



# Event Producer Service Specification

## Service# 5 Producer

Version 1.1, March 2024, for Service version 1

Compatible with CBUS ® 4.0 Rev 8j

## VLCB Event Producer Service Specification

This work is licensed under the:

Creative Commons Attribution-ShareAlike 4.0 International License.

To view a copy of this license, visit:

<http://creativecommons.org/licenses/by-sa/4.0/>

or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

### **License summary:**

You are free to:

Share, copy and redistribute the material in any medium or format

Adapt, remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

**Attribution :** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

**ShareAlike :** If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

**No additional restrictions :** You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits

This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE

Software, Libraries and hardware modules using the VLCB protocols may have additional licence restrictions.

## 0.1 Table of Contents

0.1 Table of Contents	3
0.2 Document History	3
<b>1 Introduction</b>	<b>4</b>
1.1 Dependencies on other services	4
<b>2 Interrogating module event state</b>	<b>4</b>
2.1 Event Status and Response Messages	4
2.1.1 Producer Status Messages	4
2.2 Start of Day	5
<b>3 Additional Event Messages with Data</b>	<b>5</b>
<b>4 Processing of events</b>	<b>5</b>
<b>5 Default Events</b>	<b>6</b>
<b>6 Power-on behaviour</b>	<b>6</b>
<b>7 Special Event Behaviour</b>	<b>6</b>
7.1 Produced Start of Day	7
<b>8 Opcode Summary</b>	<b>7</b>
8.1 Event Producer and Consumer Services	7
<b>9 Service specific Modes</b>	<b>8</b>
<b>10 Service Specific Status Codes</b>	<b>8</b>
<b>11 Service Specific Diagnostic Data</b>	<b>8</b>
11.1 DiagnosticCodes	8
11.2 Diagnostic Payload Data Return	8
<b>12 Service Specific Automatic Power-up Tests</b>	<b>8</b>
<b>13 Service Documentation</b>	<b>8</b>
<b>14 Service Data</b>	<b>9</b>
14.1 Parameters	9
14.2 ESD data bytes	9

## 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
25February 2024	Martin Da Costa M.6223	Explicitly state that node number / event number combination to be unique when creating default events.	1



# 1 Introduction

This document describes the service related to Event Producer 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 Producer service depends upon the mandatory Minimum Node Service.

If a module is designed to only produce events then the Consumer 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 and events may be processed by the module's application code.

# 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 producer or consumer 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.

### 2.1.1 Producer Status Messages

In addition to nodes producing state-change Event-messages (ACON/OF and ASON/OF), other nodes can query the status of those events from the producer-node. This can be useful when the system is unsure of layout-state, and wants to query the producer of that state, without forcing an event-message to be sent. For example, the position of a point can be queried, without changing its position, by the following messages.

These messages come in triples for Long- and Short-Events:

- AREQ / ARON / AROF - Accessory Request / Accessory Reply On / Accessory Reply Off
- ASRQ / ARSON / ARSOF - Accessory Short Request / Accessory Short Reply On / Accessory Short Reply Off

- NB that short-events only use the 16-bit EN value, and the NN value is independent. Therefore, the ASRQ can be directed at a specific node, or if NN==0 will be replied to by all nodes producing that short-event. See *Opcode Specification*.

NOTE: These requests DO NOT produce ACON/OFF or ASON/OFF messages.

## 2.2 Start of Day

A Producer module shall be able to be configured to respond to an event designated for “Start of Day” (SoD).

The module shall respond with the ON/OFF state for all of its configured produced events, via ACON/OFF and ASON/OFF messages. The module designer should consider whether to base the response for each event on current hardware inputs or saved state.

As a result, the module may produce a large number of events, and therefore the module must ensure that these events do not cause stress on other modules on the bus. The pace of messages should be slow enough to allow other modules to react, and possibly produce events in turn, before the next event is sent. 10 ms should be sufficient.

## 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.

## 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.

## 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. The Node Number / Event Number combination of the default event must be unique in the events table.

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

Default events must be reported when querying a module's events. If the module supports the Teach service they must be able to be deleted or reconfigured.

## 6 Power-on behaviour

At module power-on, modules shall not produce any events associated with their inputs or outputs. Modules shall silently enter the Normal state.

A module may be configured to automatically produce a special power-up event which can be used to trigger the Start of Day process, see section [2.2 Start of Day](#). This automatically produced SoD event should be transmitted a few seconds after power-on to allow other modules to complete their power-on initialisation.

Modules which produce events based upon physical or logical inputs shall restore their produced event state upon power up based upon the current input state. Therefore no immediate produced events shall be sent due to change of input from state at power down.

Modules which produce events based upon physical or logical outputs e.g. servo endpoint events shall store the state of their outputs so that modules return their outputs to the state they were at power off. The state of their produced events shall be restored to the state at power down so that upon power up they do not immediately produce any events.

Once modules have restored the state of inputs, outputs and events modules shall be able to respond to AREQ/ASRQ requests and return the current state of their events without an event needing to be produced.

## 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 produced 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 Produced Start of Day

As described in section [6 Power-on behaviour](#) a module may have an optional capability to produce an event automatically after power-on. The EV configuration used to configure the produced SoD event should be taken into account when designing the EV usage.

## 8 Opcode Summary

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

Opcode	Use by Producer
ACON{1,2,3}	Sent by a module to indicate that something ON has happened within the module as defined by the event's EVs.
ACOF{1,2,3}	Sent by a module to indicate that something OFF has happened within the module as defined by the event's EVs.
ASON{1,2,3}	Sent by a module to indicate that something ON has happened within the module as defined by the event's EVs.
ASOF{1,2,3}	Sent by a module to indicate that something OFF has happened within the module as defined by the event's EVs.

Request to Module	Module's Response	Use/meaning
AREQ	ARON	Used to request the current event status of the produced long event NN:EN.
	AROF	Used to request the current event status of the produced long event NN:EN.
ASRQ	ARSON	Used to request the current event status of the produced short event EN.
	ARSOF	Used to request the current event status of the produced short event EN.

A module may receive a AREQ/ASRQ request after it has powered up but before it has actually sent the requested event. The module shall respond with the event status it would have sent as it entered the current state. This may be performed either by storing the event state in non volatile memory or by re-evaluating the event state using the status of the trigger of the event.

Note that a module shall handle AREQ and ASRQ with the following differences:

AREQ the provided NN:EN must match an event in the module's produced event table for the module to send a response.



ASRQ if the NN in the request is equal to zero then the NN is ignored and the module shall use its own Node Number to find a short event for EN in its event table. Note that in this scenario it is possible for multiple modules to respond to the request. The module shall include its own Node Number in the response.

ASRQ if the NN in the request is non zero then the provided NN must match the module's Node Number. The module shall use its own Node Number to find a short event for EN in its produced event table for the module to send a response. The module shall include its own Node Number in the response.

## 9 Service specific Modes

None.

## 10 Service Specific Status Codes

None.

## 11 Service Specific Diagnostic Data

### 11.1 DiagnosticCodes

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

### 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 produced since power on.

## 12 Service Specific Automatic Power-up Tests

No service specific power-up tests are specified by the Event Producer 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 Produced default events.

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
8.1	Producer	Indicates if the module is able to be configured for produced events	Bit set when the Producer service is used.

### 14.2 ESD data bytes

Data1 = unused, set to 0

Data2 = unused, set to 0

Data3 = unused, set to 0