB Event Model" document for further information about eventIds, event producers, event consumers and event variables.

For events to be useful, two or more nodes have to agree on the meaning of a particular event identifier. This is done by 'teaching' the nodes to use the same event identifier so that the producer can send it on a specific change of state, and the consumers can use it to perform some resulting action.

Events are taught to modules by configuring EVs for a specific EN/NN event identifier. The EVs determine the circumstances under which an event is produced or the actions to be performed when an event is consumed.

2.1 Producer/Consumer

The Event Teach Service has no explicit distinction between produced events, consumed events, or events that are both produced and consumed. The application is free to assign as it wishes. However, such an assignment must be documented. In addition, an MDF (Module Description File) should be written so as to match this assignment.

3 Event Teaching

3.1 Learn mode

A node must be placed into Learn Mode before Teaching can take place, by sending a MODE(Learn) message. And return to Normal-mode by sending the MODE(Normal) message. NB: A module in Learn mode continues to receive and respond to Normal mode messages, i.e. Learn mode is an extension of Normal mode.

For compatibility with CBUS a NNLRN (Set specific Node into Learn-mode) message can also be used to put the node into Learn mode and a NNULN message can be used to take a specific node out of Learn-mode, and return it to Normal-mode.
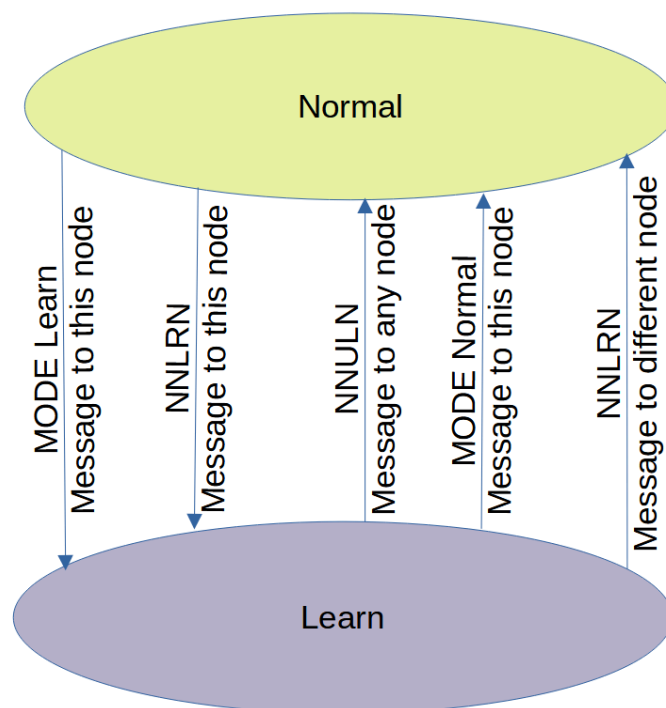
NOTE: Only one module may be in Learn Mode at a time. Should a module be in Learn-mode and receive an MODE(Learn) or NNLRN addressed to another module, it **must** immediately exit Learn-mode.

Factory default is for a module to NOT be in Learn Mode. Upon power-up the module also must NOT be in Learn Mode.

3.2 Teaching Long and Short events

Note that there are no separate messages to teach Long and Short events. However, since Long events (pNN:EN) must not have a pNN of zero, Short events can be taught by sending the event identifier as (00:EN). How this is handled in a module is implementation specific, and not part of this document. However, three common methods would be to have a single

table and maintain the (00:EN) form for Short events; tag the event identifiers; or maintain two tables, one each for Long and Short events.

Since the only way to teach events is via EVLRN or EVLRNI, which include an event, an event-variable index, and an event-variable value, when teaching an event one also teaches an event-variable at the same time.

3.3 Module event capacity

Information about the quantity of events stored by a module can be found:

- Parameter 4 returns the maximum possible number of events that can be stored by a module.
- Parameter 5 the maximum number of event variables per event
- The EVLNF response from a module when sent an NNEVN request indicates the number of event slots remaining in the module.
- The NUMEV response from a RQEVN request is the number of events currently stored by the module.

Note that the value in EVLNF (remaining space) added to the value in NUMEV (used space) must be less than or equal to Parameter 4 (maximum number of spaces).

3.4 Reading Events

A configuration tool may read all the events from a module using the NERD message. The module responds with an ENRSP message for each event stored within the module.

The module shall send the ENRSP messages at a rate allowing the configuration tool to receive and handle these messages. The module shall transmit these messages with at least 10 ms between each message.

3.5 Reading Event Variables

A configuration tool may read all the event variables associated with an event from a module using a sequence of REQEV messages.

A request for EV index 0 returns all EVs via multiple EVANS messages.

An attempt to read an EV with an index beyond the number of EVs for the event will return a CMDERR(6), and a GRSP(6,REQEV,ev#,0).

3.6 Writing Event Variables

In order to write an EV the module must first be put into Learn mode using MODE(NN, Learn) or NNLRN message.

A configuration tool may then write an EV with the EVLRN message. The configuration tool must specify the event NN/EN, event variable index and event variable value.

The meaning of the event variable values is module dependent and must be documented by the module designer.

If an invalid event variable index is specified the module will return a CMDERR(6), and a GRSP(NN,6,EVLRN,0,0).

If an invalid event variable value is specified the module will return a CMDERR(11), and a GRSP(NN,11,EVLRN,0,0).

After a successful write of EV the module will respond with WRACK, and a GRSP(NN,OK,EVLRN,ev#,actual-value-written).

After writing the events and event variables the module should be reverted to Normal mode using the MODE(NN, Normal) or NNULN message.

3.7 Removing an Event

A configuration tool may remove an event from a module using the EVULN message. The module must be put into Learn mode using the NNLRN message prior to removing the event. The event to be removed is specified with the Node Number and Event Number within the EVULN message.

Node responds with GRSP(NN,OK,EVULN,0,0), or GRSP(NN,NAK,EVULN,0,0) if not found.

All event variables for the event are also removed.

After removing the events the module should be reverted to Normal mode using the NNULN message.

3.8 Removing an Event Variable

There is currently no dedicated way to remove a single EV. If a module is required to support EV removal as opposed to complete event removal then it should dedicate an EV value to indicate that the EV is not used.

It is recommended that if all EVs for the event are set to be unused then the event is also removed.

4 Additional Event Messages with Data

There is a set of Event-messages that carry additional data, these are:

- ACON1 / ACON2 / ACON3 Accessory ON-Long-event with 1 / 2 / 3 added bytes, respectively

- ACOF1 / ACOF2 / ACOF3 Accessory OFF-Long-event with 1 / 2 / 3 added bytes, respectively

- ASON1 / ASON2 / ASON3 Accessory ON-Short-event with 1 / 2 / 3 added bytes, respectively

- ASOF1 / ASOF2 / ASOF3 Accessory OFF-Short-event with 1 / 2 / 3 added bytes, respectively

5 Data messages

Note that ACDAT, RQDAT and ARDAT are not events and are not included within the Event services.

There is no specific change in the teaching process to teach events with data. Whether a module requires data or sends data with events is application specific.

6 Default Events

Modules may implement automatic configuration of events (default events) which can make a module easier to configure and use. Default events would be implemented as automatic configuration of EVs based upon a module's manufacturer's configuration, configuration of NVs or configuration of other events.

Default events behave in the same way as user configured events, it is only the way in which they come into existence which is different.

Default events must be reported by NERD and if the module supports the Teach service they must be able to be deleted or reconfigured.

7 Opcode Summary

Refer to the VLCB Opcode Specification document for details of the opcodes.

Request to Module	Module's Response	Use/meaning
NNLRN		Enter Learn mode. Deprecated, to be replaced by MODE. However, it is currently required for backward compatibility.
NNULN		Exit Learn mode, return to Normal mode. Deprecated, to be replaced by MODE. However, it is currently required for backward compatibility.
NNCLR	WRACK	Clear all events from a node.
	GRSP	ok
NNEVN	EVNLF	Read the number of event slots available in a node.
NERD	ENRSP	Read the events from a node
RQEVN	NUMEV	Request to read number of stored events
NENRD	GRSP	Error - Command not supported
	CMDERR	Error - Command not supported
EVULN	WRACK	Remove an event from a node
	GRSP	OK or Error
	CMDERR	Event not found
REVAL	GRSP	Error - Command not supported
	CMDERR	Error - Command not supported
REQEV	EVANS	Read event variable in learn mode
	GRSP	Error indication
	CMDERR	Error indication
EVLRN	WRACK	Teach an event in learn mode
	GRSP	Ok or error
	CMDERR	Error indication
EVLRNI	GRSP	Error - Command not supported

	CMDERR	Error - Command not supported

Note Opcodes EVLRNI, NENRD and REVAL are not supported and should return an error.

8 Service specific Mode Commands

The Event Teaching service introduces the Learn mode as described in section 4.1 Learn mode. These modes must specify the node number of the module and NN=0 will not be recognised.

Mode Command	Alias	Description	All module support when NN=0
8	Learn	Request to enter Learn mode	No
9	Unlearn	Request to leave Learn mode and return to Normal mode	Yes

9 Service Specific Status Codes

The following additional GRSP status codes are specified by the Event Teach service.

Code	Short Name	Comment
2	NOT_LRN	Not in Learn mode
4	TOO_MANY_EVENTS	Too many events provisioned in module
5	NO_EV	No Event-Variable
6	INV_EV_IDX	Invalid EV index specified
7	INVALID_EVENT	Invalid event specified
8	INV_EN_IDX	Invalid Event index
11	INV_EV_VALUE	Invalid EV value specified
13	OTHER_IN_LEARN	Another module in Learn mode

10 Service Specific Diagnostic Data

10.1 DiagnosticCodes

The following DiagnosticCodes for the Event services are supported:

0x01: return the number of events taught since power on.

10.2 Diagnostic Payload Data Return

The following RDGN diagnostic data numbers are specified by the Event services.

Service	Diagnostic Code	Diagnostic Byte1	Diagnostic Byte2	Comment
Teach	0x01	Count Hi	Count Lo	Count of the number of events taught since power on.

11 Service Specific Automatic Power-up Tests

No service specific power-up tests are specified by the Event Teach service.

12 Service Documentation

Modules implementing any of the Event Teach service must provide full documentation. In particular the following are required:

- Details of the EV usage.

13 Module Design Considerations

Modules should use EVs to configure event behaviour whereas NVs can be used to configure module operation and hardware usage.

Events and their EVs may have dependencies with each other and/or with NVs. The dependencies should follow the following rules where "earlier" refers to EVs with a lower index and "later" refers to EVs with a higher index.

- An event's EV's permitted values may depend upon the values of earlier EVs.
- The value of an event's EV should not depend upon the value of a later EV.
- The value of an event's EV may change when an earlier EV's value is changed.
- The value of an EV should not change when a later EV is changed.
- EV dependencies must be uni-directional so that if EVb depends on EVa then EVa must not depend upon EVb.
- Events and their EVs may be associated with NVs and those events may be automatically changed when the value of those NVs are changed.
- NVs must not depend upon EVs.

These rules allow management systems to write NVs in increasing NV order followed by the writing of events and EVs in increasing order to restore a module's configuration from backup.

14 Service Data

14.1 Parameters

The following parameters are associated with events and are to be provided.

Address	Param#	Name	Usage	VLCB should set these values
0x823	4	No Events	Max number of events available	Max number of events supported.
0x824	5	No EV per event	Max number of Evs per event	Max no Evs per event
0x828	8.5	Learn	Indicates if the module is in Learn mode.	Set if currently in learn mode.

14.2 ESD data bytes

Data1 = unused, set to 0

Data2 = unused, set to 0

Data3 = unused, set to 0

15 Glossary

EN	Event Number (0-65536, 0x0000-0xFFFF).
Event	Represented as NN:EN and when transmitted onto the VLCB bus indicates a change of state.
EV	Event Variable. Used to define the behaviour of a module associated with an event.
Event Identifier	A unique identifier for events on the network. Short events use a 16 bit event number (EN). Long events use a 32 bit identifier consisting of node number and event number.
Event Index	A temporary identifier used as a shorthand for NN:EN.
Event Number	A 16 bit value specifying the purpose of an event.
Mode	Indicates what set of messages a node will respond to and how it behaves to those messages.
NN	Node Number (1-65536, 0x0001-0xFFFF)
Teach	The process of associating an Event and its EVs with a module.