VLCB Versatile Layout Control Bus

Event Consumer Service Specification
Service# 6 Consumer

Version 1.1, February 2025, for Service version 2

Compatible with CBUS ® 4.0 Rev 8j

VLCB Event Consumer Service Specification

0.1 Table of Contents

0.2 Document History

Date	Changed by	Summary of changes	Service version
20th January 2023	Ian Hogg M.5144	Initial document	1
14 April 2023	Ian Hogg M.5144	Changed name to VLCB	1
10 Feb 2025	Martin Da Costa M.6223	Incorporated Event Acknowledgement	2
3 March 2025	Martin Da Costa M.6223	Corrected definition of specific mode handling	2

1 Introduction

This document describes the service related to Event Consumer service and its use within VLCB. This is an optional service, which may be added to the MNS.

Please refer to the Event Teaching Service documentation for a description of the VLCB Event model.

While this service is described as a separate service, practically it can be implemented as a single library combined with the other event services. Since the event services 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; it is practical to implement them together.

1.1 Dependencies on other services

The Consumer service depends upon the mandatory Minimum Node Service.

If a module is designed to only consume events then the Producer service does not need to be used by the module.

If a module is designed to handle only default events then the Event Teach service is not required.

2 Interrogating module event state

A module's event state may be interrogated through the following processes.

2.1 Event Status and Response Messages

A module shall be capable of returning the current ON/OFF state of a produced event using the AREQ or ASRQ request. The module shall respond with the ON/OFF state of the event by sending ARON/AROF or ARSON/ARSOF as appropriate.

A module may optionally send event request messages in order to receive and process ARON/AROF and ARSON/ARSOF messages to update their event states. Some modules may benefit from sending AREQ and ASRQ messages to elicit those report messages.

2.2 Start of Day

Consumer modules which are also event producers shall be able to be configured to consume and respond to an event designated for "Start of Day" (SoD). The module shall

respond to the SoD event with the ON/OFF state for all of its configured produced events, via ACON/OFF and ASON/OFF messages.

3 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

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

If a module has been taught an event and requires associated data and that module receives the event without data then it should send a GRSP error (of some kind)

If a module handles events with no data then it can optionally handle events with data in the same way. The module's documentation must explain the module's behaviour in this respect..

4 Processing of events

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

When a module is in Event Ack mode an Event Acknowledge message is sent in response to consuming an Event message (eg ACON, ASON, etc.). This is useful for monitoring and debugging node configuration.

5 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 an Event query and if the module supports the Teach service they must be able to be deleted or reconfigured.

Default consumed Long events would use the module's Node Number. Due consideration must be given when these events are created and what happens to these events if the module's Node Number is changed.

6 Power-on behaviour

Once modules have restored the state of inputs, outputs and events modules shall be able to respond to AREQ/ASRQ requests and SoD events to return the current state of their events.

7 Special Event Behaviour

It can be useful to use a normal short event to perform special actions across the layout. Therefore it should be possible to configure a consumed event for these actions. The exact EV configuration is not specified and is left to the module designer.

Unless these are defined to be default events the special event behaviour requires the Teach service to be implemented by the module.

7.1 Consumed Start of Day

As described in section 2.2 Start of Day a module with the Producer service must also support the consumption of an event which triggers a series of events to indicate the state of every event configured to be produced. The EV configuration used to configure the consumed SoD event should be taken into account when designing the EV usage.

7.2 Consumed End of Day

If events' status are saved in non-volatile memory the module designer should take care to reduce memory (EEPROM) wear. Wear leveling techniques must be considered.

An End of Day event technique can also be used to not write every event state change to the non-volatile memory but instead keep state in RAM and only perform a write to non-volatile memory just before power-down i.e. at end of day.

Modules should consider implementing the ability to produce/consume an EoD event to flush state from RAM to non-volatile storage.

The EV configuration used to configure the consumed EoD event should be taken into account when designing the EV usage.

A module must document whether it supports an End of Day event.

7.3 Consuming Own Events

If a module supports both production and consumption of events i.e. supports both the Producer service and the Consumer service then it may also be able to consume its own produced events. This functionality is controlled by the presence of the Consume Own Event service.

8 Opcode Summary

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

Opcode	Use by Consumer
ACON{1,2,3}	Consumed by a module to perform an ON action as defined by the EVs of the event NN:EN.
ACOF{1,2,3}	Consumed by a module to perform an OFF action as defined by the EVs of the event NN:EN.
ASON{1,2,3}	Consumed by a module to perform an ON action as defined by the EVs of the event EN. The Node Number in the event is ignored.
ASOF{1,2,3}	Consumed by a module to perform an OFF action as defined by the event's EVs of the event EN. The Node Number in the event is ignored.

Request to Module	Module's Response	Use/meaning
MODE	GRSP	Enter or Exit Event Acknowledge mode

9 Service specific Modes

Mode Command	Request
0x0A	Enables Event Acknowledge
0x0B	Disables Event Acknowledge

The MODE(nn, Event_Ack_ON) command is used to turn on a module's event acknowledgement functionality. MODE(nn, Event_Ack_OFF) command is used to disable the event acknowledgement.

Event acknowledgement is disabled by manufacturer defaults. Whether the Event Ack mode is enabled or disabled must be saved in non volatile memory and restored upon power up.

When a module is in Event Ack mode an Event Acknowledge message is sent in response to receiving an Event message (eg ACON, ASON, etc.).

ENACK(pNN, eventOpcode, NN, EN), where
  - pNN is the node's node number,
  - eventOpcode is the opcode of the event-message that triggered this response,
  - NN:EN the original event that triggered this response.

10 Service Specific Status Codes

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

Service	Code	Short Name	Comment
Consumer	255	EN_DATA	Event with Data error

11 Service Specific Diagnostic Data

11.1 Diagnostic Codes

0x01: return the number of events consumed and acted upon since power up.

11.2 Diagnostic Payload Data Return

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

Diagnostic Code	Diagnostic Byte1	Diagnostic Byte2	Comment
0x01	Count Hi	Count Lo	Count of number of events consumed since power on
0x02	Count Hi	Count Lo	Count of number of events acknowledged since power on

12 Service Specific Automatic Power-up Tests

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

13 Service Documentation

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

Documentation for any default Consumed default events.

Documentation must include information for Start of Day and whether End of Day is supported.

Details of the EV usage.

14 Service Data

14.1 Parameters

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

Param#	Name	Usage	VLCB should set these values
8.0	Consumer	Indicates if the module is able to be configured for consumed events.	Bit set if the Consumer service is used.

14.2 ESD data bytes

The ESD data bytes are used to indicate how EVs are used to encode the consumed events.

Data1 = EV usage

Data2 = Additional data

Data1 meaning	Data1 value	Data2 meaning	Data2 value
Not specified	0	Not used	0
EVs specify Actions	1	Number of EVs per Action	Action length in bytes
Slots	2		

Data3 = unused, set to 0