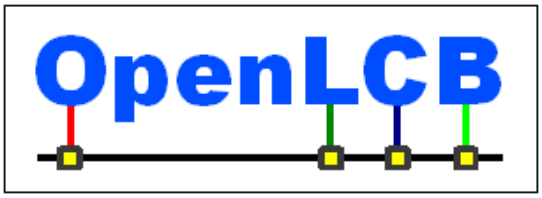

OpenLCB Standard	
Simple Node Information Protocol	
Apr 25, 2021	**Adopted**

1 Introduction (Informative)

This document defines the OpenLCB Simple Node Information Protocol, a light-weight method to request identification information from an OpenLCB node. The information returned is a few specific human-readable strings that allow the user to recognize the particular node, as well as supplying data to user interfaces to represent nodes.

2 Intended Use (Informative)

This Standard defines an optional protocol. Nodes may, but are not required to, implement this protocol.

The protocol does not depend on any transport layer protocol (such as Datagrams or Streams), thus the simplicity and low complexity makes it ideally suited to implement even in smaller nodes, such as Simple nodes.

A typical use of this protocol is for network browsers and configuration tools, which present the entire network with all its nodes to the user. When the user selects a node from a list, the human-readable strings as returned by this protocol could be used to represent the node instead of the 48-bit unique Node ID.

Nodes which have a user interface for selecting other nodes for interaction (e.g. a throttle selecting a train to operate) may also use the data supplied by this protocol to display to the user the selected or selectable nodes.

3 References and Context (Normative)

This is in the context of the following OpenLCB Standards:

- The OpenLCB Message Network Standard, which defines the basic messages and how they interact, as well as how the messages are represented on a CAN network, including how to transport long messages in multiple CAN frames. The Message Network Standard also specifies the Protocol Support Inquiry and Protocol Support Reply messages, which are referenced here.
- The OpenLCB Memory Configuration Protocol may be used alternatively to request overlapping or identical information from a node as provided by this standard. The OpenLCB Configuration Description Information (CDI) Standard defines how to interpret the configuration information. A specific option is ACDI, which, if present

gives access to the same information as the Simple Node Information Protocol via the Memory Configuration Protocol, but can also be used to change the user-provided information.

4 Message Formats (Normative)

4.1 Simple Node Information Request

Name	Simple Node	Dest ID	Event ID	Common MTI	Data Content
Simple Node Information Request	N	Y	N	0x0DE8	none

This message requests the destination node to send the Simple Node Information data to the source node.

4.2 Simple Node Information Reply

Name	Simple Node	Dest ID	Event ID	Common MTI	Data Content
Simple Node Information Reply	N	Y	N	0x0A08	payload bytes

This reply carries the Simple Node Information payload from the target node to the requesting node.

5 Payload Format (Normative)

5.1 Format

The reply payload is a sequence of bytes which is a concatenation of the following entries without any separator. The entries are of fixed or variable length.

- The byte 0x01 or 0x04 to indicate the version of the following content
- A null-terminated string for manufacturer name of no more than 41 bytes including terminating null.
- A null-terminated string for node model name of no more than 41 bytes including terminating null.
- A null-terminated string for node hardware version of no more than 21 bytes including terminating null.
- A null-terminated string for node software version of no more than 21 bytes including terminating null.

- The byte 0x01 or 0x02 to indicate the version of the following content
- A null-terminated string for user-provided node name of no more than 63 bytes including terminating null.
- A null-terminated string for user-provided node description of no more than 64 bytes including terminating null.

5.2 Versioning

Future versions of this standard may extend the number of null-terminated strings in each of the two sections. In such case the version number at the beginning of the sections will be set to the number of null-terminated strings presented in the particular section.

Nodes must therefore accept data with either of the version numbers as presented in Section 5.1.

6 Interactions (Normative)

6.1 Discovery

If a node supports the Simple Node Information Protocol, then the node must indicate so in the Protocol Support Reply message as defined by the OpenLCB Message Network Standard.

6.2 Request and reply

The requesting node sends a Simple Node Information Request message addressed to the destination node. The destination node should reply with one or more messages, each possibly being in one or more parts (frames), whose data content concatenated forms the payload, as defined in Section 5.1.

Nodes implementing the Simple Node Information Protocol should prefer sending the reply in one message in multiple frames instead of sending multiple messages. Nodes requesting must be able to process replies arriving in one or more messages, each having single or multiple frames.

Nodes requesting data via the Simple Node Information Request message must wait for the entire reply to arrive before sending another Simple Node Information Request to the same destination node.

6.3 Stability

The information presented in the Simple Node Information Reply has to be constant. If the information changes, the node must enter Initialized state again and send out a Node Initialization Complete message (as defined by the OpenLCB Message Network Standard) to inform nodes that the information they may have cached needs to be refreshed.

7 CAN Adaptation (Normative)

There are no specific provisions for CAN transport. The messages are formatted on CAN transport as defined for generic addressed messages in the OpenLCB Message Network Standard. In particular, refer to the encoding of the nibble 'f' in the first data byte on CAN representing how individual frames of longer messages have to be encoded.

Table of Contents