



OpenLCB Standard	
Configuration Description Information	
May 4, 2013	Preliminary

## 1 Introduction (Informative)

Minimal introductory material, only the stuff that's absolutely needed to understand the Standard.

## 2 Intended Use (Informative)

Any limitations to the area of use of the Standard.

## 5 3 References and Context (Normative)

Citations to other docs, as needed.

For more information on format and presentation, see:

- OpenLCB Common Information Technical Note

- 10 The CDI information shall be constant. A node may not change it after any part of it has been retrieved and before the next transition of the node away from the Initialized state. (The transition back to Initialized state tells other nodes to flush their caches and pick up any changed content)

## 4 Stuff to be merged into the above (or the TN)

15

The first byte is used to distinguish the coding.

“<” (which is part of the XML “<?xml version='1.0'?>” definition): Uncoded characters

UTF BOM: Various UniCode forms

- 20 0x01 – tag for compressed. One format defined for total start string of 0x0101. (If you want to use another format, decompress on board)

(can't use 0x00 as lead, since that's the end-of-string indication)

The schema at prototypes/xml/schema (better location?) is really the normative thing, because that's what we check.

- 25 But we add the <acdi> element to it, so the document isn't really complete; “Other protocols may add, but not remove, elements and attributes”? Remove constraints? How does extensibility work here?

- 30 Numbers for segments, offsets, etc must be specified as decimal numbers. Hexadecimal notation (0x1234) is not permitted. Although it may have made the CDI easier for people to read, the XML is primarily intended for consumption by OpenLCB nodes. Providing optional coding, such as hexadecimal numbers, makes those nodes more complex.

- 35 The 'mask' attribute is for use with legacy equipment only. New OpenLCB implementations should not use it. Instead, they should lay out their configuration variables are processed as if they each had their own addressable location. (How they are actually stored in memory is not specified.)

Nodes that use the 'mask' attribute in their CDI must properly implement the write-under-mask operation in their memory configuration protocol support.

40

## Table of Contents

Title.....	1
Section Title.....	2
Section Title.....	2