

OpenLR™

White Paper

Version: 1.4



An open standard for encoding, transmitting and decoding location references in digital maps

Copyright © 2009-2011 TomTom International B.V.

OpenLR™ is a trade mark of TomTom International B.V.

This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.





License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
- b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.
- c. **"Distribute"** means to make available to the public the original and copies of the Work through sale or other transfer of ownership.
- d. **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
- e. **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
- f. **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed



by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

- g. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
- h. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
- i. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections; and,
- b. to Distribute and Publicly Perform the Work including as incorporated in Collections.
- c. For the avoidance of doubt:
 - i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
 - ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
 - iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Adaptations. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to



exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested.

- b. If You Distribute, or Publicly Perform the Work or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
- c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination



- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
- e. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.



Version information

Version 1.0

Changes:

- Initial version

Author: TomTom International B.V

Date: 2009-09-08

Version 1.1

Changes:

- Errata for v1.0 included (sections 4.4.4, 4.6.2, and 9.1)
- Note added for physical format version number (section 4.5.1)
- Section 3.4, Rule-4: U-turns nodes are also valid nodes; invalid nodes might be chosen exceptionally if no other possibility exists
- Encoding steps, Step 1: turn restrictions are optional (section 7.1)
- Encoding steps, Step 9: explanation how to deal with lines longer than 15km (section 7.1)
- Decoding steps, Step 3: find candidate lines directly if no candidate nodes can be found (section 8.1)
- Decoding steps, Step 4: LRP attributes may vary between encoder and decoder map so that values don't need to match exactly (section 8.1)
- Decoding steps, Step 5: explanation how to deal with a situation where start and end line are equal; retrying other candidate line pairs if no valid shortest-path can be determined (section 8.1)
- Explanation added for handling of prohibited turns in decoding example (section 9.2)

Author: TomTom International B.V

Date: 2009-12-11

Version 1.2

Changes:

- Description of the XML format for OpenLR™ location references (new section 7)
- Schema definition of the XML format (new section 8)
- Old sections 7 – 10 have moved to sections 9 – 12
- Encoding and decoding example extended with the XML format (section 11)
- Encoding process (section 9) and decoding process (section 10) now handle (abstract) physical formats instead of the concrete binary format.

Author: TomTom International B.V.

Date: 2010-02-01

Version 1.3

Changes:

- Update explanation of common terms and abbreviations
- Section 2: Introduction of concepts has been updated to the point location support
- Section 3.3: Updated sub-section explaining the offsets
- Section 3.5: Data format Rule-3 has been updated
- Section 3.5: New data format Rule-5
- Section 4: New section for a high-level example of line locations
- Section 5: New section for a high-level example of point locations
- Section 6.4.7: New data type for point locations - orientation
- Section 6.4.8: New data type for point locations - side of road
- Section 6.5.1: Actual version of the binary format updated
- Section 6.5.4: New flag - point location
- Section 6.6.2: New format for offsets (percentage representation)
- Section 7: Update of the binary physical format
- Section 7.1: Simpler overview of the data format for line locations
- Section 7.2: Explanation of the different point location types
- Section 7.3: Characteristics for determining the location type



- Section 7.4: Changelist from version 2 to version 3 (binary format)
- Section 8.6: New data elements for point locations in the XML format
- Section 8.8: New data types supporting point locations in the XML format
- Section 9: Updated schema definition for the XML format
- Section 10: Update of the encoding steps to support point locations
- Section 10.3: New section explaining the offset handling during the encoding phase
- Section 11: Update of the decoding steps to support point locations
- Section 11.3: New section explaining the offset handling during the decoding phase
- Section 12.1: Update of the example for line location encoding and decoding
- Section 12.2: New section showing a small encoding and decoding example for all point location types

Author: TomTom International B.V.
Date: 2010-05-17

Version 1.4

Changes:

- Fix bearing value of the last location reference point in PointAlongLine and PoiWithAccessPoint example
- Document has been split into several main chapters (A – I) following a logical model of chapters providing mandatory information on the OpenLR™ standard, chapters providing recommendations (optional for the OpenLR™ standard) and other chapters for the introduction, examples and appendices.
- Introduction explains the structure of the document and also highlights the chapters covering the OpenLR™ standard.
- Location type section has moved to the concepts chapter.
- Location and map requirements have moved to an own chapter.
- The data types used in the logical data format are defined.
- The logical data format explains the "SideOfRoad" and "Orientation" information.
- The data format rules have been revised. Rule - 4 has been relaxed and has been extended by a recommendation.
- Problems with the relative coordinates in the binary data format have been identified and explained.
- A recommendation has been added to the offsets in the binary data format. When offsets in version 3 are used, then the encoder might add an additional intermediate LR-Point when high accuracy is required.
- Explanations regarding the encoder implementation have been revised. Some possible improvements have been added.
- Explanations regarding the decoder implementation have been revised. The execution of a direct line search has been further explained.

Author: TomTom International BV
Date: 2010-09-14



Content

| | |
|---|----|
| Chapter A Introduction | 11 |
| 1. Introduction | 12 |
| 1.1 Common terms and abbreviations | 14 |
| 2. Concepts | 16 |
| 3. Location types | 17 |
| 3.1 Line locations | 17 |
| 3.2 Point locations | 18 |
| 3.2.1 Geo-Coordinate | 18 |
| 3.2.2 Point along line | 19 |
| 3.2.3 POI (point of interest) with access point (on a line) | 19 |
| Chapter B Requirements | 21 |
| 4. Requirements | 22 |
| 4.1 Location properties | 22 |
| 4.1.1 Line location | 22 |
| 4.1.2 Point locations | 22 |
| 4.2 Map requirements | 22 |
| Chapter C Logical data format | 24 |
| 5. OpenLR™ - Logical Data format specification | 25 |
| 5.1 Data types | 25 |
| 5.2 Location Reference Point | 25 |
| 5.2.1 Topological connection of LR-points | 25 |
| 5.3 Components of LR-points | 26 |
| 5.3.1 Coordinate pair | 26 |
| 5.3.2 Functional Road Class | 26 |
| 5.3.3 Form of way | 26 |
| 5.3.4 Bearing | 27 |
| 5.3.5 Distance to next LR-point | 28 |
| 5.3.6 Lowest FRC to next LR-point | 28 |
| 5.3.7 Side of road | 28 |
| 5.3.8 Orientation | 29 |
| 5.4 Offsets | 29 |
| 5.4.1 Positive offset | 29 |
| 5.4.2 Negative offset | 29 |
| 5.5 Relationship Attributes – LR-point | 30 |
| 6. Data format rules | 31 |
| 6.1 Overview of the data format rules | 32 |
| Chapter D Physical data format (Binary) | 33 |
| 7. OpenLR™ - Binary representation | 34 |
| 7.1 Data types | 34 |
| 7.2 Coordinates (COORD) | 34 |
| 7.2.1 Absolute format | 35 |
| 7.2.2 Relative format | 35 |
| 7.3 Attribute values | 36 |
| 7.3.1 Functional Road Class (FRC) | 37 |
| 7.3.2 Form of way (FOW) | 37 |
| 7.3.3 Bearing (BEAR) | 37 |
| 7.3.4 Distance to next LR-point (DNP) | 38 |
| 7.3.5 Lowest FRC to next point (LFRCNP) | 38 |
| 7.3.6 Orientation information | 39 |
| 7.3.7 Side of road information | 39 |
| 7.4 Location Reference header | 39 |
| 7.4.1 Version (VER) | 39 |
| 7.4.2 Attribute flag (AF) | 40 |



| | | |
|--|---|----|
| 7.4.3. | Area flag (ArF) | 40 |
| 7.4.4. | Point flag (PF) | 40 |
| 7.5 | Offsets | 41 |
| 7.5.1. | Offset flags | 41 |
| 7.5.2. | Offset values | 41 |
| 7.5.2.1 | Equal distribution (Format version 2) | 41 |
| 7.5.2.2 | Percentage (Format version 3) | 42 |
| 8. | OpenLR™ - Physical data format binary specification | 44 |
| 8.1 | Line location | 44 |
| 8.1.1. | Line Location with 2 location reference points | 44 |
| 8.1.2. | Line location with n location reference points (n > 2) | 44 |
| 8.1.3. | Code size calculation | 45 |
| 8.2 | Point locations | 46 |
| 8.2.1. | Geo-coordinate | 46 |
| 8.2.2. | Point along line | 46 |
| 8.2.3. | Network node | 47 |
| 8.2.4. | Point of interest (POI) with access point | 47 |
| 8.2.5. | Point of interest (POI) with access point at network node | 48 |
| 8.3 | Identification of the location type | 48 |
| 8.4 | Changes | 49 |
| 8.4.1. | Version 2 -> Version 3 | 49 |
| Chapter E Physical data format (XML) | | 50 |
| 9. | OpenLR™ - XML representation | 51 |
| 9.1 | Data types | 51 |
| 9.2 | Coordinates | 51 |
| 9.3 | Line attribute values | 51 |
| 9.3.1. | Functional Road Class (FRC) | 52 |
| 9.3.2. | Form of way (FOW) | 52 |
| 9.3.3. | Bearing (BEAR) | 52 |
| 9.4 | Path attribute values | 52 |
| 9.4.1. | Distance to next LR-point (DNP) | 53 |
| 9.4.2. | Lowest FRC to next point (LFRCNP) | 53 |
| 9.5 | Point attribute values | 53 |
| 9.5.1. | Side of road | 53 |
| 9.5.2. | Orientation | 53 |
| 9.6 | Line location reference | 54 |
| 9.6.1. | Location reference point | 54 |
| 9.6.2. | Last location reference point | 54 |
| 9.6.3. | Offsets | 54 |
| 9.7 | Point location reference | 54 |
| 9.7.1. | GeoCoordinate | 55 |
| 9.7.2. | PointAlongLine | 55 |
| 9.7.3. | PoiWithAccessPoint | 55 |
| 9.8 | XML location reference | 55 |
| 9.9 | Binary location references | 56 |
| 9.9.1. | Binary location reference | 56 |
| 9.10 | Additional information | 56 |
| 9.10.1. | Bounding box | 56 |
| 9.10.2. | Map database | 57 |
| 9.10.3. | Additional data | 57 |
| 9.11 | Root tag: OpenLR | 57 |
| 9.11.1. | Location ID | 57 |
| 10. | OpenLR™ - Physical data format XML specification | 58 |
| Chapter F Encoder | | 65 |
| 11. | Encoding locations | 66 |
| 11.1 | Encoding steps | 66 |
| 11.2 | Overview of the encoding steps | 69 |
| 11.3 | Composition of offset values | 73 |



| | |
|--|-----|
| Chapter G Decoder | 74 |
| 12. Decoding location references | 75 |
| 12.1 Decoding steps..... | 75 |
| 12.2 Overview of the decoding steps | 77 |
| 12.3 Decoding offsets..... | 79 |
| Chapter H Examples | 80 |
| 13. OpenLR™ example | 81 |
| 13.1 Line Location | 81 |
| 13.1.1. Example setup | 81 |
| 13.1.2. Encoding..... | 82 |
| 13.1.3. Binary location reference..... | 83 |
| 13.1.4. XML location reference..... | 85 |
| 13.1.5. Decoding..... | 87 |
| 13.2 Point location..... | 90 |
| 13.2.1. Example setup..... | 90 |
| 13.2.2. GeoCoordinate | 91 |
| 13.2.2.1 Encoding | 92 |
| 13.2.2.2 Decoding..... | 92 |
| 13.2.3. PointAlongLine..... | 93 |
| 13.2.3.1 Encoding | 93 |
| 13.2.3.2 Decoding | 96 |
| 13.2.4. PoiWithAccessPoint | 97 |
| 13.2.4.1 Encoding | 97 |
| 13.2.4.2 Decoding | 100 |
| Chapter I Appendix | 102 |
| 14. Appendix..... | 103 |



Chapter A

Introduction



1. Introduction

Communication of spatial information involves the communication of location. The communication chain of a machine readable location can be described as encoding the location at the sender side, transfer of the code to the receiving system and decoding the code at the receiver side. The process of encoding a location is also called Location Referencing. This White Paper assumes a map on the sender side from which the location is encoded and a map on the receiver side in which the decoded location is found back. An obvious way of Location Referencing is using geographic coordinates. One important disadvantage of using coordinates is that it assumes identical maps at both sides of the communication chain which often is not the case. As a consequence, the decoded location may not be found back in the receiver map, or decoding (e.g. map-matching) may be inaccurate or ambiguous. OpenLR™ is a method for location referencing which does not have this disadvantage. It accommodates requirements of communication of location between systems which have dissimilar maps. OpenLR™ is communication channel independent. It takes bandwidth requirements into account in the sense that OpenLR™ requires minimal bandwidth.

OpenLR™ has been designed for the use case of transferring traffic information from a centre to in-vehicle systems, built-in or used as an add-on (PND, Smart Phone). The information transferred can consist of the current traffic situation at a certain location, a traffic forecast or special alerts. The corresponding locations are roads, a list of connected roads or points of interest.

The most well-known and most used method to transfer traffic information today is called RDS-TMC. The Location Referencing used in RDS-TMC makes use of pre-coded locations. These pre-coded locations are added to the corresponding locations in the map by the map providers of the sending and the receiving map. The process of encoding is looking up the location code in the map belonging to the relevant location. The process of decoding is finding back the location code in the map and looking up the corresponding location.

From the fact that RDS-TMC makes use of pre-coded location it follows that the amount of locations fit to be transferred is limited. OpenLR™ does not have that restriction. With OpenLR™ every location in a map can be transferred.

The technical goal of OpenLR™ is to specify a location referencing system with the following objectives:

- *dynamic generation of a location reference, including urban and low level roads*
- *compact format for transmission*
- *handling map differences caused by different map vendors or versions*

The industry goal of OpenLR™ is wide-scale adoption by the industry at large. OpenLR™ is therefore proposed as an open standard in an Open Source framework. It shall be usable for anyone dealing with locations and transmitting these between systems having dissimilar maps. The current version of OpenLR™ focuses on line locations and point locations but can easily be extended to handle area locations.

While the standard is developed and maintained by TomTom International B.V., everyone is invited to contribute to its further development.

This document defines the OpenLR™ standard. The Chapter A gives an introduction to the concepts of OpenLR™. Chapter B explains the required location and map properties. Chapter C defines the logical data format for location references and Chapter D and Chapter E define the physical formats for OpenLR™ location references.

The Chapter F provides recommendations for the implementation of an OpenLR™ encoder and Chapter G provides recommendations for the implementation of an OpenLR™ decoder. Examples



for the different location types are explained in Chapter H and the appendix can be found in Chapter I.

The following chapters define the OpenLR™ standard:

- Chapter B
Requirements
- Chapter C
Logical data format
- Chapter D
Physical data format - Binary
- Chapter E
Physical data format - XML

Additional documents

The OpenLR™ team provides the following additional documents for further understanding the OpenLR™ method. These documents explain certain topics in more detail than it can be done in this document. However, these documents are not part of the OpenLR™ standard!

- Intermediate location reference points
 - Version: 0.1
 - Description: This document describes the handling of intermediate location reference points and also the handling of loops within the location.

1.1 Common terms and abbreviations

The following tables explain common terms and abbreviations used in this document and in the context of location referencing

| Term | Description |
|---------------------------------------|--|
| Bit | A bit is a binary digit, taking a logical value of either "1" or "0" (also referred to as "true" or "false" respectively). |
| Byte | A byte is a basic unit of measurement of information storage in computer science. There is no standard but a byte most often consists of eight bits. |
| Candidate line | A line object in the network of a decoder which might be referenced by the Location Code (as generated by the encoder). |
| Candidate node | A node object in the network of a decoder which might be referenced by the Location code (as generated by the encoder). |
| Decoder | A software component which decodes a Location Code and finds the corresponding location back in a map. |
| Encoder | A software component which generates a Location code for a location in a map. |
| Intermediate | An internal Location Reference Point (LRP); not the start LRP nor the end LRP (destination). |
| Intermediate location reference point | see "Intermediate" |
| Line | A one-dimensional representation of a road or part of road in a road network. A line starts and ends at a node. It is directed, this means two-way traffic flow is represented by two (directed) lines, one per direction. |
| Location | A specification of the position on the earth surface of an object in a digital map. |
| Location reference | Location Code, created according to a specific set of rules, used to reference a location. |
| Location reference path | A route in a road network in a digital map which is referenced by the location reference. This path might be longer than the location itself. |
| Location Reference Point (LRP) | A description of an object in a digital map which consists of coordinates and additional information about a line in the map. |
| Map | A Map is the geospatial representation of an area on the earth surface. |
| Node | A zero-dimensional object in the road network. A node acts as start and end for lines. |
| Offset | The specification of a position along a path to indicate the start or the end of a location. |
| Point | A zero-dimensional element that specifies a geometric location. One coordinate pair specifies the location. |
| Route | A route is a collection of line objects in a digital map connecting a departure location and a destination location, defined according to certain criteria which might include time distance or cost. |
| TMC | Traffic Message Channel; public or commercial traffic information broadcast channel using RDS (Radio-Data-System). |
| XML | Extensible Markup Language |

Table 1: Explanation of common terms



| Abbreviation | Description |
|--------------|---|
| AF | Attribute Flag – a flag which indicates that the binary representation of the location reference includes attribute information |
| ArF | Area Flag – a flag which indicates that the location reference describes an area |
| BEAR | Bearing – angle between the direction to a point in the network and a reference direction (here: the true North) |
| COORD | Coordinates – a pair of two values (longitude and latitude) representing a position in a two-dimensional network |
| DNP | Distance to Next Point – the length in meter to the next location reference point (measured along the location reference path between these two LRP) |
| FOW | Form Of Way - Certain aspects of the physical form that a line takes. It is based on a number of certain physical and traffic properties. |
| FRC | Functional Road Class - A classification based on the importance of the role that the line performs in the connectivity of the total road network. |
| lat | Latitude – geographic coordinate used for north-south measurement |
| LFRCNP | Lowest Functional Road Class to Next Point |
| lon | Longitude – geographic coordinate used for east-west measurement |
| LRP | Location Reference Point – a point of the location which holds relevant information for a map-independent location reference |
| NOFF | Negative Offset – distance in meter along the location reference path between the real end of the location and the end of the location reference path |
| NOffF | Negative Offset Flag – a flag which indicates that a negative offset is included in the location reference |
| ORI | Orientation – determines where a point is located with respect to the driving direction of a line |
| POFF | Positive Offset – distance in meter along the location reference path between the start of the location reference path and the real start of the location |
| POffF | Positive Offset Flag – a flag which indicates that a negative offset is included in the location reference |
| POI | Point of interest |
| RFU | Reserved for future use – a bit in a binary stream which does not have a use yet |
| SOR | Side of road – determines where a point is located with respect to a line |
| TMC | Traffic Message Channel - is a technology for delivering traffic and travel information to drivers using pre-defined points and segments in a network |
| VER | Version – Version information |

Table 2: Explanation of common abbreviations



2. Concepts

OpenLR™ describes a method and a format for encoding, transmitting and decoding (map-independent) references of locations. Locations are objects in a digital map, like points, paths and areas. The method makes it possible to encode a location in a map, send it to a system having another (possibly different) map and find the location back in this receiving map. Provided that both, encoder map and decoder map are meeting “navigable map” standards (in terms of accuracy and content), the encoder does not need to know about the decoder map and the decoder also does not have to care about the map used for encoding the location. The format to transmit such location reference is compact so that it can be used in systems having bandwidth restrictions.

The current OpenLR™ standard focuses on line locations (paths) and point locations (e.g. POIs) in a digital map. OpenLR™ can be used for transmitting locations between different maps and one use case is the distribution of traffic information from a traffic information centre to several customers, like car drivers.

The main idea of OpenLR™ is to describe a location with a concatenation of (several) shortest-paths. The concatenation of such shortest-paths shall cover the location completely. Each shortest-path is specified by information about its start and its end. This information is combined in so called location reference points (LRPs). The LRPs are ordered from the start of the location to the end of the location and the shortest-path between two subsequent LRPs covers a part of the location. The concatenation of all these shortest-paths will cover the location completely and this path is called the location reference path. The location reference path may be longer than the original location and offsets will trim this path down to the size of the location path. Offsets are also used to define a location on a line more precisely (e.g. point locations along a line) than using the start and end node of that line.

Two location reference points, describing the start and end of the location reference path, are mandatory components for any OpenLR™ location reference, except simple geo-coordinates. Between these two LRPs may exist several additional location reference points (intermediates). These intermediates are added to the list of LRPs if the location can only be covered by a composition of shortest-paths.

The role of the encoder is to determine the number and the positions of the location reference points to describe the location uniquely. The role of the decoder is to resolve the received location reference points to a location in its own map.

The format for transmitting location references is also described in the OpenLR™ standard. A binary format is compact and can be used for the transmission of location reference data where the data size is important to be small. An XML format can be used if the need of having a compact format is less important and the existing environment for data transmission may already be based on XML.

The data used for a location reference point relies on map data common for navigable digital road maps. This common data includes a geographic position and line attributes like “functional road class” and “form of way”.

The location properties and the map requirements are outlined in the following sections. The OpenLR™ standard is developed and maintained by TomTom International B.V. but it is intended as an open standard that can easily be used by other parties. Any contributions to the further development are welcome and can be discussed and submitted via the OpenLR™ website <http://www.openlr.org>.

The OpenLR™ standard currently supports the location types: “Line”, “Geo-Coordinate”, “PointAlongLine”, and “PoiWithAccessPoint”. All location types are explained in section 3.

3. Location types

OpenLR™ currently supports four different types of locations and it may support other location types in the future. The following list gives an overview of the supported types and also provides a link to the section where the specific type is explained a bit more in detail.

- Line (see section 3.1)
- Geo-Coordinate (see section 3.2.1)
- PointAlongLine (see section 3.2.2)
- PoiWithAccessPoint (see section 3.2.3)

3.1 Line locations

A line location is a one-dimensional part of a road or a road network. Line locations consists of at least one line and if more than one line is used to describe the location then these lines need to form a connected path in the road network. Offsets can be used to identify locations which do not start or end exactly at a network node. Examples of line locations are jams, (temporary) speed limits, (calculated) routes. The following figures (see Figure 1 and Figure 2) show different types of line locations where the location is marked in red and the position of the offsets are shown as green dots.

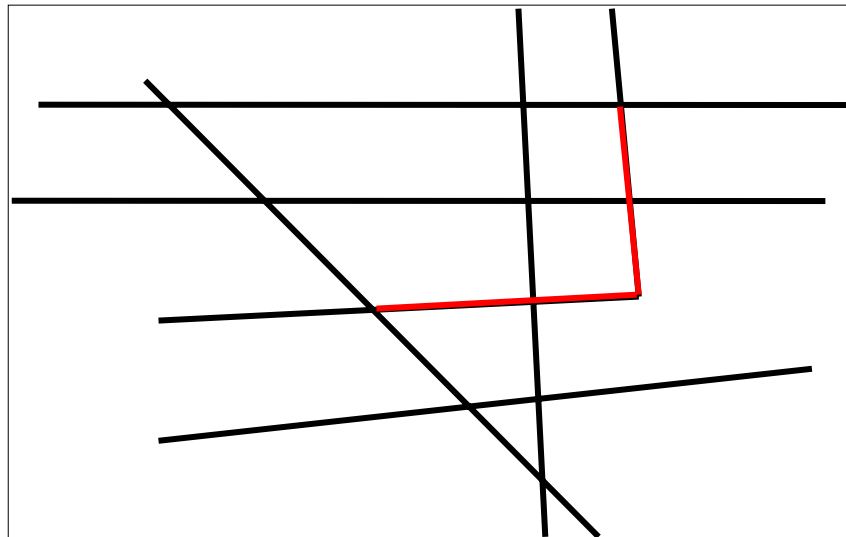


Figure 1: Line location without offsets

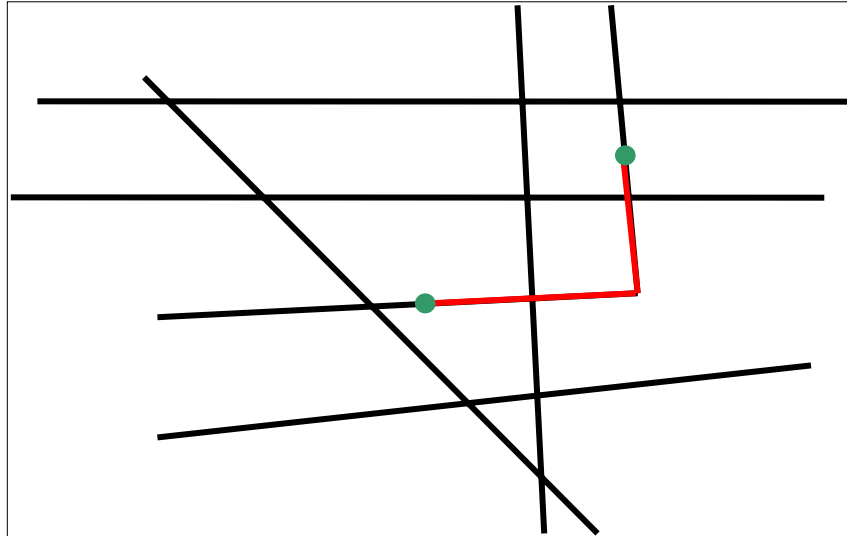


Figure 2: Line location with offsets

3.2 Point locations

A point location is a zero-dimensional element in a map that specifies a geometric location. At least one coordinate pair specifies the point location. The following sections outline different point location types when seen in combination with a (road) network and their real-world examples. In the following parts several point location types will be explained. The types differ in how the coordinate pair is related to the (road) network.

3.2.1. Geo-Coordinate

A geo-coordinate pair is a position in a map defined by its longitude and latitude coordinate values. This type of point location may or may not be bound to the network and can be everywhere on the surface. Figure 3 shows an example of such a point location (red point).

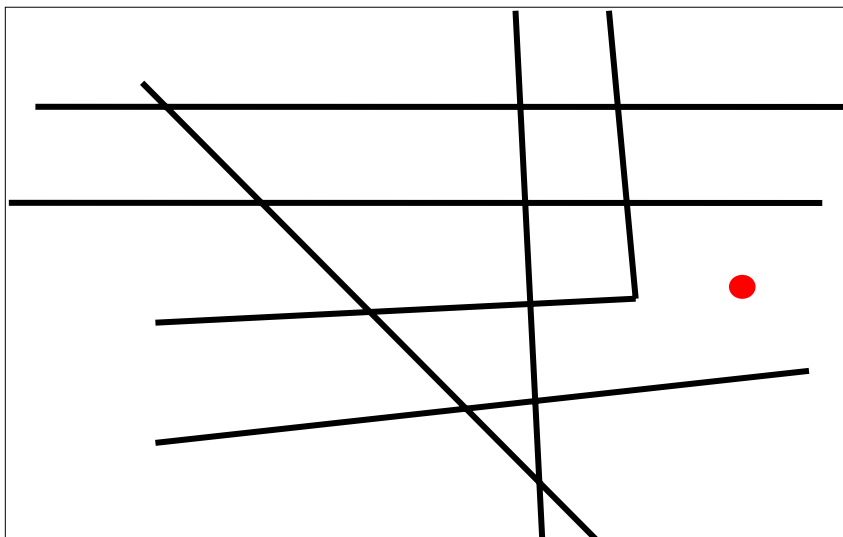


Figure 3: Example - Geo-coordinate

Real world examples for a geo-coordinate as a point location are all coordinate pairs on the surface. This is the general type of a point location. All other types can also be expressed by only using the geo-coordinate pair.

3.2.2. Point along line

The next point location type is a location along a line. Such a line is bounded by two nodes. The point location is dependent on the road network and Figure 4 shows such an example (red point). The point can be on the right side of the line, on the left side of the line, on both sides of the line, or directly on the line. Additionally the point may have an orientation to indicate in which direction of the line the information referenced at that point is useful.

Real world examples of this point location type are all points of interests like petrol stations, shopping malls and restaurants and also house numbers and address points. But it can also be used to reference to speed cams or induction loops.

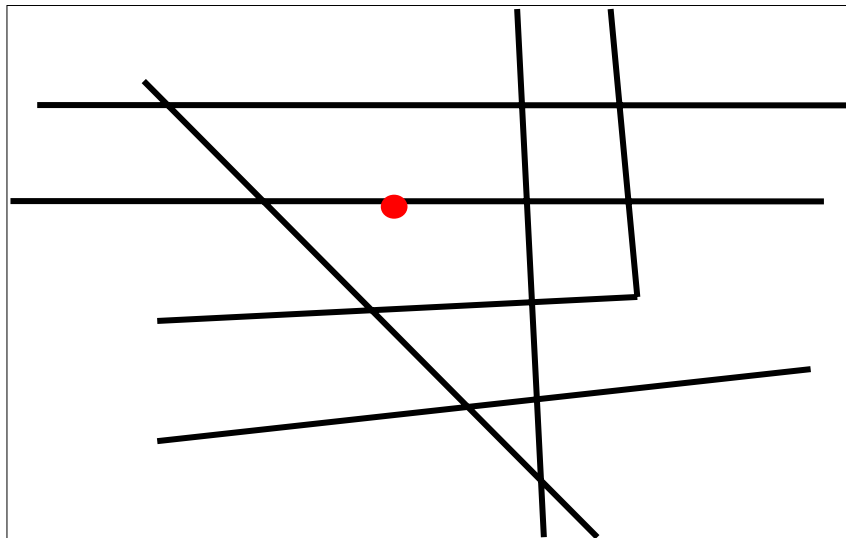


Figure 4: Example - Point along line

3.2.3. POI (point of interest) with access point (on a line)

Another point location type is a simple point location (geo-coordinate) in combination with a so called “access point” which is part of a line. The access point can be on the right side of the line, on the left side of the line, on both sides of the line, or directly on the line. Additionally the point may have an orientation to indicate in which direction of the line the information referenced at that point is useful.

Figure 5 shows an example of this point location type. The red point indicates the point location to be referenced. In combination with this point location there is an access point (green point). The access point is the point within the network in order to access the red point location. An application can use the green point to navigate the user to the desired point location.

Examples for such point locations can be address points but also all point of interests (POI) not being closely related to the road network. One might also think of access to petrol stations or parking garages.

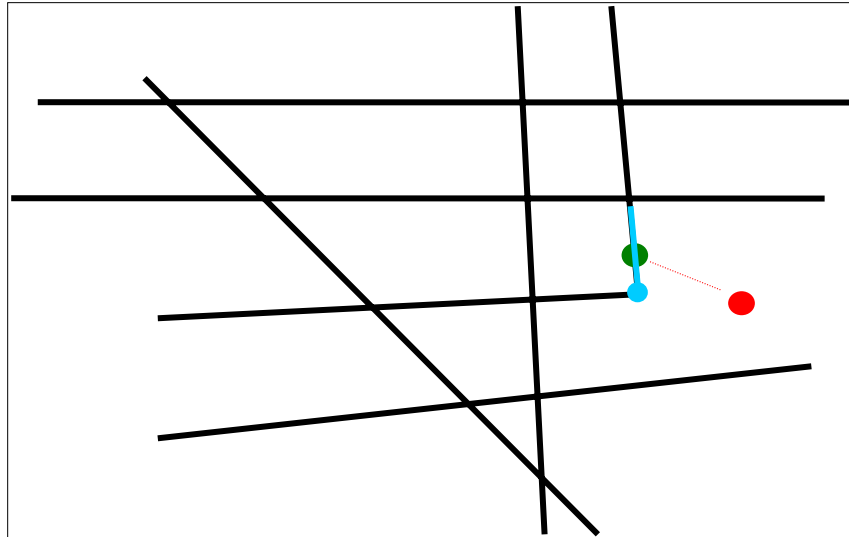


Figure 5: Example - Poi with access point



Chapter B

Requirements



4. Requirements

The OpenLR™ standard focuses on creating map independent references for line and point locations. Line locations are two-dimensional objects in a network that normally consist of a list of line elements. Point locations are zero-dimensional elements that specify a geometric location.

The digital map used for encoding and decoding and the location itself must fulfill some requirements in order to guarantee acceptable results for the OpenLR™ method.

4.1 Location properties

4.1.1. Line location

Line locations must fulfill the following requirements:

- A line location must be connected.
 - Two subsequent lines in the location must also be connected and adjacent in the underlying network
 - If a driving direction is available then the location shall be traversable from its start to its end
- A line location is represented by an ordered list of line elements.
 - The list of line elements needs to be ordered from the start of a location to the end of a location

Line locations which do not fulfill all requirements cannot be encoded using OpenLR™.

4.1.2. Point locations

Point locations must fulfill the following requirements:

- Geo coordinates must be represented in the WGS84 format

Point locations which do not fulfill all requirements cannot be encoded using OpenLR™.

4.2 Map requirements

A digital map consists of nodes and lines. Lines represent real world roads or parts of a road. Nodes represent crossings of roads or parts of roads. The encoder and decoder map might differ but nevertheless the OpenLR™ standard provides a method to reference to the same location represented in both maps.

For being able to generate a map-independent location reference and also being able to resolve locations properly a map should contain information about the following data:

- Functional road class (FRC)
 - Every line in the network should have a functional road class value indicating its importance in the network.
- Form of way (FOW)
 - Every line in the network should have a form of way value indicating its physical properties.
- Geometry
 - Every line should know about its real geometry in the real world.
 - Lines shall not be abstracted by the airline.
- Coordinates in WGS84
 - Every node in the network should have coordinates in the WGS84 format.
 - The preferable accuracy is decamicrodegrees for each value.
- Length in meter



- Every line should have a length value in meter indicating its real dimension along the geometry.

There needs to be a proper mapping between the “functional road class” values in the digital map and the FRC values covering the range of navigable roads from highest to lowest importance, as defined in the logical data format (see section 5.3.2). If no such functional road class values are provided by the digital map data then the FRC values should be derived from other information available in the digital map.

There needs to be a proper mapping between the form of way values in the digital map and the FOW values, as defined in the logical data format (see section 5.3.3). If no such “form of way” values are provided by the digital map data then the FOW values should be derived from other information available in the digital map.

Note:

If a map provides less information than required the encoding and decoding does generally still succeed but the error rate may increase considerably. At least two out of the following three information types should be available in the map data: functional road class, form of way, geometry (for the calculation of bearing values [angle to the true North]). Coordinates in WGS84 and length values in meters are mandatory.



Chapter C

Logical data format

5. OpenLR™ - Logical Data format specification

A location reference is a description of a designated part of a digital map or a sequence of geographical positions. OpenLR™ uses for this description the model of “location reference points” (LR-points, see 5.2).

A location reference for line locations contains at least two LR-points but there is no maximum number of LR-points defined. The location reference path is the path in the digital map described by the LR-points. It can be found by a shortest-path calculation between each consecutive pair of LR-points. Point locations consist of geo-coordinates or exactly two LR-points or a combination of both.

The logical data format describes the logical model for location references according to the OpenLR™ standard.

5.1 Data types

The logical data format makes use of the following data types:

| Data type | Description |
|-----------|--|
| float | Single precision floating-point numbering format |
| integer | Natural number including 0 together with the negatives of the non-zero natural numbers |

Table 3: Data types (logical data format)

5.2 Location Reference Point

The basis of a location reference is a sequence of location reference points (LR-points). Such an LR-point contains a coordinate pair, specified in WGS84 longitude and latitude values and additionally several attributes.

The coordinate pair (see 5.3.1) represents a geographical position within a map/network and is mandatory for an LR-point. The coordinate pair may belong to a “real” node within a network (in most cases these nodes represent junctions in the real world).

The attributes (see section 5.3.2 to 5.3.6) describe values of a line within a network at which the line is connected to the node described by the coordinate pair. In this context it is not defined if the attributes refer to an incoming or outgoing line regarding the node. This will be specified in section 5.5.

5.2.1. Topological connection of LR-points

The location reference points must be stored in a topological order so that a point A will directly follow a point B if B also comes after A in the location reference path and there is no other LR-point in between. This topological order defines a “next point”-relationship of successive LR-points. The last point in this order has no next point in this relationship.

Figure 6 shows an example of this relationship. The LR-points are indicated by A, B and C and the black lines and arrows indicate the order of the points from A to C in the location reference path. In this example the LR-point A has B as next point, B has C as next point and C has no next point.

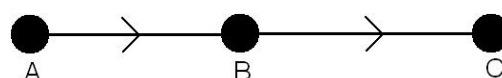


Figure 6: Connection of LR-points

5.3 Components of LR-points

This section describes the components of a location reference point (LR-point).

5.3.1. Coordinate pair

Coordinate pair stands for a pair of WGS84 longitude (lon) and latitude (lat) values. This coordinate pair specifies a geometric point in a digital map. The lon and lat values are stored in a decamicrodegrees resolution (10^{-5} , five decimals).

Abbreviation: COORD

Type: (float, float)

5.3.2. Functional Road Class

The functional road class (FRC) of a line is a road classification based on the importance of the road represented by the line. The possible values of the FRC attribute are shown in Table 4, covering the range of navigable roads from highest to lowest importance. If there are fewer or more FRC values defined in the encoder map, or decoder map respectively, than these 8 values are used for location referencing, then a proper mapping needs to be done or less important classes need to be ignored.

| FRC |
|---------------------------|
| FRC 0 – Main road |
| FRC 1 – First class road |
| FRC 2 – Second class road |
| FRC 3 – Third class road |
| FRC 4 – Fourth class road |
| FRC 5 – Fifth class road |
| FRC 6 – Sixth class road |
| FRC 7 – Other class road |

Table 4: Logical format: Functional road class

Abbreviation: FRC

Type: integer

5.3.3. Form of way

The form of way (FOW) describes the physical road type of a line. The possible values of the FOW attribute are shown in Table 5.

| FOW | Description |
|----------------------|---|
| UNDEFINED | The physical road type is unknown. |
| MOTORWAY | A Motorway is defined as a road permitted for motorized vehicles only in combination with a prescribed minimum speed. It has two or more physically separated carriageways and no single level-crossings. |
| MULTIPLE_CARRIAGEWAY | A multiple carriageway is defined as a road with physically separated carriageways regardless of the number of lanes. If a road is also a motorway, it should be coded as such and not as a multiple carriageway. |
| SINGLE_CARRIAGEWAY | All roads without separate carriageways are considered as roads with a single carriageway. |
| ROUNDABOUT | A Roundabout is a road which forms a ring on which traffic traveling in only one direction is allowed. |
| TRAFFICSQUARE | A Traffic Square is an open area (partly) enclosed by roads which is used for non-traffic purposes and which is not a Roundabout. |
| SLIPROAD | A Slip Road is a road especially designed to enter or |

| | |
|-------|--|
| | leave a line. |
| OTHER | The physical road type is known but does not fit into one of the other categories. |

Table 5: Logical Format: Form of wayAbbreviation: FOWType: integer

5.3.4. Bearing

The bearing (BEAR) describes the angle between the true North and a line which is defined by the coordinate of the LR-point and a coordinate which is BEARDIST along the line defined by the LR-point attributes. If the line length is less than BEARDIST then the opposite point of the line is used (regardless of BEARDIST). The bearing is measured in degrees and always positive (measuring clockwise from North). The parameter BEARDIST is defined in Table 6.

Abbreviation: BEARType: integer

| Abbreviation | Description | Value | Unit |
|--------------|---|-------|--------|
| BEARDIST | distance between two coordinates which form a line for the calculation of the bearing value | 20 | meters |

Table 6: Logical format: Parameter BEARDIST

Figure 7 shows how the second point for the bearing calculation is determined. The figure shows a line from A to B which is longer than BEARDIST. The shaded part of this line is exactly BEARDIST meters long so that the point marked with B' is BEARDIST meters away from A when we walk along the line from A to B. The straight line from A to B' is now considered for the calculation of the bearing value. Note, this is different to the angle we would have calculated if we always take the opposite node of line (in this case, this would be B).

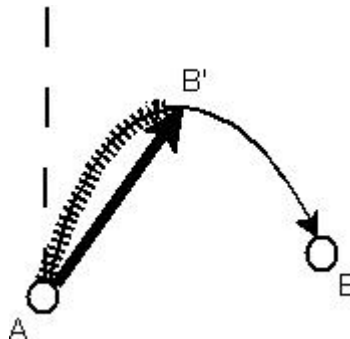
**Figure 7: Logical format: Bearing point**

Figure 8 shows two examples of the bearing value calculation. There are two lines, one from A to B and one from A to C. For both lines the arcs indicate the angles to the North.

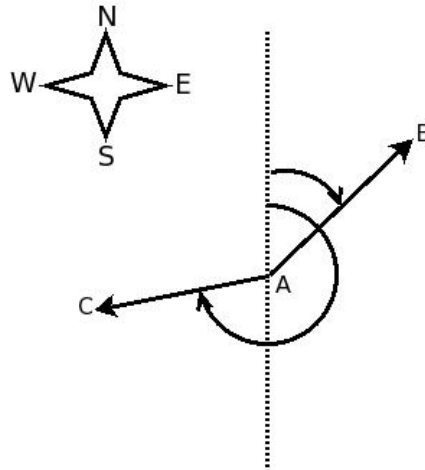


Figure 8: Logical format: Bearing

5.3.5. Distance to next LR-point

The distance to next point (DNP) field describes the distance to the next LR-point in the topological connection of the LR-points. The distance is measured in meters and is calculated along the location reference path between two subsequent LR-points. The last LR-point has the distance value 0.

Abbreviation: DNP

Type: integer

Figure 9 shows an example of the distance calculation and assignment. The three LR-points are in a sequence from A over B to C. Therefore the distance between A and B along the location reference path is assigned to A. The LR-point B holds the distance between B and C and the LR-point C has a distance value of 0.

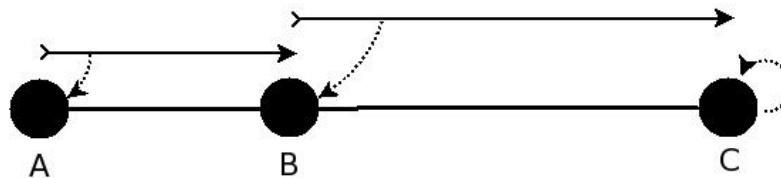


Figure 9: Logical format: Distance to next point

5.3.6. Lowest FRC to next LR-point

The lowest FRC (LFRCNP) is the lowest FRC value which appears in the location reference path between two consecutive LR-points. This information could be used to limit the number of road classes which need to be scanned during the decoding. The highest FRC value is 0 and the lowest possible FRC value is valued with 7.

Abbreviation: LFRCNP

Type: integer

5.3.7. Side of road

The side of road information (SOR) describes the relationship between the point of interest and a referenced line. The point can be on the right side of the line, on the left side of the line, on both sides of the line, or directly on the line.

| SOR | Description |
|-----|---|
| 0 | Point is directly on (or above) the road, or determination of right/left side is not applicable (default) |

| | |
|---|---------------------------------------|
| 1 | Point is on right side of the road |
| 2 | Point is on the left side of the road |
| 3 | Point is on both sides of the road |

Abbreviation: SOR

Type: integer

5.3.8. Orientation

The orientation information (ORI) describes the relationship between the point of interest and the direction of a referenced line. The point may be directed in the same direction as the line, against that direction, both directions, or the direction of the point might be unknown.

| ORI | Description |
|-----|--|
| 0 | Point has no sense of orientation, or determination of orientation is not applicable (default) |
| 1 | Point has orientation from first LRP towards second LRP |
| 2 | Point has orientation from second LRP towards first LRP |
| 3 | Point has orientation in both directions |

Table 7: Orientation of a point location

Abbreviation: ORI

Type: integer

5.4 Offsets

Offsets are needed to locate start and end of a line location and the position of a point location precisely (point location along a line). Offset values shorten the location reference path down to the desired location and are measured in meter.

Line locations in OpenLR™ rely on the condition that the path defined by the location reference points (location reference path) covers the location completely. The location reference path may even be longer than the location path. Point locations identified by a network line might also use offsets to determine the concrete position of the desired point on that line.

OpenLR™ defines two offset values. A positive offset (POFF) is used to locate the precise start of a location. The POFF defines the distance between the start of the location reference path and the start of the location. The negative offset (NOFF) is used to locate the precise end of the location and it defines the distance between the end of the location and the end of the location reference path.

5.4.1. Positive offset

The positive offset (POFF) is the difference of the start point of the location and the start point of the desired location along the location reference path. The value is measured in meters. Figure 10 shows an example for the calculation of the positive and negative offset. The lines indicate the location reference path and the hatching indicates the desired location.

Abbreviation: POFF

Type: integer

5.4.2. Negative offset

The negative offset (NOFF) is the difference of the end point of the desired location and the end point of the location reference path. The value is measured in meters. Figure 10 shows an example for the calculation of the positive and negative offset. The lines are indicating the location reference path and the hatching indicates the desired location.

Abbreviation: NOFF

Type: integer



Figure 10: Logical format: Positive and negative offset

5.5 Relationship Attributes – LR-point

All attributes are linked to an LR-point. For all LR-points (except that last LR-point) the attributes describe an outgoing line of the node at the LR-point coordinate which belongs to the location. The attributes of the last LR-point direct to an incoming line of the node at the LR-point coordinate which belongs to the location.

Figure 11 shows an example for the relationship between an LR-point and the attributes. The lines indicate the location reference path and the nodes A, B and C are the LR-points. Note that there is also a line whose start and end node is not a LR-point (the third line in the sequence). This line does not need to be referenced because it is covered by the shortest path between the LR-points B and C.

The LR-points A and B direct to an outgoing line and the last LR-point C directs to an incoming line.

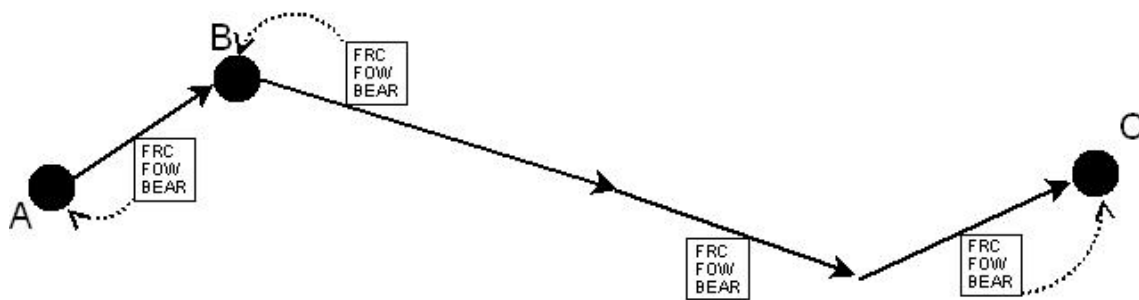


Figure 11: Relationship: Attributes – LR-point

6. Data format rules

The OpenLR™ rules describe additional regulations for OpenLR™ location references, supplementary to the general “shortest-path” rule for sub-dividing the location reference path. These rules are used to simplify the encoding and decoding process and to increase the accuracy of the results.

Rule – 1 *The maximum distance between two location reference points shall not exceed 15km. The distance is measured along the location reference path. If this condition is not fulfilled for a location reference then a sufficient number of additional LR-points shall be inserted.*

The maximum distance between two consecutive location reference points is restricted in order to speed up shortest-path computation because several short routes can be computed quicker than one large route if the routing algorithm has to take the whole network into account. The restriction also provides the opportunity to form a compact physical format with an acceptable accuracy.

Rule – 2 *All lengths are integer values. If there are float values available then we will round these values to get an integer representation.*

Different maps might store the length values in different formats and also with different precision. The uniform basis for all is the usage of integer values.

Rule – 3 *Two LR-points are mandatory for line locations and the number of intermediate LR-points is not limited.*

A line location reference must always have at least two location reference points indicating the start and the end of the location. In addition, further (mandatory) intermediate location reference points are to be included if conditions of shortest-path or Rule 1 are met. If the encoder detects critical situations where the decoder (on a different map) might get into trouble, the location reference might be enhanced with additional (optional) intermediate LR-points.

Rule – 4 *The coordinates of the LR-points shall preferably be chosen on valid network nodes.*

Valid network nodes shall represent junctions in the real world and it is expected that these junctions can be found in different maps with a higher probability than positions somewhere on a line. Additional nodes other than junctions (invalid nodes) shall be avoided if these can be easily skipped during a route search. At these invalid nodes it is not possible to deviate from a route.

Nodes having only one incoming and one outgoing line (as directed connectivity) shall be invalid and therefore avoided since these nodes are not related to junctions (see Figure 12, node in the middle). These nodes can be stepped over during a route search. Nodes which have two incoming and two outgoing lines and there exists only two adjacent nodes shall also be invalid, unless a U-turn is possible at that node (see Figure 13, node in the middle).



Figure 12: Invalid node (1)



Figure 13: Invalid node (2)

If one of these nodes is selected for an LR-point then this LR-point should be shifted along the location reference path in order to find a suitable node. This can be done since a route calculation will step over such invalid nodes without leaving the desired path.

If the encoder cannot determine a valid node at all, then exceptionally an invalid node may be chosen for an LR-Point. Examples are lines which are longer than 15 km (see Rule – 1).

In order to fulfill Rule – 4 it is recommended to check the start node and end node of the location path. If the start or the end of a location is placed on invalid nodes then the encoder might expand the location uniquely and should find a suitable node outside of the location. This expansion must never go into the location because this will shorten the location's spatial extent.

Rule – 5 *Offsets must refer to the location reference path defined by the first two (the last two) location reference points.*

Offsets may be placed only between the first two location reference points (positive offset) and between the last two location reference points (negative offset). Otherwise the first (or last) location reference point is useless as it brings no additional information to describe the location or the offsets. Figure 14 shows where the offsets values may be placed along a location and it also indicates that the maximum offset value is 15000 meters (see Rule – 1).



Figure 14: Offset placement and restrictions

6.1 Overview of the data format rules

The following Table 8 summarizes the data format rules.

| Rule | Description | Value |
|--------|--|---|
| Rule 1 | max distance between two consecutive LR-points | 15000 m |
| Rule 2 | road length values | treated as integer values |
| Rule 3 | number of LR-points | at least two LR-points (for line locations) |
| Rule 4 | invalid nodes | LR-points shall preferably be placed on valid network nodes |
| Rule 5 | offsets | offsets must refer to the first two (last two) LR-points |

Table 8: Data format rules overview



Chapter D

Physical data format (Binary)

7. OpenLR™ - Binary representation

This physical data format describes a byte-oriented stream format for the OpenLR™ data format. It bases on the components described in the logical data format in section 5. This section defines the binary format versions 2 and 3. It is recommended to use version 3.

7.1 Data types

The physical data format uses the following data types. Table 9 gives an overview of all available data types and specifies the name, the type and the designated size of each data type. In the following sections the data type names are used to indicate the size and type for each data component.

| Data type name | Type | Size | Range |
|----------------|---------------------------|---------|--------------------------|
| Boolean | flag with true=1, false=0 | 1 bit | 0 – 1 |
| uByte | unsigned integer | 1 byte | 0 – 255 |
| uShort | unsigned integer | 2 bytes | 0 – 65535 |
| uSmallInt | unsigned integer | 3 bytes | 0 – 16777215 |
| uInteger | unsigned integer | 4 bytes | 0 – 4294967295 |
| sByte | signed integer | 1 byte | -128 – 127 |
| sShort | signed integer | 2 bytes | -32768 – 32767 |
| sSmallInt | signed integer | 3 bytes | -8388608 – 8388607 |
| sInteger | signed integer | 4 bytes | -2147483648 – 2147483647 |
| String[n] | array of n characters | n bytes | variable size |
| BitField[n] | array of n bits | n bits | variable size |

Table 9: Physical format: Data types (Binary)

Negative integer values are stored in the two's complement format.

7.2 Coordinates (COORD)

Each point in a map consists of a coordinate pair “longitude” (lon) and “latitude” (lat) represented in WGS84 coordinates. The directions north and east are represented by positive values (longitude and latitude respectively). The lon and lat values are stored in a decamicrodegrees resolution (10^{-5} , five decimals).

The coordinate values will be transmitted as integer values. These values will be generated using Equation 1 which calculates a 24-bit integer representation. The resolution parameter is set to 24. This translation leads to a coordinate resolution (“error”) of about 2.4 meter at most. The backward translation is described in Equation 2. Both equations make use of the signum function which is -1 for negative values, 1 for positive values and 0 otherwise.

$$\text{int} = \left(\text{sgn}(\text{deg}) * 0.5 + \frac{\text{deg} * 2^{\text{Re solution}}}{360^\circ} \right)$$

Equation 1: Transformation from decimal coordinates into integer values (Binary)

$$\text{deg} = \left(\frac{(\text{int} - \text{sgn}(\text{int}) * 0.5) * 360^\circ}{2^{\text{Re solution}}} \right)$$

Equation 2: Transformation from integer values into decimal coordinates (Binary)

The physical format makes use of an absolute and a relative coordinate format. The absolute format represents the designated values of the geographical position and the relative value is the offset of the coordinates relative to the preceding coordinate.

7.2.1. Absolute format

The absolute format describes geographical positions in a 24-bit resolution. Table 10 shows the data type used for the absolute format.

| Data type | Value | Description |
|-----------|---------------------|-----------------------|
| sSmallInt | -8388608 – +8388607 | 24 bit representation |

Table 10: Physical format: Coordinate format (absolute) (Binary)

7.2.2. Relative format

The relative format is used to describe differences between two consecutive coordinates. The difference is calculated for each value (lon/lat) separately as shown in Equation 3. The current and previous values represent the latitude (longitude) value in degrees. The difference between these two values is multiplied with 100000 (10^5) in order to resolve an integer value.

$$relative = round(100000 * (currentPo\ int - previousPo\ int))$$

Equation 3: Relative coordinates calculation (Binary)

Table 11 shows the data type for relative offsets.

| Data type | Value | Description |
|-----------|-----------------|------------------------------|
| sShort | -32768 – +32767 | 2 bytes relative coordinates |

Table 11: Physical format: Coordinate format (relative) (Binary)

Table 12 shows the maximum distances which are possible for a location in the Netherlands. The ranges for relative coordinates are calculated for a given coordinate in the Netherlands at lon = 5° and lat = 52°.

| byte | latitude | | longitude | |
|------|-------------|-------------|-------------|-------------|
| | lower bound | upper bound | lower bound | upper bound |
| 2 | -36459 m | 36460 m | -22504 m | 22504 m |

Table 12: Physical format: Longitude/Latitude ranges for relative coordinates (Example)

Important Note

The interval lengths for the latitude are fixed and do not change if the longitude values change. But the interval lengths for the longitude depend on the latitude values and they change if the latitude values change. Figure 15 shows a chart of the maximum ranges for the longitude values depending on the latitude values. The difference between the longitude values is exactly 0.32767° , the maximum value being covered by the 16 bit representation.

The graph indicates that Rule – 1 of the data format rules is applicable for all latitude values in the range $-65.70^\circ < lat < 65.70^\circ$. The maximum distance of the longitude values is here greater than the 15km defined in Rule – 1.

For latitude values lower than -65.70° and also for latitude values greater than 65.70° the distance between two longitude values which differ by 0.32767° is less than 15 km. In these areas Rule – 1 needs to be adapted to a lower maximum range. If the encoder still uses the value of 15 km then the binary format cannot be used any further. The available range for relative coordinates will be exceeded.

Therefore it is recommended for such cases to use a lower maximum distance value. Lower values also fulfill the requirements of Rule – 1 and do not break the validity of the OpenLR™ method but the resulting location reference might contain more LR-points. Other physical data formats may not be affected by this issue.

Distance between longitudes (vary by 0.32767°)

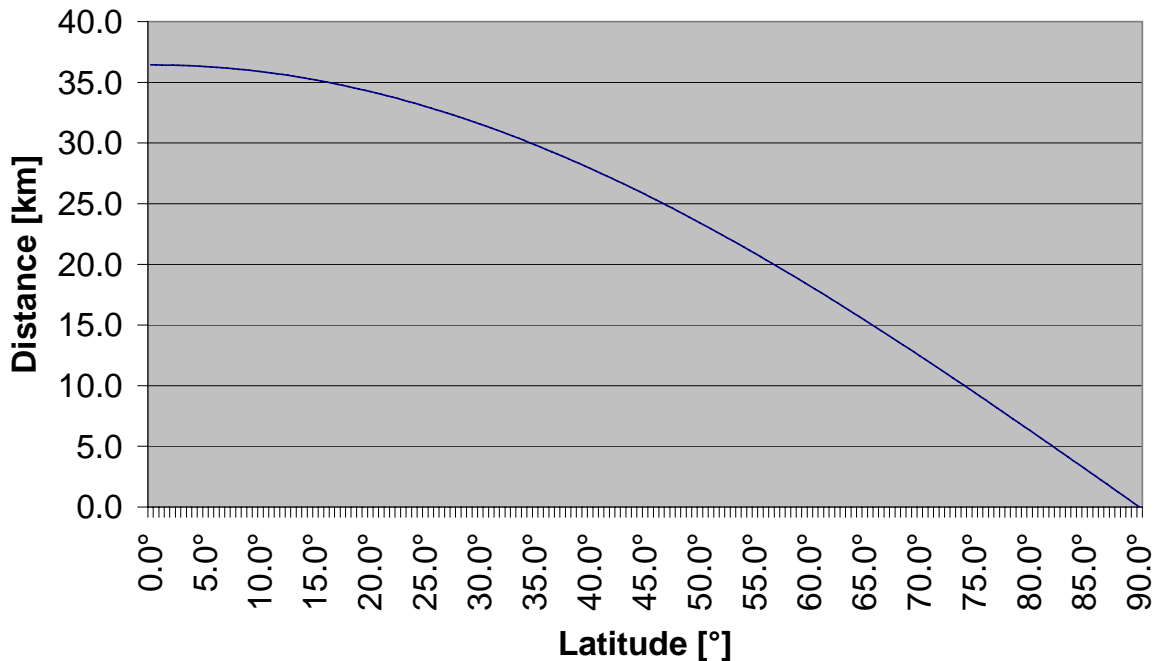


Figure 15: Distances between longitudes

The Equation 4 shows the spherical law of cosines formula. This formula can be used to calculate the distance between two coordinates. The latitude and longitude values are expressed in radians and need to be converted from degrees to radians first.

$$d = \arccos(\sin(lat1) * \sin(lat2) + \cos(lat1) * \cos(lat2) * \cos(lon2 - lon1)) * 6371$$

Equation 4: Spherical law of cosines

The Equation 4 can be simplified for calculating the maximum range of longitudes. The resulting distance can then be used in Rule – 1 as a new maximum distance between two subsequent LR-points.

The term $\cos(lon2 - lon1)$ can be simplified to the constant value $\cos(rad(0.32767^\circ))$. This expresses the maximum difference in degree which is covered by the 16 bit relative offset representation. The equation can further be simplified by determining the largest latitude value (lowest latitude value for negative numbers) along the location. This ensures that the maximum distance value is valid for the whole location.

The resulting formula is shown in Equation 5: Calculation of the maximum longitude range using “mlat” as the largest (lowest) latitude value along the location.

$$\max_{Distance} = \arccos(\sin^2(rad(mlat)) + \cos^2(rad(mlat)) * \cos(rad(0.32767^\circ))) * 6371$$

Equation 5: Calculation of the maximum longitude range

7.3 Attribute values

The binary format of the attributes will follow in this section.

7.3.1. Functional Road Class (FRC)

The functional road class (FRC) can hold eight different values as described in the logical format. These eight values are represented by 3 bits and the mapping is shown in Table 13.

| Data type | Value (integer) | Value (binary) | Description |
|-------------|-----------------|----------------|---|
| BitField[3] | 0 | 000 | FRC 0 – Main road, highest importance |
| | 1 | 001 | FRC 1 – First class road |
| | 2 | 010 | FRC 2 – Second class road |
| | 3 | 011 | FRC 3 – Third class road |
| | 4 | 100 | FRC 4 – Fourth class road |
| | 5 | 101 | FRC 5 – Fifth class road |
| | 6 | 110 | FRC 6 – Sixth class road |
| | 7 | 111 | FRC 7 – Other class road, lowest importance |

Table 13: Physical format: Functional road class (Binary)

7.3.2. Form of way (FOW)

The form of way (FOW) can hold eight different values as described in the logical format. These eight values are represented by 3 bits and the mapping is shown in Table 14.

| Data type | Value (integer) | Value (binary) | Description |
|-------------|-----------------|----------------|----------------------|
| BitField[3] | 0 | 000 | UNDEFINED |
| | 1 | 001 | MOTORWAY |
| | 2 | 010 | MULTIPLE_CARRIAGEWAY |
| | 3 | 011 | SINGLE_CARRIAGEWAY |
| | 4 | 100 | ROUNDAABOUT |
| | 5 | 101 | TRAFFICSQUARE |
| | 6 | 110 | SLIPROAD |
| | 7 | 111 | OTHER |

Table 14: Physical format: Form of way (Binary)

7.3.3. Bearing (BEAR)

The bearing describes the angle between the true North and the road as described in the logical format. The physical data format defines 32 sectors whereby each sector covers 11.25° of the circle. These 32 sectors are represented by 5 bits. Table 15 shows the data type for the bearing attribute and Table 16 shows the mapping from the sectors to the concrete value.

| Data type | Value | Description |
|-------------|-------|---|
| BitField[5] | 0-31 | Number of the sector in which the angle between the true North and the line specified in the logical data format is located; the full circle is divided into 32 sectors each covering an angle of 11.25°. |

Table 15: Physical format: Bearing (Binary)

| Value | Sector | Value | Sector |
|-------|------------------------|-------|------------------------|
| 0 | 000.00° <= x < 011.25° | 16 | 180.00° <= x < 191.25° |
| 1 | 011.25° <= x < 022.50° | 17 | 191.25° <= x < 202.50° |
| 2 | 022.50° <= x < 033.75° | 18 | 202.50° <= x < 213.75° |
| 3 | 033.75° <= x < 045.00° | 19 | 213.75° <= x < 225.00° |
| 4 | 045.00° <= x < 056.25° | 20 | 225.00° <= x < 236.25° |
| 5 | 056.25° <= x < 067.50° | 21 | 236.25° <= x < 247.50° |
| 6 | 067.50° <= x < 078.75° | 22 | 247.50° <= x < 258.75° |
| 7 | 078.75° <= x < 090.00° | 23 | 258.75° <= x < 270.00° |
| 8 | 090.00° <= x < 101.25° | 24 | 270.00° <= x < 281.25° |
| 9 | 101.25° <= x < 112.50° | 25 | 281.25° <= x < 292.50° |

| | | | |
|----|-----------------------|----|-----------------------|
| 10 | 112.50° ≤ x < 123.75° | 26 | 292.50° ≤ x < 303.75° |
| 11 | 123.75° ≤ x < 135.00° | 27 | 303.75° ≤ x < 315.00° |
| 12 | 135.00° ≤ x < 146.25° | 28 | 315.00° ≤ x < 326.25° |
| 13 | 146.25° ≤ x < 157.50° | 29 | 326.25° ≤ x < 337.50° |
| 14 | 157.50° ≤ x < 168.75° | 30 | 337.50° ≤ x < 348.75° |
| 15 | 168.75° ≤ x < 180.00° | 31 | 348.75° ≤ x < 360.00° |

Table 16: Physical format: Bearing value definition (Binary)

Equation 6 outlines the calculation of the bearing value and Figure 16 provides a graphical overview of the sectors.

$$value = \left\lfloor \frac{angle}{11.25^\circ} \right\rfloor, 0^\circ \leq angle < 360^\circ$$

Equation 6: Calculation of the bearing value (Binary)

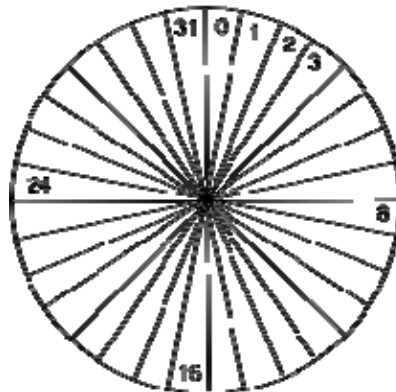


Figure 16: Physical format: Bearing sectors (Binary)

7.3.4. Distance to next LR-point (DNP)

The DNP attribute measures the distance between two consecutive LR-points along the location reference path as described in the logical format.

The physical data format defines an 8-bit representation and Table 17 shows the data type used for DNP. This representation defines 255 intervals and in combination with Rule – 1 of the data format rules (maximum length between two consecutive LR-points is limited by 15000m) each interval will have a length of 58.6 meters.

| Data type | Value | Description |
|-------------|-------|---|
| BitField[8] | 0-255 | Distance interval according to Equation 7 |

Table 17: Physical format: Distance to next point (Binary)

Equation 7 shows how the DNP values can be calculated.

$$value = \left\lfloor \frac{length}{58.6m} \right\rfloor$$

Equation 7: Calculation of the DNP value (Binary)

7.3.5. Lowest FRC to next point (LFR CNP)

The lowest FRC to the next point indicates the lowest functional road class used in the location reference path to the next LR-point. This information could be used to limit the number of road classes which need to be scanned during the decoding. See Table 18 for a definition of the data type.

| Data type | Value | Description |
|-------------|-------|--|
| BitField[3] | 0-7 | Holds the same values as described in Table 13 |

Table 18: Physical format: Lowest FRC to next point (Binary)

7.3.6. Orientation information

A point location may have additional information in which direction the information at that point is relevant. Examples for such point locations are speed cams or traffic signs. This additional information is valid for the types “point along line” and “poi with access point”. The orientation information type will be expressed by two bits. The meaning of the allowed values is shown in Table 19.

The orientation information is available since binary format version 3.

| Data type | Value (binary) | Meaning |
|-------------|----------------|--|
| BitField[2] | 00 | Point has no sense of orientation, or determination of orientation is not applicable (default) |
| | 01 | Point has orientation from first LRP towards second LRP |
| | 10 | Point has orientation from second LRP towards first LRP |
| | 11 | Point has orientation in both directions |

Table 19: Orientation of a point location

7.3.7. Side of road information

A point location may have additional information on which side of the road the point is located. Examples for such point locations are particular house numbers, speed cams or traffic signs. This additional information is valid for the types “point along line” and “poi with access point”. The side information type will be expressed by two bits. The meaning of the allowed values is shown in Table 20.

The side of road information is available since binary format version 3.

| Data type | Value (binary) | Meaning |
|-------------|----------------|---|
| BitField[2] | 00 | Point is directly on (or above) the road, or determination of right/left side is not applicable (default) |
| | 01 | Point is on right side of the road |
| | 10 | Point is on the left side of the road |
| | 11 | Point is on both sides of the road |

Table 20: Side of the road of a point location

7.4 Location Reference header

The Location Reference header contains general information about the reference.

7.4.1. Version (VER)

The version is used to distinguish between several physical and data formats for location references. The version number is represented by 3 bits and the data type is shown in Table 21.

| Data type | Value | Description |
|-------------|-------|------------------------|
| BitField[3] | 0-7 | current version number |

Table 21: Physical format: Version (Binary)

Note:

The actual version of the (binary) physical data format is 3 so that the VER field is constantly set to binary 011.

Note:

The version number of the physical format is independently controlled from the version of the OpenLR™ method. An implementation of a given version of the OpenLR™ method may support multiple versions of the physical format. Likewise, a particular version of the physical format may be used throughout the evolution of OpenLR™ across several versions of the method.

7.4.2. Attribute flag (AF)

The attribute flag indicates whether there are attributes appended to each LR-point or not. The AF value is 0 if no attributes are appended and therefore the location reference only consists of coordinates. Otherwise a value of 1 indicates that attributes are appended to each LR-point. The data type for the AF is shown in Table 22 and Table 23.

| Data type | Value | Description |
|-----------|-------|--|
| Boolean | 0, 1 | flag, indicating whether attributes are appended to each LR-point or not |

Table 22: Physical format: Attribute flag (Binary)

| Value | Description |
|-------|---|
| 0 | no attributes are appended |
| 1 | for each LR-point a set of attributes is appended |

Table 23: Physical format: Attribute flag values (Binary)

Note:

Since the current version of the physical format supports only location references including attributes the AF flag will constantly be set to 1.

7.4.3. Area flag (ArF)

The area flag indicates whether the location reference describes an area or not. If this flag is set then the location shall be connected and we describe an area.

| Data type | Value | Description |
|-----------|-------|--|
| Boolean | 0, 1 | flag, indicating whether the location reference describes an area or not |

Table 24: Physical format: Area flag (Binary)

| Value | Description |
|-------|--|
| 0 | location reference does not describe an area |
| 1 | location reference describes an area |

Table 25: Physical format: Area flag values (Binary)

Note:

The current physical data format supports only point and line locations so that the ArF is constantly set to 0.

7.4.4. Point flag (PF)

The point flag indicates whether the location reference describes a point or not. If this flag is set then the location shall describe a point location.

| Data type | Value | Description |
|-----------|-------|-------------|
|-----------|-------|-------------|

| | | |
|---------|------|--|
| Boolean | 0, 1 | flag, indicating whether the location reference describes a point or not |
|---------|------|--|

Table 26: Physical format: Point flag (Binary)

| Value | Description |
|-------|--|
| 0 | location reference does not describe a point |
| 1 | location reference describes a point |

Table 27: Physical format: Point flag values (Binary)

7.5 Offsets

Offsets are used to locate the start and end of a location more precisely than being bound to the nodes in a network. The logical format defines two offsets, one at the start of the location and one at the end of the location and both offsets operate along the lines of the location and are measured in meters. The offset values are not mandatory and a missing offset value means an offset of 0 meters. Offsets are also only valid for location references which have attributes included.

7.5.1. Offset flags

Offset flags indicate whether the data includes a specific offset information or not. The physical data format deals with two flags corresponding to the two different offset values. The positive offset flag (PoffF) and the negative offset flag (NoffF) are described in Table 28 and Table 29.

| Data type | Value | Description |
|-----------|-------|--|
| Boolean | 0, 1 | flag, indicating whether the corresponding offset value is included in the data or not |

Table 28: Physical format: Offset flag (Binary)

| Value | Description |
|-------|---|
| 0 | location reference data does NOT include the corresponding offset information |
| 1 | location reference data includes the corresponding offset information |

Table 29: Physical format: Offset flag values (Binary)

7.5.2. Offset values

There are two different versions of how the offsets values are packed into the binary format. The “equal distribution” method is only supported by the binary data format version 2. The more accurate “percentage” method is used for version 3 of the binary data format.

7.5.2.1 Equal distribution (Format version 2)

The offset values (positive and negative, POFF and NOFF) indicate the distance between the start (end) of the location reference path and the “real” start (end) of the location.

The physical data format defines an 8-bit representation for each offset value. Table 30 shows the data type used for POFF and NOFF. This representation allows us to define 256 intervals with a length of each interval of 58.6 meter. The interval number calculation for offsets is outlined in Equation 8.

| Data type | Value | Description |
|-------------|-------|-------------------------------------|
| BitField[8] | 0-255 | offset length interval according to |



| | | |
|--|--|------------|
| | | Equation 8 |
|--|--|------------|

Table 30: Physical format: Offset (Binary)

$$value = \left\lfloor \frac{offset\ length}{58.6m} \right\rfloor$$

Equation 8: Calculation of offset values (Binary)

This method distributes the (possible) offset values over the range of 256 buckets and makes use of the fact that the maximum offset value is 15000 meter. Each bucket refers to an interval of 58.6 meter. Figure 17 shows the distribution graphically.

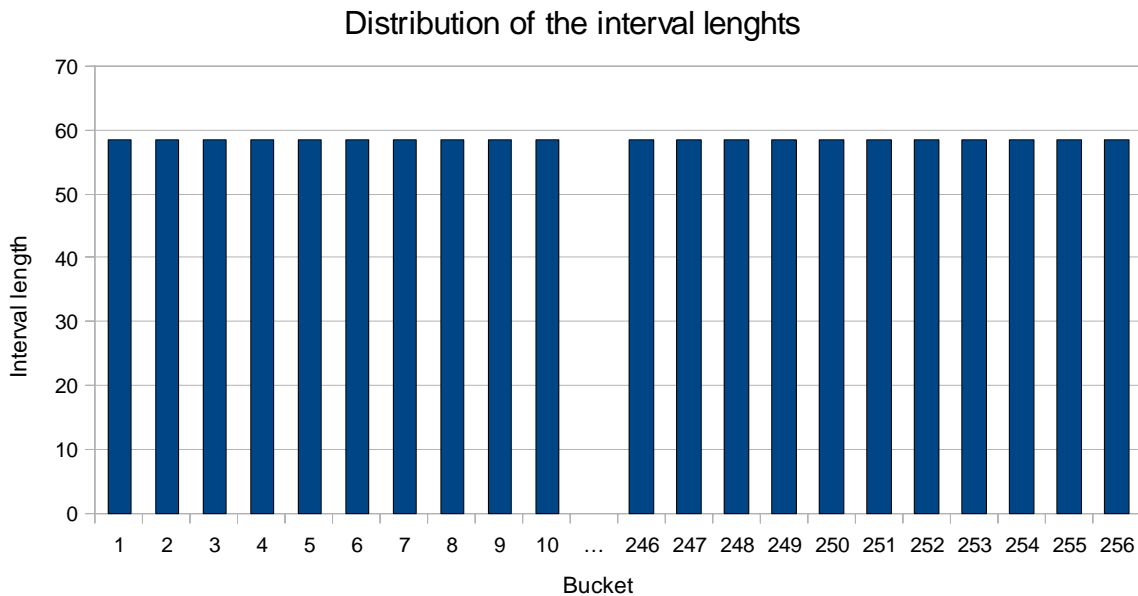


Figure 17: Distribution of the interval lengths

7.5.2.2 Percentage (Format version 3)

The “percentage” method bases on relative offset values instead of absolute values as used in the “equal distribution” method. The value being used here is the relation of the offset length to the length of the path between the first two location reference points (last two location reference points for the negative offset). The length between these two LR-points shall be called LRP length.

The relative value (or percentage) will then be equally distributed over the available 256 buckets so that every bucket covers 0.390625% of the LRP length. This method takes the length of a location into account which results in a better accuracy for shorter locations. Even for shorter locations the whole bucket range will be used and longer locations will not have an accuracy dropdown (compared to the “equal distribution” implementation) as no offset value can be greater than 15000 meter irrespective of the location length.

This method also uses eight bits for representing the percentage values and Table 31 shows the data type used for POFF and NOFF.

| Data type | Value | Description |
|-------------|-------|-------------------------------------|
| BitField[8] | 0-255 | offset length interval according to |

Table 31: Physical format: Offset (Binary)

Equation 9 shows how the bucket index can be calculated having the offset and location length available. Figure 18 shows some interval lengths (in meter) for selected LRP lengths. It is obvious that the accuracy is higher if the location length is lower.

$$y = \left\lfloor \frac{256 * offset_length}{LRP_length} \right\rfloor$$

Equation 9: Bucket index calculation (percentage method)

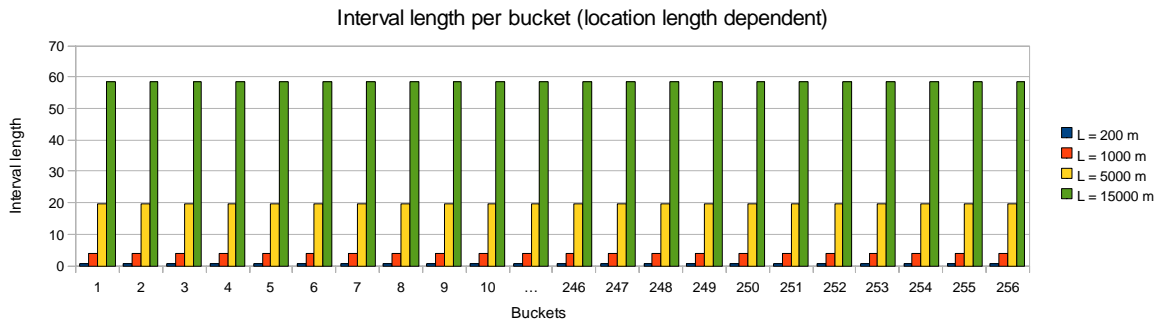


Figure 18: Interval lengths (percentage method)

Recommendation

The intention of using the “percentage” method is to specify the offset values more accurate than it could be done with the “equal distribution” method. If the encoder is forced to generate highly accurate offset values, it is recommended to add an additional LR-point. This additional LR-point shall reduce the LRP_length value and shall be placed between the first two (last two) LR-points. The path between the first (last) LR-point and the additional LR-point must be at least as long as the offset but shorter than the original distance between the first two (last two) LR-points.

This recommendation might violate Rule – 4 if the additional LR-point cannot be placed on a valid node but the generated location reference shall still be valid. It might even happen that the additional LR-point is placed directly on a line. In such a case the coordinates are taken from a point along the line and the attributes of the LR-point are derived from that line.

The decoder might not find a node for this LR-point so it needs to be ensured that the decoder also implements a strategy where lines can be determined without using nodes. An implementation following this recommendation also increases the location reference data size.

8. OpenLR™ - Physical data format binary specification

This section describes the arrangement of the data fields in a byte stream. It is assumed that a byte-oriented stream applies and 8 bits per byte are used. The byte order being used is the big endian format. This means that the most significant byte (MSB) comes first in the sequence of data.

The following tables show the order of the data from top to bottom. Each table describes a special location type and contains a sequence number, a field name and the assignment of each bit.

8.1 Line location

This location type was already supported in version 2. In version 3 the line location format will use the “percentage” offset implementation.

8.1.1. Line Location with 2 location reference points

A line location reference consists of at least 2 LR-points.

| Byte number | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|---------------|---------------------------------|--------------|-----------------------|-------------|--------------------|---------------|-------|-------|
| 1 | Status | RFU | RFU | 0 (no point) | 0 (no area) | 1 (has attributes) | 011 (version) | | |
| 2 | Longitude (1) | longitude highest byte | | | | | | | |
| 3 | Longitude (2) | longitude middle byte | | | | | | | |
| 4 | Longitude (3) | longitude lowest byte | | | | | | | |
| 5 | Latitude (1) | latitude highest byte | | | | | | | |
| 6 | Latitude (2) | latitude middle byte | | | | | | | |
| 7 | Latitude (3) | latitude lowest byte | | | | | | | |
| 8 | Attribute 1 | RFU | RFU | functional road class | | | form of way | | |
| 9 | Attribute 2 | lowest frc next point | | | bearing | | | | |
| 10 | Attribute 3 | distance to next point | | | | | | | |
| 11 | Longitude (1) | relative longitude highest byte | | | | | | | |
| 12 | Longitude (2) | relative longitude lowest byte | | | | | | | |
| 13 | Latitude (1) | relative latitude highest byte | | | | | | | |
| 14 | Latitude (2) | relative latitude lowest byte | | | | | | | |
| 15 | Attributes 1 | RFU | RFU | functional road class | | | form of way | | |
| 16 | Attributes 4 | RFU | 0/1 (posOff) | 0/1 (negOff) | bearing | | | | |
| 17 | PosOffset | positive offset (optional) | | | | | | | |
| 18 | NegOffset | negative offset (optional) | | | | | | | |

Table 32: Line location with 2 LR-Points

Data size: 16 + [0/1/2] bytes

Note: The binary format version 2 defines bit 5 in the status byte (byte 1) as RFU and the bits 2 – 0 are set to 010. The offset values (bytes 17 and 18) use the equal distribution method.

8.1.2. Line location with n location reference points (n > 2)

This type covers the previous type and just shows where intermediate location reference points are placed.

| Byte number | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|---------------|------------------------|-------|--------------|-------------|--------------------|---------------|-------|-------|
| 1 | Status | RFU | RFU | 0 (no point) | 0 (no area) | 1 (has attributes) | 011 (version) | | |
| 2 | Longitude (1) | longitude highest byte | | | | | | | |
| 3 | Longitude (2) | longitude middle byte | | | | | | | |
| 4 | Longitude (3) | longitude lowest byte | | | | | | | |
| 5 | Latitude (1) | latitude highest byte | | | | | | | |
| 6 | Latitude (2) | latitude middle byte | | | | | | | |



| | | | | | |
|--------------|---------------|---------------------------------|--------------|-----------------------|-------------|
| 7 | Latitude (3) | latitude lowest byte | | | |
| 8 | Attribute 1 | RFU | RFU | functional road class | form of way |
| 9 | Attribute 2 | lowest frc next point | | bearing | |
| 10 | Attribute 3 | distance to next point | | | |
| 11 | Longitude (1) | relative longitude highest byte | | | |
| 12 | Longitude (2) | relative longitude lowest byte | | | |
| 13 | Latitude (1) | relative latitude highest byte | | | |
| 14 | Latitude (2) | relative latitude lowest byte | | | |
| 15 | Attributes 1 | RFU | RFU | functional road class | form of way |
| 16 | Attributes 2 | lowest frc next point | | bearing | |
| 17 | Attributes 3 | distance to next point | | | |
| ... | | | | | |
| 18 + 7*(i-4) | Longitude (1) | relative longitude highest byte | | | |
| 19 + 7*(i-4) | Longitude (2) | relative longitude lowest byte | | | |
| 20 + 7*(i-4) | Latitude (1) | relative latitude highest byte | | | |
| 21 + 7*(i-4) | Latitude (2) | relative latitude lowest byte | | | |
| 22 + 7*(i-4) | Attributes 1 | RFU | RFU | functional road class | form of way |
| 23 + 7*(i-4) | Attributes 2 | lowest frc next point | | bearing | |
| 24 + 7*(i-4) | Attributes 3 | distance to next point | | | |
| ... | | | | | |
| 18+7*(n-3) | Longitude (1) | relative longitude highest byte | | | |
| 19+7*(n-3) | Longitude (2) | relative longitude lowest byte | | | |
| 20+7*(n-3) | Latitude (1) | relative latitude highest byte | | | |
| 21+7*(n-3) | Latitude (2) | relative latitude lowest byte | | | |
| 22+7*(n-3) | Attributes 1 | RFU | RFU | functional road class | form of way |
| 23+7*(n-3) | Attributes 4 | RFU | 0/1 (posOff) | 0/1 (negOff) | bearing |
| 24+7*(n-3) | PosOffset | positive offset (optional) | | | |
| 25+7*(n-3) | NegOffset | negative offset (optional) | | | |

Table 33: Line location with n location reference points (n>2)

Note: $2 < i < n$

Data size: $16 + (n-2)*7 + [0/1/2]$ bytes

Note: The binary format version 2 defines bit 5 in the status byte (byte 1) as RFU and the bits 2 – 0 are set to 010. The offset values (bytes $24+7*(n-3)$ and $25+7*(n-3)$) use the equal distribution method.

8.1.3. Code size calculation

The code size of a line location reference depends on the number of LR-points included in the location reference. There must be at least two LR-points in the location reference. Also mandatory is the header with the status information. The following Table 34 shows a calculation of the message sizes depending on the number of LR-points.

| # LR-points | Message size |
|-------------|--|
| 2 | 16 bytes (+1 or +2 bytes offset, if included) |
| 3 | 23 bytes (+1 or +2 bytes offset, if included) |
| 4 | 30 bytes (+1 or +2 bytes offset, if included) |

| | |
|---------|---|
| 5 | 37 bytes (+1 or +2 bytes offset, if included) |
| 6 | 44 bytes (+1 or +2 bytes offset, if included) |
| 7 | 51 bytes (+1 or +2 bytes offset, if included) |
| 8 | 58 bytes (+1 or +2 bytes offset, if included) |
| ... | ... |
| n (n>1) | $1 + 9 + (n-2)*7 + 6$ bytes (+1 or +2 bytes offset, if included) |

Table 34: Message sizes depending on the number of LR-points

8.2 Point locations

Point locations are supported since binary data format 3.

8.2.1. Geo-coordinate

The simplest form of a point location is a single geo-coordinate.

| Byte number | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|---------------|------------------------|-------|-----------------|----------------|----------------------|---------------|-------|-------|
| 1 | Status | RFU | RFU | 1 (is point) | 0 (no area) | 0 (no attributes) | 011 (version) | | |
| 2 | Longitude (1) | longitude highest byte | | | | | | | |
| 3 | Longitude (2) | longitude middle byte | | | | | | | |
| 4 | Longitude (3) | longitude lowest byte | | | | | | | |
| 5 | Latitude (1) | latitude highest byte | | | | | | | |
| 6 | Latitude (2) | latitude middle byte | | | | | | | |
| 7 | Latitude (3) | latitude lowest byte | | | | | | | |

Table 35: Point location – geo-coordinate

Data size: 7 bytes

8.2.2. Point along line

A point along line is referenced by a line and an offset value (measured from the start of the line).

| Byte number | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|---------------|---------------------------------|---------------|-----------------------|----------------|-----------------------|---------------|-------|-------|
| 1 | Status | RFU | RFU | 1 (is point) | 0 (no area) | 1 (has attributes) | 011 (version) | | |
| 2 | Longitude (1) | longitude highest byte | | | | | | | |
| 3 | Longitude (2) | longitude middle byte | | | | | | | |
| 4 | Longitude (3) | longitude lowest byte | | | | | | | |
| 5 | Latitude (1) | latitude highest byte | | | | | | | |
| 6 | Latitude (2) | latitude middle byte | | | | | | | |
| 7 | Latitude (3) | latitude lowest byte | | | | | | | |
| 8 | Attribute 1 | orientation | | functional road class | | | form of way | | |
| 9 | Attribute 2 | lowest frc next point | | | bearing | | | | |
| 10 | Attribute 3 | distance to next point | | | | | | | |
| 11 | Longitude (1) | relative longitude highest byte | | | | | | | |
| 12 | Longitude (2) | relative longitude lowest byte | | | | | | | |
| 13 | Latitude (1) | relative latitude highest byte | | | | | | | |
| 14 | Latitude (2) | relative latitude lowest byte | | | | | | | |
| 15 | Attributes 1 | side of the road | | functional road class | | | form of way | | |
| 16 | Attributes 4 | RFU | 1 (posOff) | 0 (negOff) | bearing | | | | |
| 17 | PosOffset | positive offset | | | | | | | |

Table 36: Point location – Point along line

Data size: 17 bytes

8.2.3. Network node

A network node is a point along a network line without any offsets.

| Byte number | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|---------------|------------------------|---------------|-----------------------|---------------------------------|-----------------------|---------------|-------|-------|
| 1 | Status | RFU | RFU | 1 (is point) | 0 (no area) | 1 (has attributes) | 011 (version) | | |
| 2 | Longitude (1) | | | | longitude highest byte | | | | |
| 3 | Longitude (2) | | | | longitude middle byte | | | | |
| 4 | Longitude (3) | | | | longitude lowest byte | | | | |
| 5 | Latitude (1) | | | | latitude highest byte | | | | |
| 6 | Latitude (2) | | | | latitude middle byte | | | | |
| 7 | Latitude (3) | | | | latitude lowest byte | | | | |
| 8 | Attribute 1 | orientation | | functional road class | | | form of way | | |
| 9 | Attribute 2 | lowest frc next point | | | bearing | | | | |
| 10 | Attribute 3 | distance to next point | | | | | | | |
| 11 | Longitude (1) | | | | relative longitude highest byte | | | | |
| 12 | Longitude (2) | | | | relative longitude lowest byte | | | | |
| 13 | Latitude (1) | | | | relative latitude highest byte | | | | |
| 14 | Latitude (2) | | | | relative latitude lowest byte | | | | |
| 15 | Attributes 1 | side of the road | | functional road class | | | form of way | | |
| 16 | Attributes 4 | RFU | 0 (posOff) | 0 (negOff) | bearing | | | | |

Table 37: Point location – Network node

Data size: 16 bytes

8.2.4. Point of interest (POI) with access point

A point of interest with access point combines a point of interest located somewhere on the surface and an access point within a network.

| Byte number | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|---------------|--|---------------|-----------------------|---------------------------------|-----------------------|---------------|-------|-------|
| 1 | Status | RFU | RFU | 1 (is point) | 0 (no area) | 1 (has attributes) | 011 (version) | | |
| 2 | Longitude (1) | | | | longitude highest byte | | | | |
| 3 | Longitude (2) | | | | longitude middle byte | | | | |
| 4 | Longitude (3) | | | | longitude lowest byte | | | | |
| 5 | Latitude (1) | | | | latitude highest byte | | | | |
| 6 | Latitude (2) | | | | latitude middle byte | | | | |
| 7 | Latitude (3) | | | | latitude lowest byte | | | | |
| 8 | Attribute 1 | orientation | | functional road class | | | form of way | | |
| 9 | Attribute 2 | lowest frc next point | | | bearing | | | | |
| 10 | Attribute 3 | distance to next point | | | | | | | |
| 11 | Longitude (1) | | | | relative longitude highest byte | | | | |
| 12 | Longitude (2) | | | | relative longitude lowest byte | | | | |
| 13 | Latitude (1) | | | | relative latitude highest byte | | | | |
| 14 | Latitude (2) | | | | relative latitude lowest byte | | | | |
| 15 | Attributes 1 | side of the road | | functional road class | | | form of way | | |
| 16 | Attributes 4 | RFU | 1 (posOff) | 0 (negOff) | bearing | | | | |
| 17 | PosOffset | positive offset | | | | | | | |
| 18 | Longitude (1) | Relative longitude highest byte (poi coord, relative to abs lon) | | | | | | | |
| 19 | Longitude (2) | relative longitude lowest byte (poi coord, relative to abs lon) | | | | | | | |
| 20 | Latitude (1) | relative latitude highest byte (poi coord, relative to abs lon) | | | | | | | |
| 21 | Latitude (2) | relative latitude lowest byte (poi coord, relative to abs lon) | | | | | | | |

Table 38: Point location – Point of interest with access point

Data size: 21 bytes



8.2.5. Point of interest (POI) with access point at network node

This is the special case of a point of interest with access point but having no offset value so that the access point is a network node.

| Byte number | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|---------------|--|---------------|-----------------------|----------------|-----------------------|---------------|-------|-------|
| 1 | Status | RFU | RFU | 1 (is point) | 0 (no area) | 1 (has attributes) | 011 (version) | | |
| 2 | Longitude (1) | longitude highest byte | | | | | | | |
| 3 | Longitude (2) | longitude middle byte | | | | | | | |
| 4 | Longitude (3) | longitude lowest byte | | | | | | | |
| 5 | Latitude (1) | latitude highest byte | | | | | | | |
| 6 | Latitude (2) | latitude middle byte | | | | | | | |
| 7 | Latitude (3) | latitude lowest byte | | | | | | | |
| 8 | Attribute 1 | orientation | | functional road class | | | form of way | | |
| 9 | Attribute 2 | lowest frc next point | | | bearing | | | | |
| 10 | Attribute 3 | distance to next point | | | | | | | |
| 11 | Longitude (1) | relative longitude highest byte | | | | | | | |
| 12 | Longitude (2) | relative longitude lowest byte | | | | | | | |
| 13 | Latitude (1) | relative latitude highest byte | | | | | | | |
| 14 | Latitude (2) | relative latitude lowest byte | | | | | | | |
| 15 | Attributes 1 | side of the road | | functional road class | | | form of way | | |
| 16 | Attributes 4 | RFU | 0 (posOff) | 0 (negOff) | bearing | | | | |
| 17 | Longitude (1) | Relative longitude highest byte (poi coord, relative to abs lon) | | | | | | | |
| 18 | Longitude (2) | relative longitude lowest byte (poi coord, relative to abs lon) | | | | | | | |
| 19 | Latitude (1) | relative latitude highest byte (poi coord, relative to abs lon) | | | | | | | |
| 20 | Latitude (2) | relative latitude lowest byte (poi coord, relative to abs lon) | | | | | | | |

Table 39: Point location – Point of interest with access point at network node

Data size: 20 bytes

8.3 Identification of the location type

All location types need to be identified during decoding. This identification is based on the data size and the combination of the flags point, area and attributes. The Table 40 shows a mapping from location type to the possible data sizes and the settings of the flags. Table 41 shows a mapping from data size to the possible location types.

| Location type | Data size | Point flag | Area flag | Attributes flag |
|---|--------------------------------|------------|-----------|-----------------|
| Line | $16 + (n-2)*7 + [0/1/2]$ bytes | 0 | 0 | 1 |
| Geo-coordinate | 7 bytes | 1 | 0 | 0 |
| Along line | 17 bytes | 1 | 0 | 1 |
| Network node | 16 bytes | 1 | 0 | 1 |
| Point of interest with access point | 21 bytes | 1 | 0 | 1 |
| Point of interest with access point at network node | 20 bytes | 1 | 0 | 1 |

Table 40: Identification depending on flags

| Number of bytes | Location type |
|-----------------|-----------------|
| 1 | - not allowed - |
| 2 | - not allowed - |
| 3 | - not allowed - |
| 4 | - not allowed - |

| | |
|----|--|
| 5 | - not allowed - |
| 6 | - not allowed - |
| 7 | <i>Point location – geo-coordinate</i> |
| 8 | - not allowed - |
| 9 | - not allowed - |
| 10 | - not allowed - |
| 11 | - not allowed - |
| 12 | - not allowed - |
| 13 | - not allowed - |
| 14 | - not allowed - |
| 15 | - not allowed - |
| 16 | <i>Line location (2 LRP) or Network node</i> |
| 17 | <i>Line location (2 LRP, 1 offset) or Point along line</i> |
| 18 | <i>Line location (2 LRP, 2 offsets)</i> |
| 19 | - not allowed - |
| 20 | Point location with access point at network node |
| 21 | Point location with access point |
| 22 | - not allowed - |
| 23 | <i>Line location (3 LRP)</i> |
| 24 | <i>Line location (3 LRP, 1 offset)</i> |

Table 41: Identification depending on data size

8.4 Changes

The following lists summarize the changes of the binary physical format.

8.4.1. Version 2 -> Version 3

- Offsets are using the relative format (calculating the relation between the offset length and the location length)
- Version bits (0-2) in the status byte indicate version 3 = (011b)
 - Important note: the value 0 = (000b) is not used yet and might indicate in a later version of the format that a second status byte is used (representing a version number higher than 7!)
- Bit 5 of the status byte indicates whether or not the location is a point location (0: no point location, 1: point location)
- Bits 7 and 6 of the first attribute of the first location reference point are used for the orientation of a point location (only valid for point locations)
- Bits 7 and 6 of the first attribute of the last location reference point are used for the side of the road of a point location (only valid for point locations)



Chapter E

Physical data format (XML)

9. OpenLR™ - XML representation

This physical data format describes an XML format for the OpenLR™ data format. It uses the components described in the logical data format in section 5.

9.1 Data types

The physical data format uses the following data types. Table 42 gives an overview of all available data types and specifies the name and the type each data type. In the following sections the data type names are used to indicate the type for each data component.

| Data type name | Type |
|----------------|--|
| integer | The integer data type is used to specify a numeric value without a fractional component. |
| double | The double data type is used to specify a numeric value with a fractional component. |
| string | The string data type can contain characters, line feeds, carriage returns, space and tab characters. |
| DATA | The data type can hold any kind of XML data not specified by the schema. It allows documents to contain additional elements. |
| Base64Data | Binary data type to express binary-formatted data (Base64-encoded). |

Table 42: Physical format: Data types (XML)

9.2 Coordinates

Each point in a map consists of a coordinate pair “longitude” (lon) and “latitude” (lat) represented in WGS84 coordinates. The directions north and east are represented by positive values (longitude and latitude respectively).

The coordinate values will be transmitted as double values. The physical format makes use of absolute coordinates which represents the designated values of the geographical position. The longitude and latitude fields are mandatory.

```
<openlr:Coordinates>
  <openlr:Longitude> [double] </openlr:Longitude>
  <openlr:Latitude> [double] </openlr:Latitude>
</openlr:Coordinates>
```

9.3 Line attribute values

Line attributes are part of a location reference point and consists of functional road class (FRC), form of way (FOW) and bearing (BEAR) data. All components are mandatory fields.

```
<openlr:LineAttributes>
  <openlr:FRC> [FRC value] </openlr:FRC>
  <openlr:FOW> [FOW value] </openlr:FOW>
  <openlr:BEAR> [integer] </openlr:BEAR>
</openlr:LineAttributes>
```

9.3.1. Functional Road Class (FRC)

The functional road class (FRC) can hold eight different values as described in the logical format. These eight values are shown in Table 43.

```
<openlr:FRC> [FRC value] </openlr:FRC>
```

| Data type | Value | Description |
|-----------|-------|-------------------------------------|
| FRC value | FRC0 | Main road, highest importance |
| | FRC 1 | First class road |
| | FRC 2 | Second class road |
| | FRC 3 | Third class road |
| | FRC 4 | Fourth class road |
| | FRC 5 | Fifth class road |
| | FRC 6 | Sixth class road |
| | FRC 7 | Other class road, lowest importance |

Table 43: Physical format: Functional road class (XML)

9.3.2. Form of way (FOW)

The form of way (FOW) can hold eight different values as described in the logical format. These eight values are shown in Table 44.

```
<openlr:FOW> [FOW value] </openlr:FOW>
```

| Data type | Value |
|-----------|----------------------|
| FOW value | UNDEFINED |
| | MOTORWAY |
| | MULTIPLE_CARRIAGEWAY |
| | SINGLE_CARRIAGEWAY |
| | ROUNDBOUT |
| | TRAFFICSQUARE |
| | SLIPROAD |
| | OTHER |

Table 44: Physical format: Form of way (XML)

9.3.3. Bearing (BEAR)

The bearing describes the angle between the true North and the road as described in the logical format. The physical data format defines the bearing field as an integer value between 0 and 360 whereby "0" is included and "360" is excluded from that range. The values should be rounded if not available as an integer.

```
<openlr:BEAR> [integer] </openlr:BEAR>
```

9.4 Path attribute values

The field path attributes is part of a location reference point (except for the last location reference point) and consists of lowest functional road class (LFRCNP) and distance to next point (DNP) data. All components are mandatory fields.

```
<openlr:PathAttributes>
  <openlr:LFRCNP> [FRC value] </openlr:LFRCNP>
```

```
<openlr:DNP> [integer] </openlr:DNP>
</openlr:PathAttributes>
```

9.4.1. Distance to next LR-point (DNP)

The DNP attribute measures the distance in meters between two consecutive LR-points along the location reference path as described in the logical format.

```
<openlr:DNP> [integer] </openlr:DNP>
```

9.4.2. Lowest FRC to next point (LFRCNP)

The lowest FRC to the next point indicates the lowest functional road class used in the location reference path to the next LR-point. This information could be used to limit the number of road classes which need to be scanned during the decoding. See Table 43 for a definition of the data type.

```
<openlr:LFRCNP> [FRC value] </openlr:LFRCNP>
```

9.5 Point attribute values

9.5.1. Side of road

A point location may have additional information on which side of the road the point is located. Examples for such point locations are particular house numbers, speed cams or traffic signs. This additional information is valid for the types “point along line” and “poi with access point”.

```
<openlr:SideOfRoad> [SideOfRoad value] </openlr:SideOfRoad>
```

| SideOfRoad value | Meaning |
|--------------------|---|
| ON_ROAD_OR_UNKNOWN | Point is directly on (or above) the road, or determination of right/left side is not applicable (default) |
| RIGHT | Point is on right side of the road |
| LEFT | Point is on the left side of the road |
| BOTH | Point is on both sides of the road |

Table 45: Physical format: SideOfRoad values (XML)

9.5.2. Orientation

A point location may have additional information in which direction the information at that point is relevant. Examples for such point locations are speed cams or traffic signs. This additional information is valid for the types “point along line” and “poi with access point”.

```
<openlr:Orientation> [Orientation value] </openlr:Orientation>
```

| Orientation value | Meaning |
|---------------------------|--|
| NO_ORIENTATION_OR_UNKNOWN | Point has no sense of orientation, or determination of orientation is not applicable (default) |
| WITH_LINE_DIRECTION | Point has orientation from first LRP towards second LRP |



| | |
|------------------------|---|
| AGAINST_LINE_DIRECTION | Point has orientation from second LRP towards first LRP |
| BOTH | Point has orientation in both directions |

Table 46: Orientation of a point location

9.6 Line location reference

A line location reference describes a path within a map and consists of location reference point(s), a last location reference point and offset data. There must be at least one location reference point and exactly one last location reference point. The offset field is optional.

```
<openlr:LineLocationReference>
  [LocationReferencePoint(s)]
  [LastLocationReferencePoint]
  [Offsets]
</openlr:LineLocationReference>
```

9.6.1. Location reference point

A location reference point consists of coordinate, line attribute and path attribute data. All fields are mandatory.

```
<openlr:LocationReferencePoint>
  [Coordinate]
  [LineAttributes]
  [PathAttributes]
</openlr:LocationReferencePoint>
```

9.6.2. Last location reference point

A last location reference point consists of coordinate and line attribute data. All fields are mandatory.

```
<openlr>LastLocationReferencePoint>
  [Coordinate]
  [LineAttributes]
</openlr>LastLocationReferencePoint>
```

9.6.3. Offsets

Offsets are used to locate the start and end of a location more precisely than bounding to the nodes in a network. The logical format defines two offsets, one at the start of the location and one at the end of the location and both offsets operate along the lines of the location and are measured in meters. The offset values are optional and a missing offset value means an offset of 0 meters. If there are no offsets at all the offset field can be omitted.

```
<openlr:Offsets>
  <openlr:PosOff> [integer] </openlr:PosOff>
  <openlr:NegOff> [integer] </openlr:NegOff>
</openlr:Offsets>
```

9.7 Point location reference

A point location reference consists of exactly one out of GeoCoordinate, PointAlongLine or PoiWithaccessPoint.

```
<openlr:PointLocationReference>
  [GeoCoordinate]
  [PointAlongLine]
  [PoiWithAccessPoint]
</openlr:PointLocationReference>
```

9.7.1. GeoCoordinate

A GeoCoordinate consists of a single coordinate pair.

```
<openlr:GeoCoordinate>
  [Coordinate]
</openlr:GeoCoordinate>
```

9.7.2. PointAlongLine

A PointAlongLine consists of a location reference point, a last location reference point, optional offsets, and information about the side of the road and the orientation.

```
<openlr:PointAlongLine>
  [LocationReferencePoint]
  [LastLocationReferencePoint]
  [Offsets]
  <openlr:SideOfRoad> [SideOfRoad] </openlr:SideOfRoad>
  <openlr:Orientation> [Orientation] </openlr:Orientation>
</openlr:PointAlongLine>
```

9.7.3. PoiWithAccessPoint

A PoiWithAccessPoint consists of a location reference point, a last location reference point, a coordinate, optional offsets, and information about the side of the road and the orientation.

```
<openlr:PointAlongLine>
  [LocationReferencePoint]
  [LastLocationReferencePoint]
  [Coordinate]
  [Offsets]
  <openlr:SideOfRoad> [SideOfRoad] </openlr:SideOfRoad>
  <openlr:Orientation> [Orientation] </openlr:Orientation>
</openlr:PointAlongLine>
```

9.8 XML location reference

A XML location reference currently consists of a line location or point location reference. This inner field is mandatory.

```
<openlr:XMLLocationReference>
  <openlr:LineLocationReference> [LineLocationReference]
  </openlr:LineLocationReference>
  <openlr:PointLocationReference> [PointLocationReference]
  </openlr:PointLocationReference>
</openlr:XMLLocationReference>
```

9.9 Binary location references

The field “binary location references” holds the binary representation of the same location reference. There may exist several different versions of the binary format but at least one binary location reference is mandatory.

```
<openlr:BinaryLocationReferences>
  [BinaryLocationReference(s)]
</openlr:BinaryLocationReferences>
```

9.9.1. Binary location reference

A binary location reference represents the binary format of the same location reference as described in this XML representation. This field contains the Base64-encoded binary data. The attributes id and version are mandatory whereby the id identifies this specific binary location reference in the list of possibly several binary location references and the version expresses the binary format version being used.

```
<openlr:BinaryLocationReference id="[name]" version="[version]">
  [Base64Data]
</openlr:BinaryLocationReference>
```

9.10 Additional information

The additional information field consists of extra information not being relevant for the OpenLR™ location reference data. This field is optional and also all inner fields are optional. If no additional data is specified this field can be omitted. This field consists of bounding box, map database and any other kind of XML data.

```
<openlr:AdditionalInformation>
  <openlr:BoundingBox> [BoundingBox] </openlr:BoundingBox>
  <openlr:MapDatabase> [string] </openlr:MapDatabase>
  <openlr:AdditionalData> [DATA] </openlr:AdditionalData>
</openlr:AdditionalInformation>
```

9.10.1. Bounding box

The (optional) bounding box describes a rectangle which covers the location reference path completely. The location reference path is the path defined by the location reference points so that this path might be longer than the location. The bounding box is defined by the coordinates of its lower left and upper right corner. Both coordinates are mandatory fields and consists of a longitude and latitude pair.

```
<openlr:BoundingBox>
  <openlr:LowerLeft>
    <openlr:Longitude> [double] </openlr:Longitude>
    <openlr:Latitude> [double] </openlr:Latitude>
  </openlr:LowerLeft>
  <openlr:UpperRight>
    <openlr:Longitude> [double] </openlr:Longitude>
    <openlr:Latitude> [double] </openlr:Latitude>
  </openlr:UpperRight>
</openlr:BoundingBox>
```



9.10.2. Map database

The (optional) map database field provides information about the map being used during decoding. The information can be of any type but must be expressible as a string.

```
<openlr:MapDatabase> [string] </openlr:MapDatabase>
```

9.10.3. Additional data

The (optional) additional data field makes the XML format extendable with any kind of other XML data not specified in the XML format schema definition. This field may be used to transmit additional data like debug data or decoding hints.

```
<openlr:AdditionalData> [DATA] </openlr:AdditionalData>
```

9.11 Root tag: OpenLR

The root tag of the XML format is OpenLR. This tag includes the namespace definitions and schema locations. The tag consists of a location id and XML location reference, binary location reference and additional information data. The location id is mandatory; all other data fields are optional. But there should be at least one of the data fields XML location reference or binary location references.

```
<?xml version="1.0" encoding="UTF-8"?>
<openlr:OpenLR xmlns:openlr=http://www.openlr.org/openlr
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openlr.org/openlr openlr.xsd ">
  [LocationID]
  [XMLLocationReference]
  [BinaryLocationReferences]
  [AdditionalInformation]
</openlr:OpenLR>
```

9.11.1. Location ID

The location id identifies this location reference uniquely. The identifier can be of any type but must be expressible as a string.

```
<openlr:LocationID> [string] </openlr:LocationID>
```

10. OpenLR™ - Physical data format XML specification

The XML format for OpenLR™ location references is defined by the following XML schema definition. The schema includes additional documentation.

Figure 19 shows the structure of such XML documents in a graphical way.

```

<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:openlr="http://www.openlr.org/openlr" attributeFormDefault="unqualified"
  elementFormDefault="qualified" targetNamespace="http://www.openlr.org/openlr">

  <xs:annotation>
    <xs:documentation xml:lang="en">
      This schema describes the XML format for OpenLR location references. This format includes a
      XMLLocationReference but also supports BinaryLocationReference. The XMLLocationReference shall make use of
      the raw data for any type so that no compression techniques are used (e.g. use of intervals for the description of
      length values). The BinaryLocationReference shall represent the compressed format of the location reference. For
      further information on the relevant fields see the OpenLR white paper at http://www.openlr.org.

      OpenLR is a trade mark of TomTom International B.V.
      author: TomTom International B.V.
    </xs:documentation>
  </xs:annotation>

  <xs:simpleType name="FRC_Type">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        The functional road class is a road classification based on the importance of a road.
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="FRC0" />
      <xs:enumeration value="FRC1" />
      <xs:enumeration value="FRC2" />
      <xs:enumeration value="FRC3" />
      <xs:enumeration value="FRC4" />
      <xs:enumeration value="FRC5" />
      <xs:enumeration value="FRC6" />
      <xs:enumeration value="FRC7" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="FOW_Type">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        The form of way describes the physical road type.
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNDEFINED" />
      <xs:enumeration value="MOTORWAY" />
      <xs:enumeration value="MULTIPLE_CARRIAGEWAY" />
      <xs:enumeration value="SINGLE_CARRIAGEWAY" />
      <xs:enumeration value="ROUNDBABOUT" />
      <xs:enumeration value="TRAFFICSQUARE" />
      <xs:enumeration value="SLIPROAD" />
      <xs:enumeration value="OTHER" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="Bearing_Type">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        The bearing describes the angle between the true North and a line. The values range from 0 to 359 degree.
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0" />
      <xs:maxExclusive value="360" />
    </xs:restriction>
  </xs:simpleType>

```



```

</xs:simpleType>

<xs:simpleType name="SideOfRoad_Type">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The side of the road is valid only for point locations and indicates on which side of the referenced road the point
      location is located.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="ON_ROAD_OR_UNKNOWN" />
    <xs:enumeration value="RIGHT" />
    <xs:enumeration value="LEFT" />
    <xs:enumeration value="BOTH" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="Orientation_Type">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The orientation is only valid for point locations and indicates for which direction the point information is relevant.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="NO_ORIENTATION_OR_UNKNOWN" />
    <xs:enumeration value="WITH_LINE_DIRECTION" />
    <xs:enumeration value="AGAINST_LINE_DIRECTION" />
    <xs:enumeration value="BOTH" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="LineAttributes">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Line attributes describe features of a line in a network. The attributes the functional road class, form of way and
      the bearing of a line.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="FRC" type="openlr:FRC_Type" />
    <xs:element name="FOW" type="openlr:FOW_Type" />
    <xs:element name="BEAR" type="openlr:Bearing_Type" />
  </xs:all>
</xs:complexType>

<xs:complexType name="PathAttributes">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Path attributes describes features of a path. The LFR CNP attribute is the lowest functional road class value
      which appears in the path. The highest FRC value is FRC0 and the lowest possible FRC value is FRC7.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="LFR CNP" type="openlr:FRC_Type" />
    <xs:element name="DNP" type="xs:integer" />
  </xs:all>
</xs:complexType>

<xs:complexType name="Coordinates">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Coordinates stands for a pair of WGS84 longitude and latitude values. This coordinate pair specifies a
      geometric point in a digital map.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="Longitude" type="xs:double" />
    <xs:element name="Latitude" type="xs:double" />
  </xs:all>
</xs:complexType>

<xs:complexType name="BoundingBox">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The bounding box describes a rectangle which covers the location completely. The rectangle is defined by the
      coordinates of the lower left and upper right corners.
    </xs:documentation>
  </xs:annotation>

```



```

</xs:documentation>
</xs:annotation>
<xs:all>
  <xs:element name="LowerLeft" type="openlr:Coordinates" />
  <xs:element name="UpperRight" type="openlr:Coordinates" />
</xs:all>
</xs:complexType>

<xs:complexType name="Offsets">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Offsets are used to shorten the location reference path at its start and end. The new positions along the location
      reference path indicate the real start and end of the location. The positive offset is the difference of the start
      point of the location reference and the start point of the desired location along the location reference path. The
      negative offset is the difference of the end point of the desired location and the end point of the location
      reference along the location reference path. Both values are measured in meter.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="0" name="PosOff" type="xs:integer" />
    <xs:element minOccurs="0" name="NegOff" type="xs:integer" />
  </xs:all>
</xs:complexType>

<xs:complexType name="AdditionalData">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      AdditionalData can be used to add any other kind of information needed like debug information or additional
      encoder/decoder flags.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AdditionalInformation">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      AdditionalInformation can be added if necessary. This includes the bounding box covering the location and
      information on the map used for encoding. The AdditionalData section also provides the freedom to add any
      other information needed.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="0" name="BoundingBox" type="openlr:BoundingBox" />
    <xs:element minOccurs="0" name="MapDatabase" type="xs:string" />
    <xs:element minOccurs="0" name="AdditionalData" type="openlr:AdditionalData" />
  </xs:all>
</xs:complexType>

<xs:complexType name="LocationReferencePoint">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The basis of a location reference is a sequence of location reference points (LRPs). Such a LRP contains a
      coordinate pair, specified in WGS84 longitude and latitude values and additionally several line and path
      attributes.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="1" name="Coordinates" type="openlr:Coordinates" />
    <xs:element minOccurs="1" name="LineAttributes" type="openlr:LineAttributes" />
    <xs:element minOccurs="1" name="PathAttributes" type="openlr:PathAttributes" />
  </xs:all>
</xs:complexType>

<xs:complexType name="LastLocationReferencePoint">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The sequence of location reference points is terminated by a last location reference point. This LRP does not
      need to have any path attributes.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="1" name="Coordinates" type="openlr:Coordinates" />

```

```

<xs:element minOccurs="1" name="LineAttributes" type="openlr:LineAttributes" />
</xs:all>
</xs:complexType>

<xs:complexType name="BinaryLocationReference">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The BinaryLocationReference is the Base64-encoded binary representation of the location reference. Each
      binary location reference needs to have an identifier and a version. The version is defined by the version of the
      binary physical format.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:base64Binary">
      <xs:attribute name="id" type="xs:string" use="required" />
      <xs:attribute name="version" type="xs:string" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="LineLocationReference">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      A LineLocationReference is defined by an ordered sequence of location reference points and a terminating last
      location reference point.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded"
      name="LocationReferencePoint" type="openlr:LocationReferencePoint" />
    <xs:element minOccurs="1" maxOccurs="1"
      name="LastLocationReferencePoint" type="openlr>LastLocationReferencePoint" />
    <xs:element minOccurs="0" name="Offsets" type="openlr:Offsets" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GeoCoordinate">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      GeoCoordinate is a point location which consists of exactly one oordinate pair.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="1" maxOccurs="1" name="Coordinates"
      type="openlr:Coordinates" />
  </xs:all>
</xs:complexType>

<xs:complexType name="PointAlongLine">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Point along line is a point location which is defined by a line and an offset value. The line will be referenced by two
      location reference points and the concrete position on that line is referenced using the positive offset. Additionally
      information about the side of the road where the point is located and the orientation with respect to the direction of
      the line can be added.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="1" maxOccurs="1"
      name="LocationReferencePoint" type="openlr:LocationReferencePoint" />
    <xs:element minOccurs="1" maxOccurs="1"
      name="LastLocationReferencePoint" type="openlr>LastLocationReferencePoint" />
    <xs:element minOccurs="0" maxOccurs="1" name="Offsets"
      type="openlr:Offsets" />
    <xs:element minOccurs="0" maxOccurs="1" name="SideOfRoad"
      type="openlr:SideOfRoad_Type" default="ON_ROAD_OR_UNKNOWN" />
    <xs:element minOccurs="0" maxOccurs="1" name="Orientation"
      type="openlr:Orientation_Type" default="NO_ORIENTATION_OR_UNKNOWN" />
  </xs:all>
</xs:complexType>

<xs:complexType name="PoiWithAccessPoint">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Point along line with access is a point location which is defined by a line, an offset value and a coordinate. The

```



line will be referenced by two location reference points and the concrete position of the access point on that line is referenced using the positive offset. The point of interest is identified by the coordinate pair. Additionally information about the side of the road where the point is located and the orientation with respect to the direction of the line can be added.

```

</xs:documentation>
</xs:annotation>
<xs:all>
  <xs:element minOccurs="1" maxOccurs="1"
    name="LocationReferencePoint" type="openlr:LocationReferencePoint" />
  <xs:element minOccurs="1" maxOccurs="1"
    name="LastLocationReferencePoint" type="openlr>LastLocationReferencePoint" />
  <xs:element minOccurs="0" maxOccurs="1" name="Offsets"
    type="openlr:Offsets" />
  <xs:element minOccurs="1" maxOccurs="1" name="Coordinates"
    type="openlr:Coordinates" />
  <xs:element minOccurs="0" maxOccurs="1" name="SideOfRoad"
    type="openlr:SideOfRoad_Type" default="ON_ROAD_OR_UNKNOWN" />
  <xs:element minOccurs="0" maxOccurs="1" name="Orientation"
    type="openlr:Orientation_Type" default="NO_ORIENTATION_OR_UNKNOWN" />
</xs:all>

</xs:complexType>

<xs:complexType name="PointLocationReference">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      A PointLocationReference represents a point of interest. The point may be bound to the underlying network.
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element minOccurs="1" maxOccurs="1" name="GeoCoordinate"
      type="openlr:GeoCoordinate" />
    <xs:element minOccurs="1" maxOccurs="1" name="PointAlongLine"
      type="openlr:PointAlongLine" />
    <xs:element minOccurs="1" maxOccurs="1"
      name="PoiWithAccessPoint" type="openlr:PoiWithAccessPoint" />
  </xs:choice>
</xs:complexType>

<xs:complexType name="BinaryLocationReferences">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      It may exist several different formats for binary location references and all formats or a subset can be included.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded"
      name="BinaryLocationReference" type="openlr:BinaryLocationReference" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="XMLLocationReference">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      A XMLLocationReference describes an OpenLR location reference. Currently only line and point locations are
      defined but other location types may follow.
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element minOccurs="1" maxOccurs="1"
      name="LineLocationReference" type="openlr:LineLocationReference" />
    <xs:element minOccurs="1" maxOccurs="1"
      name="PointLocationReference" type="openlr:PointLocationReference" />
  </xs:choice>
</xs:complexType>

<xs:element name="OpenLR">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      An OpenLR location reference consists of a unique identifier and a location reference as a
      XMLLocationReference, BinaryLocationReference or both. The schema accepts OpenLR tags without any
      location reference but these tags will be ignored. The XMLLocationReference shall make use of the raw data for
      any type so that no compression techniques are used (e.g. use of intervals for the description of length values).
      The BinaryLocationReference shall represent the compressed format of the location reference. If both types are
      used the location reference must represent the same location. Additional information may be added if
      necessary.
    </xs:documentation>
  </xs:annotation>

```



```
</xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="LocationID"
      type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1"
      name="XMLLocationReference" type="openlr:XMLLocationReference" />
    <xs:element minOccurs="0" maxOccurs="1"
      name="BinaryLocationReferences" type="openlr:BinaryLocationReferences" />
    <xs:element minOccurs="0" maxOccurs="1"
      name="AdditionalInformation" type="openlr:AdditionalInformation" />
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

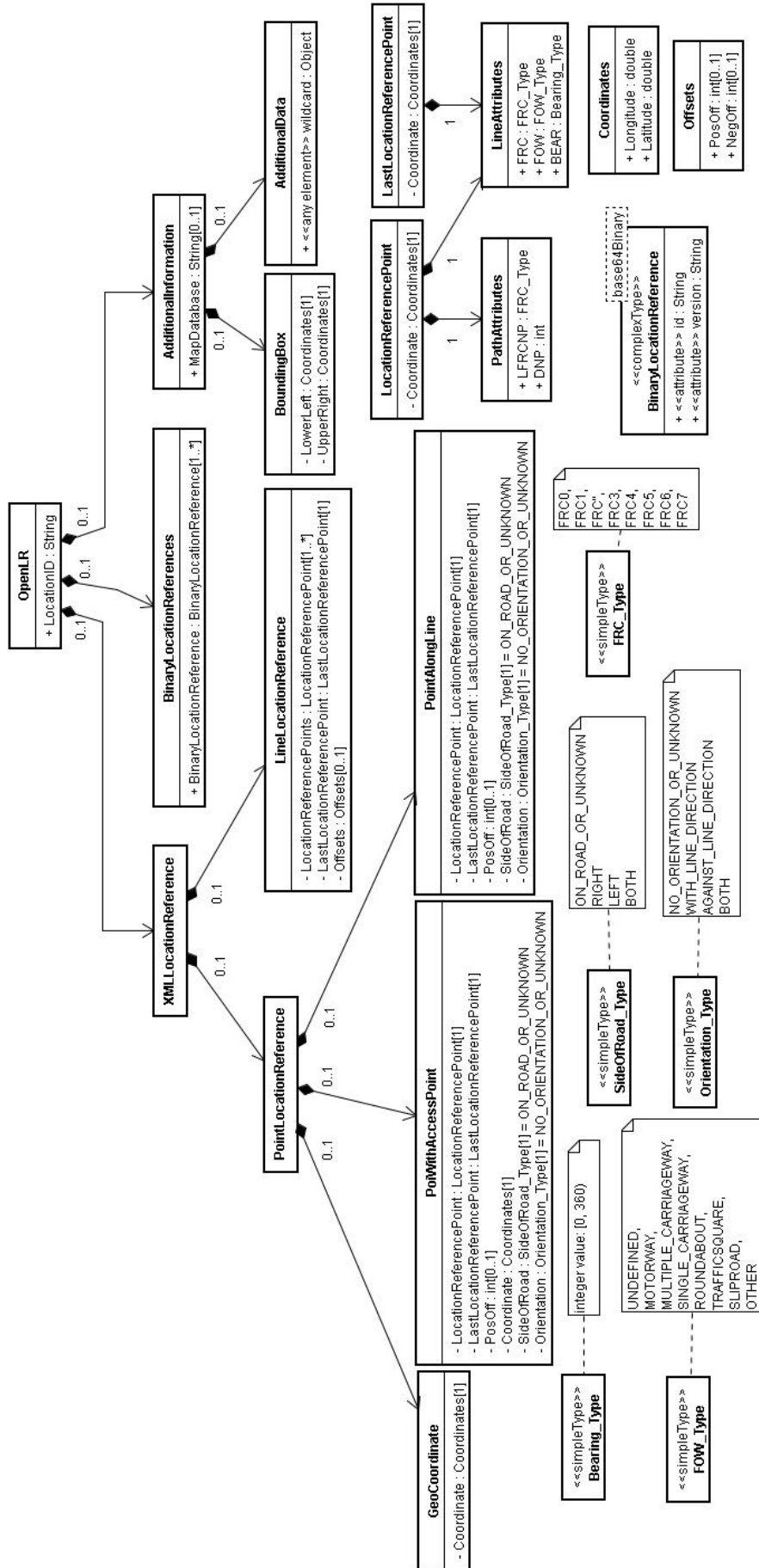


Figure 19: XML structure overview



Chapter F

Encoder



11. Encoding locations

The OpenLR™ encoder generates a map-independent location reference for a (map-dependent) location. The output of this encoder can be used to distribute location information to other parties. The decoder side will be able to resolve the location on its own map. If the encoding of such a location fails and no location reference can be created then the encoder should report on this.

Note:

This section provides recommendations how an OpenLR™ encoder might be implemented. There might be other implementations which are also compliant with the OpenLR™ standard.

11.1 Encoding steps

The following encoding steps describe the process of generating a location reference for a location. The location may consist of the following elements:

- An ordered list of connected lines and offset values (line location)
- A single geo-coordinate (point location – Geo-Coordinate)
- A single line and one offset value (point location – PointAlongLine)
- A single line, one offset value and a geo-coordinate (point location – PoiWithAccessPoint).

The PointAlongLine and PoiWithAccessPoint locations may also have given side of road and orientation information. The location needs to be checked whether it is valid for encoding or not. The following steps act as a guideline for the encoding process.

Step – 1 *Check validity of the location and offsets to be encoded.*

A line location is valid if the following constraints are fulfilled:

- The location is a connected path.
- The location is traversable from its start to its end.

A point location is valid if the following constraints are fulfilled:

- The coordinate values must be valid (if a coordinate is present)
 - boundaries for the longitude value: $-180^\circ \leq \text{lon} \leq 180^\circ$
 - boundaries for the latitude value: $-90^\circ \leq \text{lat} \leq 90^\circ$
- The location contains only a single line

For all location types the offsets must fulfill the following constraints:

- The sum of the positive and negative offset cannot be greater than the total length of the location lines.
- Positive offset value shall be less than the length of the first line
 - Otherwise the first line can be removed from the list of location lines and the offset value must be reduced in the same way
 - This procedure shall be repeated until this constraint is fulfilled
- Negative offset value shall be less than the length of the last line
 - Otherwise the last line can be removed from the list of location lines and the offset value must be reduced in the same way
 - This procedure shall be repeated until this constraint is fulfilled

Optionally the location can be checked against turn restrictions. This check can only be made if turn restriction information is supported by the map and if this check is enabled. Checking of turn restrictions is only made in the location validation check and does not affect the location generation phase.

If it is intended to use the binary physical format this step should additionally calculate the maximum (minimum) latitude values along the location and adjust the maximum distance between two LR-

points in Rule – 1, if necessary. The value defined in Rule – 1 is not applicable for locations above the latitude value 65.70° (or below the latitude value -65.70°).

If the location is not valid the encoder should fail.

Step – 2 *Adjust start and end node of the location to represent valid map nodes.*

If the location is a Geo-Coordinate location then go to step 10. In all other cases Rule – 4 of the data format rules recommends to place location reference points on valid nodes. Valid nodes are such nodes where a shortest-path calculation needs to decide between several different ways. Otherwise invalid nodes are such nodes where a shortest path calculation can step over (see also section 6).

Since the start and end of a location will become a location reference point these nodes need to be adjusted to valid nodes if necessary (expansion of location). The expansion shall take Rule – 1 (see section 6) into account so that the maximum distance between two location reference points will not be exceeded.

For line locations the real start of the location can then be referenced using offsets (positive offset for the start node and negative offset for the end node, respectively). For point locations start and end of the line shall be expanded but only the positive offset value needs to be adjusted. See also section 11.3 for a detailed explanation of the composition of offset values.

If adjusting the nodes to valid nodes fails then the encoder could proceed with the next steps and does not need to fail.

Step – 3 *Determine coverage of the location by a shortest-path (line locations only).*

If the encoder deals with a point location then go to step 8 otherwise calculate a shortest-path between the current start line and the end line of the location. This step will be executed until a coverage of the location has been found.

If no intermediate location reference point was detected so far, the current start line is identical to the start line of the location. If an intermediate location reference point was detected in a previous step, then the line corresponding to the intermediate location reference point acts as current start line. The start line is always part of the location.

The shortest path algorithm should take the whole network or a well-defined subset of the network into account in order to calculate a shortest path between the current start and end. Additionally it should fulfill the following constraints:

- All lengths of the lines should be measured in meters and should also be converted to integer values, so that float values need to be rounded correctly.
- The search is node based and starts at the start node of the first line and ends at the end node of the last line.
- The algorithm shall return an ordered list of lines representing the calculated shortest-path.

If no shortest-path can be calculated the encoding should fail.

Step – 4 *Check whether the calculated shortest-path covers the location completely. Go to step 5 if the location is not covered completely, go to step 7 if the location is covered.*

Compare the calculated shortest-path against the part of the location which is currently not covered by a shortest path in a previous step. The shortest-path covers the location if the following constraints are fulfilled:

- Every line of the location (between current start and end) needs to be part of the shortest-path.
- The order of the lines in the location (between current start and end) needs to be the same order as the lines in the shortest-path.

If one of the constraints is not fulfilled go to step 5 in order to determine a proper intermediate location reference point. If all constraints are fulfilled then go to step 7.

Step – 5 *Determine the position of a new intermediate location reference point so that the part of the location between the start of the shortest-path calculation and the new intermediate is covered completely by a shortest-path.*

If the location (between current start and end) is not fully part of the shortest-path or the order of the lines is mixed up then a proper intermediate location reference point needs to be determined. This intermediate must fulfill the following constraints:

- The shortest-path between the current start and the line indicated by the intermediate location reference point must cover the corresponding part of the location completely.
- The start node of the line indicated by the intermediate location reference point shall be positioned on a valid node (if no valid node can be determined, an invalid node may be chosen).

Note: The OpenLR™ team provides a separate document which describes the handling of such intermediate location reference points and also the handling of loops within the location. This additional document is not part of the OpenLR™ standard.

Step – 6 *Go to step 3 and restart shortest path calculation between the new intermediate location reference point and the end of the location.*

The remaining part of the location still needs to be covered by a shortest-path. Go to step 3 and calculate a shortest-path between the last intermediate found and the end of the location.

Step – 7 *Concatenate the calculated shortest-path(s) for a complete coverage of the location and form an ordered list of location reference points (from the start to the end of the location).*

The calculated shortest-path(s) need(s) to be concatenated so that the resulting location reference path covers the location completely. According to the concatenation of the path an ordered list of the location reference points needs to be formed, starting with the start of the location, followed by all intermediates, ending with the end of the location.

All relevant attributes for a location reference point need to be collected and calculated. This includes the functional road class, the form of way, the bearing of the corresponding line, and the distances and lowest functional road class between two subsequent location reference points.

This step may also include a check whether an intermediate LR-point is required to create a unique location reference. If the encoder uses a heuristic function during the route search it might happen that this heuristic leads the search to the end of the location but not along the location itself. The decoder might also use a heuristic function but this will lead the search towards the end of the recent part of the location and not to the end of the whole location. This difference might end up in intermediate LR-points added by the encoder but not required by the decoder. An intermediate LR-point check might determine these unneeded LR-points and remove them.

Step – 8 *Check validity of the location reference path. If the location reference path is invalid then go to step 9, if the location reference path is valid then go to step 10.*

The location reference as a list of ordered location reference points is checked for fulfilling the following constraints:

- The distance between two subsequent location reference points shall not exceed the maximum distance value as defined in the data format rules (see 6).
- The offset values must match the path between the first two (last two) location reference points. These values cannot be longer than the length of the path between the corresponding LR-points.

If the location reference does not fulfill all constraints then go to step 9, otherwise go to step 10.



A point location reference shall contain only two location reference points and for any other location type the location reference may contain additional location reference points if necessary. The encoder may detect situations where it might be difficult for a decoder to choose the correct path. In this case additional intermediate location reference points are allowed. The intermediate must fulfill the same constraints as defined in step 5.

Step – 9 *Add a sufficient number of additional intermediate location reference points if the distance between two location reference points exceeds the maximum distance. Remove the start (end) LR-point if the positive (negative) offset value exceeds the length of the corresponding path.*

In order to fulfill the constraints in step 8 additional intermediate location reference points need to be added or removed.

If the maximum distance between two subsequent location reference points is exceeded additional location reference points shall be placed at valid nodes along the location reference path between these two location reference points. If placing on valid nodes is not possible an invalid node shall be used. Examples for such a case are lines which are longer than 15 km. In such a case the coordinates of additional LRP may be resolved from a point being less than 15 km away from the start node of that line. Such a point is directly on the line and is not represented by a node. All LRP attributes need to be calculated relative to this point. The calculated attributes need to be updated and the new intermediate location reference point must be added to the list without breaking the order from start to end.

If the positive (negative) offset value is greater than the length of the path between the first two (last two) LR-points, then the first (last) LR-point needs to be removed from the list of LR-points. The offset value needs to be adjusted accordingly.

The process of adding/removing LR-points needs to be repeated until all constraints in step 8 are fulfilled.

Step – 10 *Create physical representation of the location reference.*

The calculated internal (raw) data needs to be transformed into a physical representation. The raw data may consist of:

- Ordered list of location reference points (all types, except GeoCoordinate)
- Offset information (all types, except GeoCoordinate)
 - Offsets need to be encoded according to the physical format and the physical format version
- Coordinates (GeoCoordinate and PoiWithAccessPoint)
- SideOfRoad and Orientation information (PointAlongLine and PoiwithAccessPoint)

11.2 Overview of the encoding steps

The following Table 47 summarizes the encoding rules and the following figures outline the workflow for GeoCoordinate locations (Figure 20), PointAlongLine and PoiWithAccessPoint locations (Figure 21) and Line locations (Figure 22).

| Step | Action |
|-----------------|--|
| Step – 1 | <i>Check validity of the location and offsets to be encoded</i> |
| Step – 2 | <i>Adjust start and end node of the location to represent valid map nodes</i> |
| Step – 3 | <i>Determine coverage of the location by a shortest-path (line locations only)</i> |
| Step – 4 | <i>Check whether the calculated shortest-path covers the location completely or. Go to step 5 if the location is not covered completely, go to step 7 if the location is covered</i> |

| | |
|------------------|---|
| Step – 5 | <i>Determine the position of a new intermediate location reference point so that the part of the location between the start of the shortest-path calculation and the new intermediate is covered completely by a shortest-path.</i> |
| Step – 6 | <i>Go to step 3 and restart shortest path calculation between the new intermediate location reference point and the end of the location.</i> |
| Step – 7 | <i>Concatenate the calculated shortest-path(s) for a complete coverage of the location and form an ordered list of location reference points (from the start to the end of the location)</i> |
| Step – 8 | <i>Check validity of the location reference path. If location reference path is invalid then go to step 9, if location reference path is valid then go to step 10.</i> |
| Step – 9 | <i>Add a sufficient number of additional intermediate location reference points if the distance between two location reference points exceeds the maximum distance.</i> |
| Step – 10 | <i>Create physical representation of the location reference</i> |

Table 47: Encoding steps overview

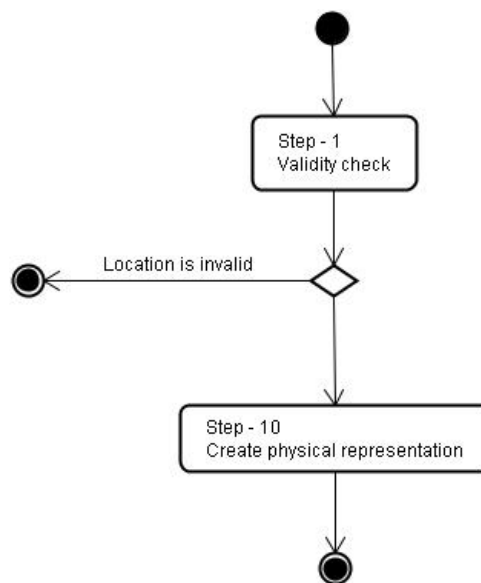


Figure 20: Workflow – OpenLR™ encoding (GeoCoordinate locations)

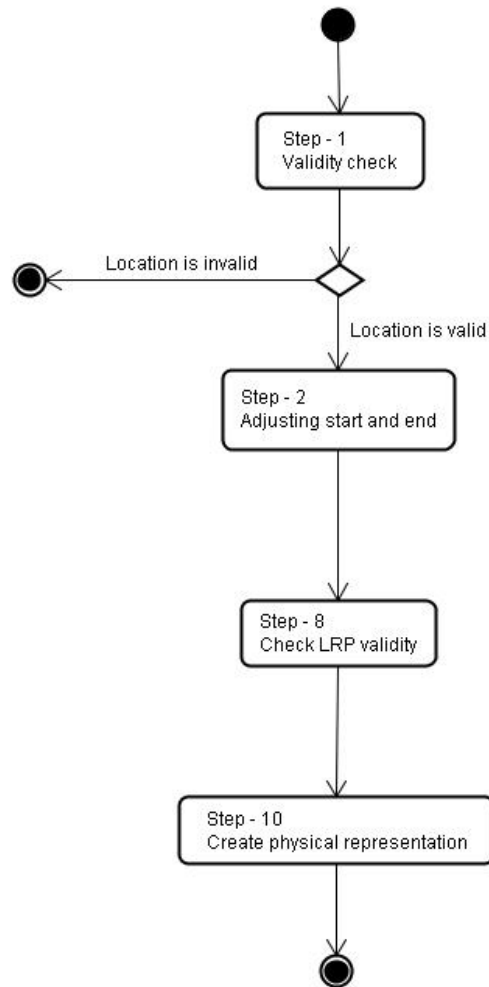


Figure 21: Workflow – OpenLR™ encoding (PointAlongLine and PoiWithAccessPoint locations)

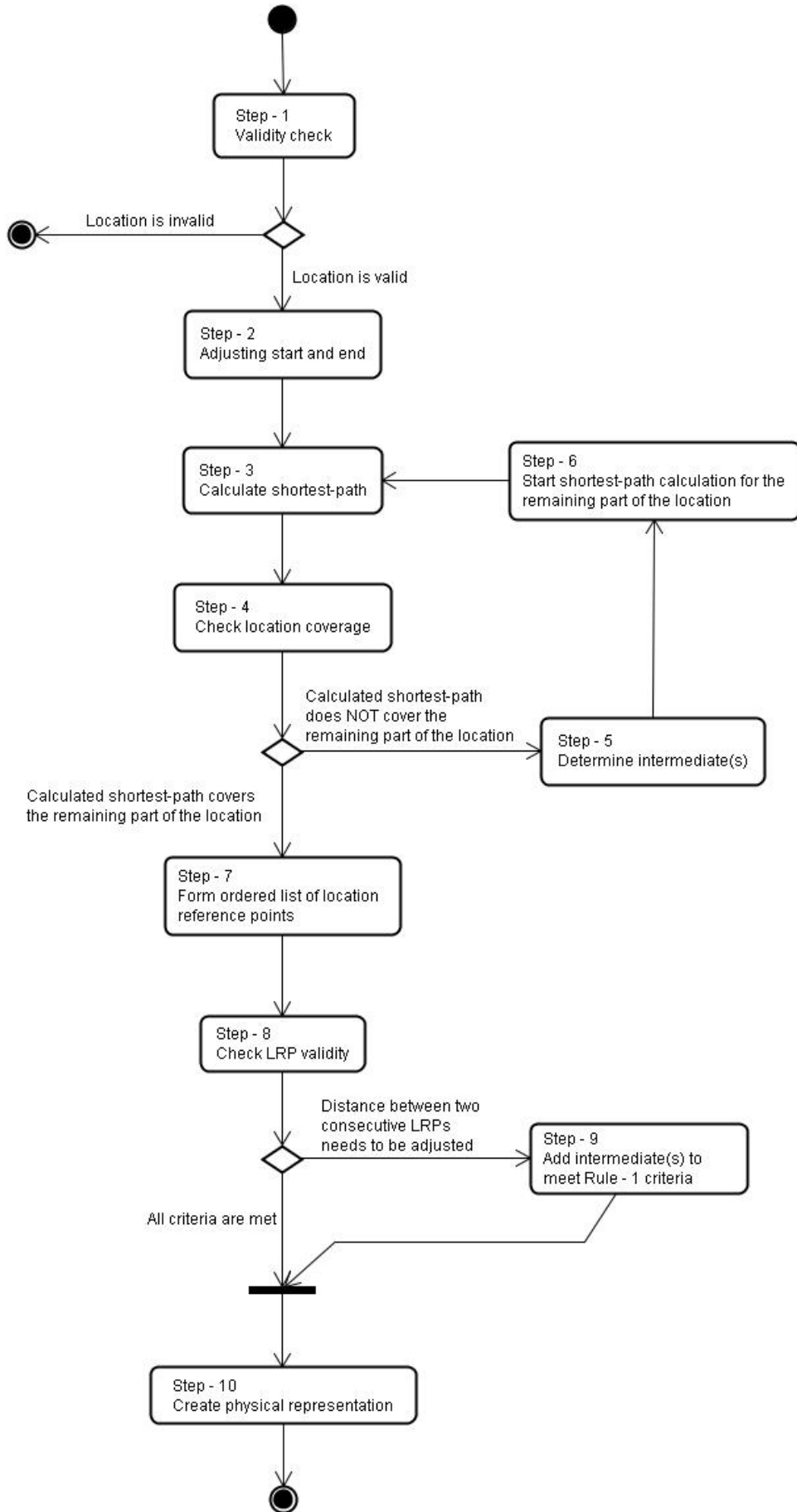


Figure 22: Workflow – OpenLR™ encoding (Line locations)

11.3 Composition of offset values

The following example shows how the offset values are being composed. Figure 23 shows a part of an abstract network with only 6 nodes and 5 lines. The first and the last node shall be valid nodes according to the definition of valid nodes. All intermediate nodes shall be invalid and therefore these nodes shall not be used for location reference points.

The situation a) shows the line location in red. The location is not bounded by any node. The user needs to choose the blue lines in situation b) as location lines. These location lines need to be encoded. The differences between the start node and the start of the location and the end of the location and the end node are the location offsets. Situation c) shows the positive offset at the start in dark green and the negative offset at the end in light green.

The location lines and the location offsets form the input for the OpenLR™ encoder. The encoding process starts with validating the location and if the location is valid then the start and end of the location lines shall use valid nodes. The location lines will be extended so far that the start and end nodes are valid and this step results in situation d). The yellow lines indicate the expanded location lines. The additional lines are shown in situation e) where the positive expansion offset is shown in dark brown and the negative expansion offset is shown in light brown.

The expanded location will be encoded and together with the concatenated offsets compiled into a physical format. The concatenated offsets are shown in situation f) where orange indicates the positive offset and purple the negative offset.

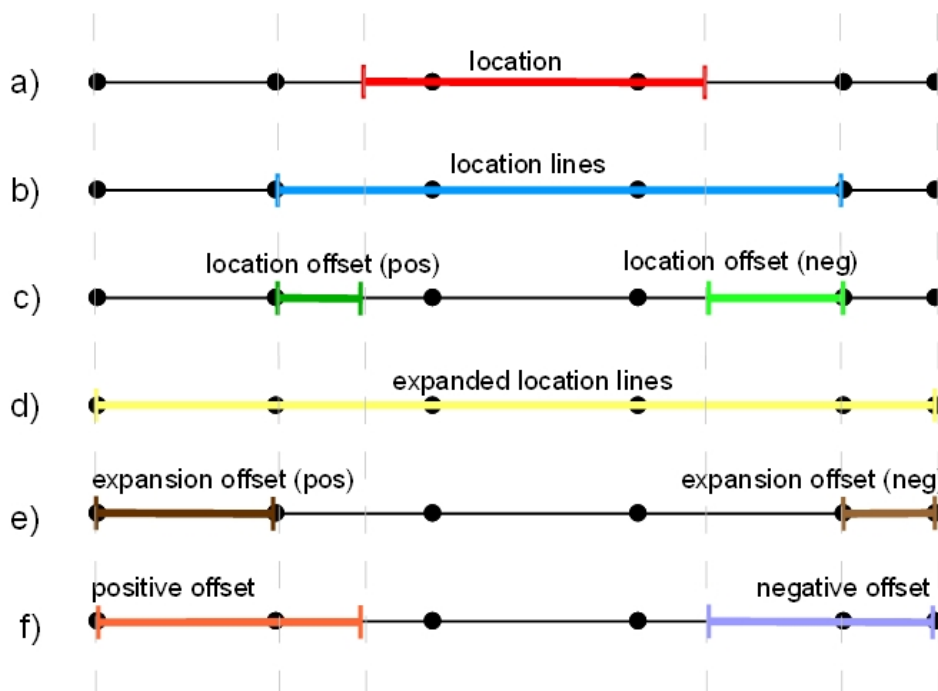


Figure 23: Offset calculation

Note:

In order to ensure a high accuracy for the offset values with the binary physical format (version 3) additional intermediate location reference points may be used. These need to be placed directly after the first location reference point for the positive offset or directly before the last location reference point for the negative offset. The “LRP_length” to be used for the percentage method (see Equation 9) then refers to the length between the first LRP and the inserted intermediate (inserted intermediate and last LRP respectively).



Chapter G

Decoder

12. Decoding location references

The OpenLR™ decoder retrieves a map-independent location reference which was encoded by an OpenLR™ encoder described in section 11. The decoder resolves a (map-dependent) location using its own map. This map might differ from the one used during encoding.

Note:

This section provides recommendations how an OpenLR™ decoder might be implemented. There might be other implementations which are also compliant with the OpenLR™ standard.

12.1 Decoding steps

The following decoding steps describe the process of resolving a location from a location reference. The location reference comes in a physical representation. The decoder resolves a location in the decoder map or it reports on errors if no location can be resolved. The following steps act as a guideline for the decoding process.

Step – 1 *Decode physical data and check its validity*

The decoder receives a physical representation of a location reference and it needs to check whether the data is valid. If a check fails the decoder should report an error and stop the decoding process.

The decoder should also extract the physical data including offset information and resolve the interval information into lower and upper boundaries (if appropriate). The location type shall be resolved and the relative coordinates should be converted into absolute coordinates (if appropriate).

If the location type is Geo-Coordinate then return the resolved coordinates and finish the process.

Step – 2 *For each location reference point find candidate nodes*

Each location reference point contains coordinates specifying a node in the encoder map. The decoder should try to find so called candidate nodes in the decoder map whereby the coordinates of the candidate nodes are close by the coordinates of the location reference point coordinates. The airline distance should be used to identify close-by nodes.

It might happen that several candidate nodes for one location reference point exist. Nodes in the decoder map which are far away from the location reference point coordinates should not be considered as candidate nodes in the further processing.

If no candidate node can be determined for a location reference point the decoder should try to determine a candidate line directly. The LRP coordinate can be projected onto lines which are not far away from that coordinate. The following steps should take the projection points into account instead of one of the start or end node of the determined lines.

The direct search of lines using a projection point may also be executed even if candidate nodes are found. This might increase the number of candidate nodes but it could help to determine the correct candidate line in the next step if the nodes in the encoder and decoder map differ significantly.

Step – 3 *For each location reference point find candidate lines*

For each location reference point the decoder tries to determine lines which should fulfill the following constraints:

- The start node, end node for the last location reference point or projection point shall be close by the coordinates of the location reference point.

- The candidate lines should be outgoing lines (incoming lines for the last location reference point) of the candidate nodes or projection points determined in the previous step.
- The candidate lines should match the attributes functional road class, form of way and bearing as being extracted from the physical data. Slight variances are allowed and shall be taken into account in step 4.

If no candidate line can be found for a location reference point, the decoder should report an error and stop further processing.

Step – 4 *Rate candidate lines for each location reference point*

All candidate lines for a location reference point shall be rated according to the following criteria:

- The start node, end node for the last location reference point or projection point shall be as close by as possible to the coordinates of the location reference point
- The functional road class of the candidate line should match the functional road class of the location reference point
- The form of way of the candidate line should match the form of way of the location reference point
- The bearing of the candidate line should match indicated bearing angles of the location reference point
- Slight variances in the concrete values are allowed and shall be considered in the rating function.

The candidate lines should be ordered in a way that the best matching line comes first.

Step – 5 *Determine shortest-path(s) between two subsequent location reference points*

The decoder needs to compute a shortest-path between each pair of subsequent location reference points. For each pair of location reference point suitable candidate lines must be chosen and the candidate line of first location reference point of this pair acts as start for the shortest-path calculation. The candidate line of the second location reference point of this pair is the end of the shortest-path calculation. If the lines being chosen are equal, then no shortest-path calculation needs to be started.

The shortest path algorithm should take the part of the network into account which contains all lines having a functional road class lower than or equal to the lowest functional road class of the first location reference point of the pair. This value might be altered if the decoder anticipates having different functional road class values than the encoder map.

Additionally the shortest-path algorithm should fulfill the following constraints:

- All lengths of the lines should be measured in meters and should also be converted to integer values, so that float values need to be rounded correctly.
- The search is node based and will start at the start node of the first line and will end at the end node of the last line.
- The algorithm shall return an ordered list of lines representing the calculated shortest-path.

If no shortest-path can be calculated for two subsequent location reference points, the decoder might try a different pair of candidate lines or finally fail and report an error. If trying a different pair of candidate lines it might happen that the start line needs to be changed. In such a case this also affects the end line of the previous shortest-path and this path also needs to be re-calculated and checked again. The number of retries of shortest-path calculations should be limited in order to guarantee a fast decoding process.

Step – 6 *Check validity of the calculated shortest-path(s)*

After the shortest-path calculation the length of such a path should be checked against the distance to next point information of the first location reference point of a pair. If the length information differ

too much the decoder could decide to try a different pair of candidate lines (see also Step – 5) or to fail and report an error.

Step – 7 *Concatenate shortest-path(s) to form the location and trim path according to the offsets*

If all shortest-paths are calculated and checked the decoder concatenates the shortest-path(s) according to the order of the location reference points. This concatenation shall be trimmed using the positive and negative offset of the physical data. Some hints on how to trim the location using the different offset implementations are given in section 12.3. The resulting path forms the decoder location and the decoder returns the location.

12.2 Overview of the decoding steps

The following Table 48 summarizes the decoding steps and the following figures outline the workflow of OpenLR™ decoding for Geo-Coordinate locations (Figure 24) and for all other location types (Figure 25).

| Step | Action |
|-----------------|---|
| Step – 1 | <i>Decode physical data and check its validity</i> |
| Step – 2 | <i>For each location reference point find candidate nodes</i> |
| Step – 3 | <i>For each location reference point find candidate lines</i> |
| Step – 4 | <i>Rate candidate lines for each location reference point</i> |
| Step – 5 | <i>Determine shortest-path(s) between two subsequent location reference points</i> |
| Step – 6 | <i>Check validity of the calculated shortest-path(s)</i> |
| Step – 7 | <i>Concatenate shortest-path(s) to form the location and trim path according to the offsets</i> |

Table 48: Decoding steps overview

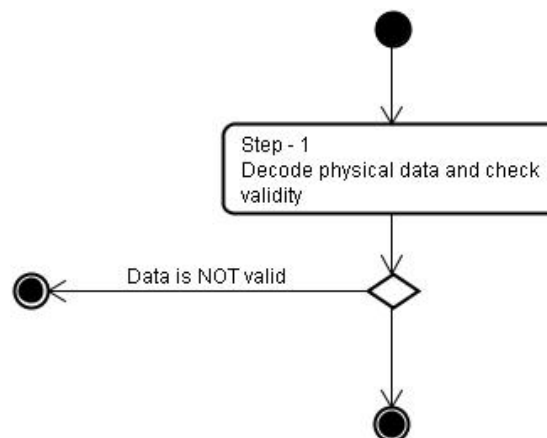


Figure 24: Workflow – OpenLR™ decoding (Geo-Coordinate locations)

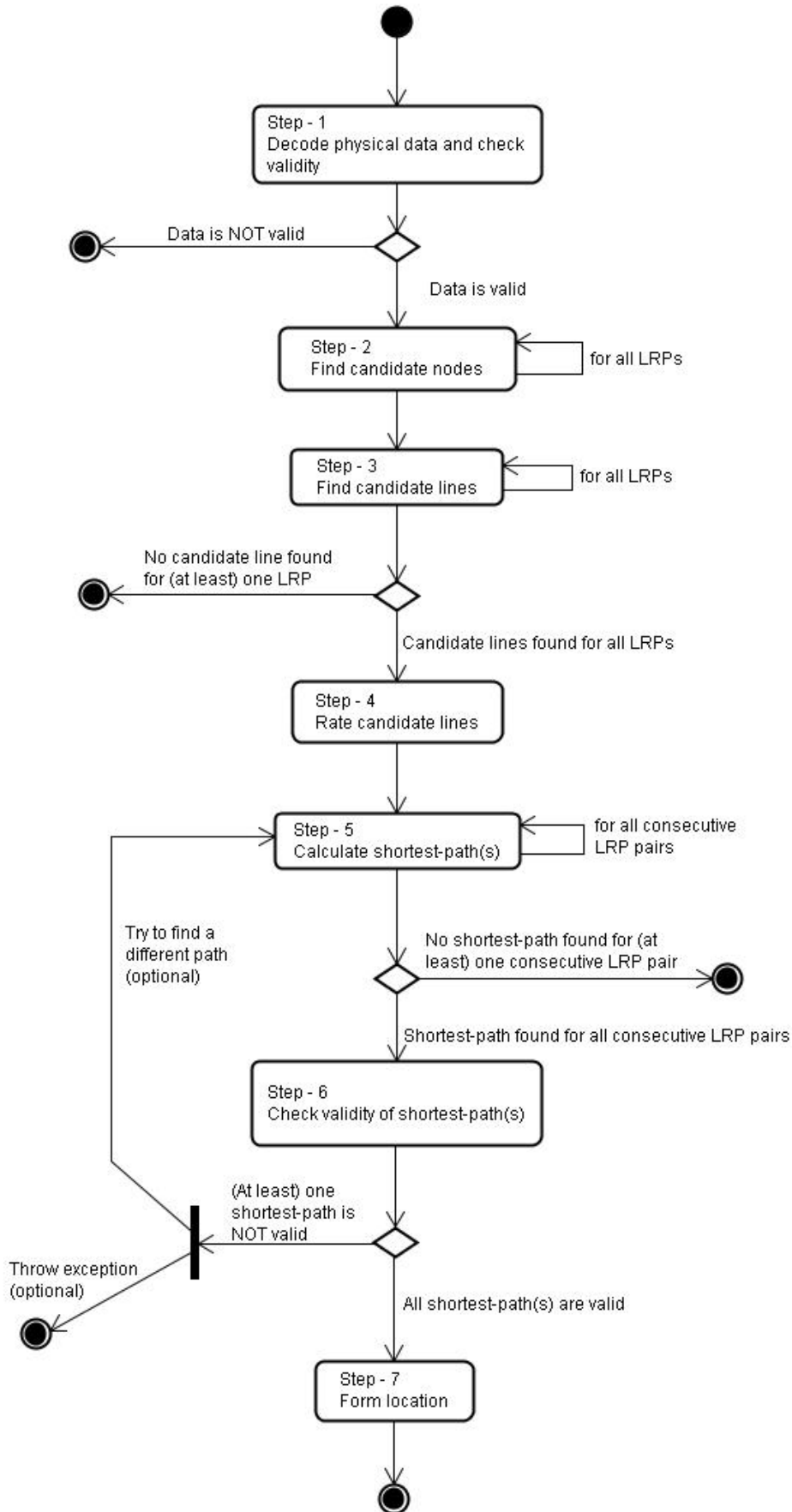


Figure 25: Workflow – OpenLR™ decoding (except Geo-Coordinate locations)



12.3 Decoding offsets

Decoding the offsets and trimming the location depend on the physical format. The XML format allows to transmit the real values, the binary format version 3 (percentage method) defines a relative value and needs the distance between the first two location reference points for the calculation of the positive offset (last two location reference points for negative offset). The binary format version 2 defines intervals.

The most accurate format is XML, followed by the binary format version 3. The most inaccurate format is the binary format version 2. The decoder implementation must take these differences into account.



Chapter H

Examples



13. OpenLR™ example

OpenLR™ shall now be explained using an abstract example. The example defines two different maps and it executes the encoder steps as described in section 11.1, outlines the binary and XML representation and it executes the decoder steps as described in section 12.1.

Section 13.1 explains an example of encoding and decoding a line location and section 13.2 explains an example of encoding and decoding the different types of point locations.

13.1 Line Location

13.1.1. Example setup

The encoder map is shown in Figure 26 and consists of 15 nodes and 23 lines (two-way lines are counted twice). The nodes are numbered from 1 to 15. The necessary line attributes are shown beside every line using the format: <FRC>, <FOW>, <Length in meter>. The arrows indicate the driving direction for each line. The coordinates chosen exemplarily and do not correspond to any real world objects!

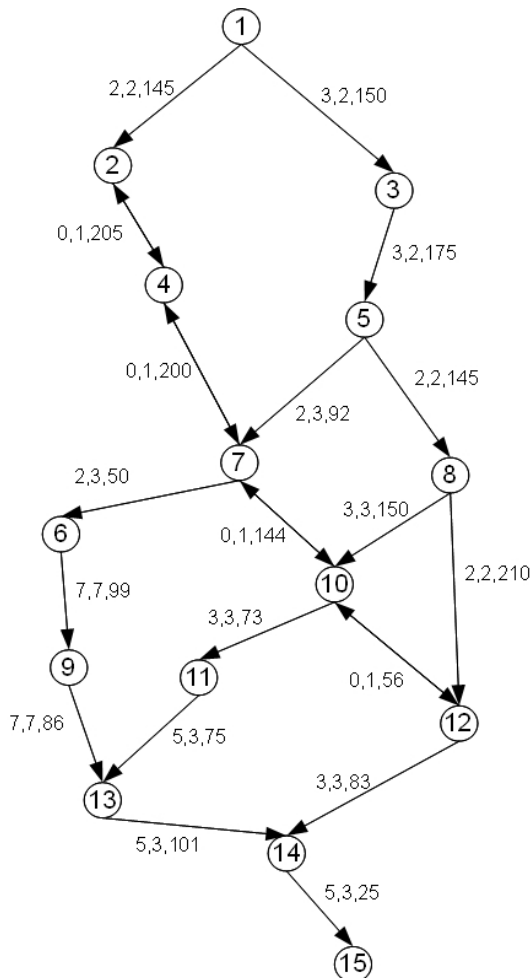


Figure 26: Example network

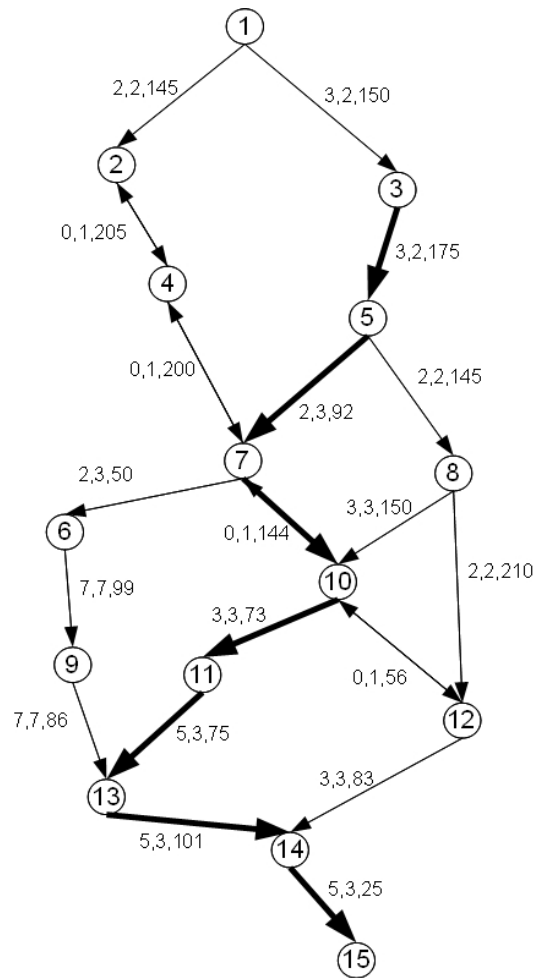


Figure 27: Location

The location to be encoded is shown in Figure 27 using bold lines. The location starts at node 3 and continues over the nodes 5, 7, 10, 11, 13, 14, and ends at node 15. Its total length in the encoder

map is 685 meters. The ordered list of lines and the map to be used during encoding serves as input for the OpenLR™ encoder.

13.1.2. Encoding

In step 1 of the encoding process the location will first be checked for validity. Since the location is connected and drivable and all functional road classes along the location are between 0 and 7, this location is valid. Given that turn restrictions are omitted from the map data used for this example, the encoder can ignore this check.

The encoder step 2 checks the start and end node of the location being valid nodes according to the data format rules in section 6. The end node 15 has only one incoming line and is therefore valid.

The start node 3 also has two incident lines but here it is one outgoing and one incoming line. Therefore this node is not valid and the encoder searches for a valid node outside the location. The encoder will find node 1 to be a valid node and it also expands the location uniquely. Node 1 is chosen as the new start node for the location reference and there will be a positive offset of 150 meters. The total length of the location reference path results in 835 meters.

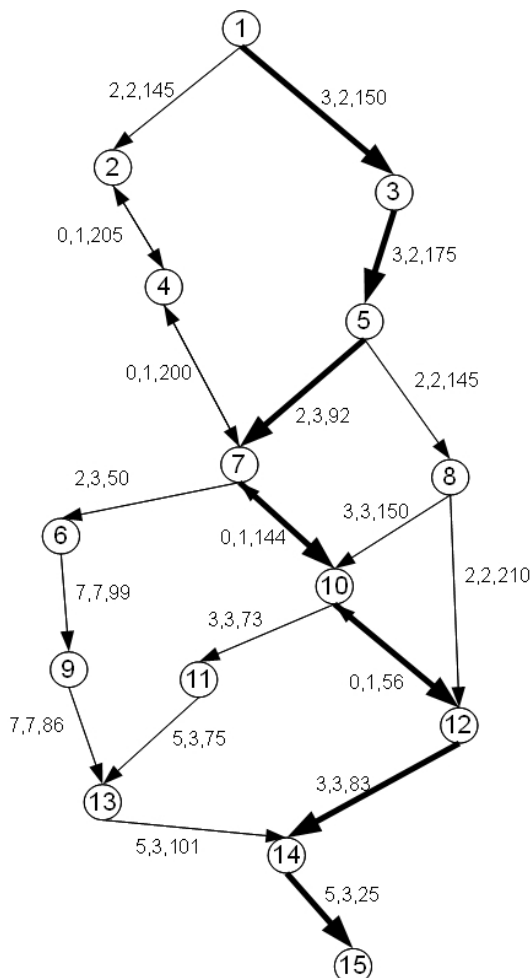


Figure 28: Shortest path

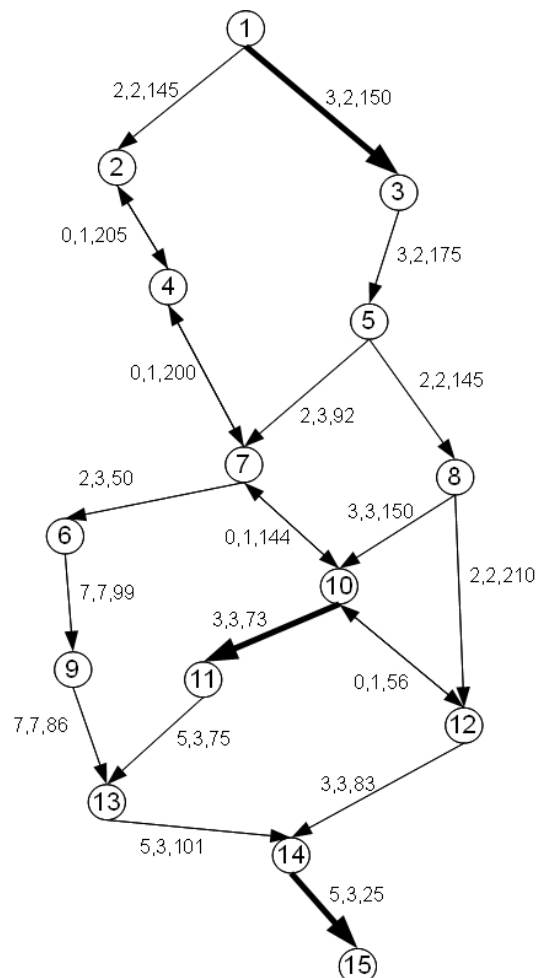


Figure 29: Location reference points

The encoder proceeds with step 3 and calculates a shortest-path between the start line (line between nodes 3 and 5) and the end line (line between nodes 14 and 15) of the location. The line between the nodes 1 and 3 may be ignored because this line was added during the extension phase. The resulting shortest-path (including the extended line) is outlined in Figure 28 using bold lines. The shortest-path has a length of 725 meters (without the extended line: 675 meters).

Step 4 of the encoding process now check whether the location is covered by the calculated shortest-path. It will determine that this is not the case and there is a deviation after node 10. Proceed with step 5 the encoder will determine the line from node 10 to 11 as becoming a new intermediate location reference point. Node 10 is a valid node since it cannot be stepped over during route search and the shortest-path to this line covers the corresponding part of the location completely. The length of the location being covered after this first shortest-path calculation is 561 meters.

The following step 6 prepares the route calculation in order to determine a shortest-path for the remaining part of the location (from node 10 over 11 and 14 to 15). The shortest-path calculation will therefore start at the line from 10 to 11 and ends at the line from 14 to 15.

Step 3 will determine a shortest path (length: 274 meters) between 10 and 15 and step 4 will return that the location is now completely covered by the calculated shortest paths.

In step 7 the location reference path will be composed of the two shortest-paths and the ordered list of location reference points will be formed. Figure 29 shows the lines being selected for the location reference points. The first location reference point points to the line from node 1 to 3 and indicates the start of the location reference path, the second location reference point points to the line from node 10 to 11 and this line was necessary to avoid the deviation from the location. The last location reference point points to the line from node 14 to 15 and indicates the end of the location reference path.

The following step 8 checks the validity of the location reference. Since all lengths between two subsequent location reference points are less than the maximum distance, the location reference emphasizes as being valid.

The last step of the encoding process (step 10) will generate the binary and XML representation of the location reference.

13.1.3. Binary location reference

The location reference consists of three location reference points and Table 49 shows the coordinates for the nodes 1, 10 and 15. These nodes are the corresponding nodes to the location reference points. In preparation of the binary format this table also shows the relative coordinates. The node 1 corresponds to the location reference point 1 and will have coordinates in absolute format. Node 10 corresponding to location reference point 2 will have relative coordinates to the location reference point 1. Node 15 corresponding to location reference point 2 will also have relative coordinates but now referencing to location reference point 2.

| Node ID | LRP index | Longitude | Latitude | Relative longitude | Relative latitude |
|---------|-----------|-----------|-----------|--------------------|-------------------|
| 1 | 1 | 6.12683° | 49.60851° | -- | -- |
| 10 | 2 | 6.12838° | 49.60398° | 155 | -453 |
| 15 | 3 | 6.12817° | 49.60305° | -21 | -93 |

Table 49: Example coordinates

The offsets being calculated in step 2 of the encoding process are shown in Table 50. In the binary data only the positive offset will appear because the negative offset is 0 and a missing offset will be treated as 0. The length between the first two location reference points is 561m

| Field | Value | % |
|-----------------|-------|-------|
| positive Offset | 150 | 26,74 |
| negative Offset | 0 | -- |

Table 50: Example offset values

Table 51 collects the relevant data for each location reference point. This includes the functional road class, the form of way and the bearing of the corresponding line. The needed information about the path between two subsequent location reference points is also shown (lowest functional road class and distance to the next location reference point).



| LRP index | FRC | FOW | BEAR | LFRCNP | DNP |
|-----------|------|----------------------|------|--------|-----|
| 1 | FRC3 | MULTIPLE_CARRIAGEWAY | 135° | FRC3 | 561 |
| 2 | FRC3 | SINGLE_CARRIAGEWAY | 227° | FRC5 | 274 |
| 3 | FRC5 | SINGLE_CARRIAGEWAY | 290° | -- | -- |

Table 51: Location reference points determined during encoding

These tables above hold all relevant information for creating the binary data. The following tables outline the binary data according to section 8:

- Status byte: see Table 52
- LRP 1: see Table 53 to Table 57
- LRP 2: see Table 58 to Table 62
- LRP 3: see Table 63 to Table 66
- Offset: see Table 67

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-----|----|-----|----|---------|---|---|
| Description | RFU | RFU | PF | ArF | AF | Version | | |
| Value | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

Table 52: Binary example: status byte

| Byte | First | | | | | | | | Second | | | | | | | | Third | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |

Table 53: Binary example: LRP 1 – absolute longitude

| Byte | First | | | | | | | | Second | | | | | | | | Third | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

Table 54: Binary example: LRP1 – absolute latitude

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-----|-----|---|---|-----|---|---|
| Description | RFU | RFU | FRC | | | FOW | | |
| Value | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |

Table 55: Binary example: LRP1 – attribute 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|--------|---|---|---|---------|---|---|---|
| Description | LFRCNP | | | | Bearing | | | |
| Value | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

Table 56: Binary example: LRP1 – attribute 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|---|---|---|---|---|---|---|
| Description | DNP | | | | | | | |
| Value | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

Table 57: Binary example: LRP1 – attribute 3

| Byte | First | | | | | | | | Second | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |

Table 58: Binary example: LRP2 – relative longitude

| Byte | First | | | | | | | | Second | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

Table 59: Binary example: LRP2 – relative latitude



| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-----|-----|---|---|-----|---|---|
| Description | RFU | RFU | FRC | | | FOW | | |
| Value | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |

Table 60: Binary example: LRP2 – attribute 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|--------|---|---|---------|---|---|---|---|
| Description | LFRCNP | | | Bearing | | | | |
| Value | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |

Table 61: Binary example: LRP2 – attribute 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|---|---|---|---|---|---|---|
| Description | DNP | | | | | | | |
| Value | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Table 62: Binary example: LRP2 – attribute 3

| Byte | First | | | | | | | | Second | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |

Table 63: Binary example: LRP3 – relative longitude

| Byte | First | | | | | | | | Second | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

Table 64: Binary example: LRP3 – relative latitude

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-----|-----|---|---|-----|---|---|
| Description | RFU | RFU | FRC | | | FOW | | |
| Value | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

Table 65: Binary example: LRP3 – attribute 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-------|-------|---------|---|---|---|---|
| Description | RFU | PoffF | NoffF | Bearing | | | | |
| Value | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |

Table 66: Binary example: LRP3 – attribute 4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------|---|---|---|---|---|---|---|
| Description | POFF | | | | | | | |
| Value | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

Table 67: Binary example: positive Offset

The full binary data stream will have a length of 24 bytes and consists of the following (ordered as bytes from left to right and top to down, see integer values in parentheses):

```

00001010(10)  00000100(4)   01011011(91)  01011011(91)   00100011(35)  01000110(70)
11110100(-12) 00011010(26)  01101100(108) 00001001(9)   00000000(0)   10011011(-101)
11111110(-2)  00111011(59)  00011011(27)  10110100(-76) 00000100(4)   11111111(-1)
11101011(-21) 11111111(-1)  10100011(-93) 00101011(43)  01011001(89)  01000100(68)
    
```

13.1.4. XML location reference

The location reference consists of three location reference points in total. The XML format therefore contains two location reference points and a last location reference points. The offset field contains only the positive offset value as there is no negative offset.

The XML representation also contains the Base64-encoded binary representation of the location reference.



```

<?xml version="1.0" encoding="UTF-8"?>
<openlr:OpenLR xmlns:openlr="http://www.openlr.org/openlr"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openlr.org/openlr openlr.xsd ">
  <openlr:LocationID>ExampleLocation</openlr:LocationID>
  <openlr:XMLLocationReference>
    <openlr:LineLocationReference>
      <openlr:LocationReferencePoint>
        <openlr:Coordinates>
          <openlr:Longitude>6.12683</openlr:Longitude>
          <openlr:Latitude>49.60851</openlr:Latitude>
        </openlr:Coordinates>
        <openlr:LineAttributes>
          <openlr:FRC>FRC3</openlr:FRC>
          <openlr:FOW>MULTIPLE_CARRIAGEWAY</openlr:FOW>
          <openlr:BEAR>135</openlr:BEAR>
        </openlr:LineAttributes>
        <openlr:PathAttributes>
          <openlr:LFR CNP>FRC3</openlr:LFR CNP>
          <openlr:DNP>561</openlr:DNP>
        </openlr:PathAttributes>
      </openlr:LocationReferencePoint>
      <openlr:LocationReferencePoint>
        <openlr:Coordinates>
          <openlr:Longitude>6.12838</openlr:Longitude>
          <openlr:Latitude>49.60398</openlr:Latitude>
        </openlr:Coordinates>
        <openlr:LineAttributes>
          <openlr:FRC>FRC3</openlr:FRC>
          <openlr:FOW>SINGLE_CARRIAGEWAY</openlr:FOW>
          <openlr:BEAR>227</openlr:BEAR>
        </openlr:LineAttributes>
        <openlr:PathAttributes>
          <openlr:LFR CNP>FRC5</openlr:LFR CNP>
          <openlr:DNP>274</openlr:DNP>
        </openlr:PathAttributes>
      </openlr:LocationReferencePoint>
      <openlr>LastLocationReferencePoint>
        <openlr:Coordinates>
          <openlr:Longitude>6.12817</openlr:Longitude>
          <openlr:Latitude>49.60305</openlr:Latitude>
        </openlr:Coordinates>
        <openlr:LineAttributes>
          <openlr:FRC>FRC5</openlr:FRC>
          <openlr:FOW>SINGLE_CARRIAGEWAY</openlr:FOW>
          <openlr:BEAR>290</openlr:BEAR>
        </openlr:LineAttributes>
      </openlr>LastLocationReferencePoint>
      <openlr:Offsets>
        <openlr:PosOff>150</openlr:PosOff>
      </openlr:Offsets>
    </openlr:LineLocationReference>
  </openlr:XMLLocationReference>
  <openlr:BinaryLocationReferences>
    <openlr:BinaryLocationReference id="BinaryExample" version="2">
      CgRbWyNG9BpsCQCb/jsbtAT/6/+jK1kC
    </openlr:BinaryLocationReference>
  </openlr:BinaryLocationReferences>
  <openlr:AdditionalInformation>
    <openlr:BoundingBox>
      <openlr:LowerLeft>
        <openlr:Longitude>6.12601</openlr:Longitude>
        <openlr:Latitude>49.60243</openlr:Latitude>
      </openlr:LowerLeft>
      <openlr:UpperRight>
        <openlr:Longitude>6.12951</openlr:Longitude>
        <openlr:Latitude>49.60891</openlr:Latitude>
      </openlr:UpperRight>
    </openlr:BoundingBox>
  </openlr:AdditionalInformation>

```



```

</openlr:UpperRight>
</openlr:BoundingBox>
<openlr:MapDatabase>ExampleMap Version 0.1</openlr:MapDatabase>
<openlr:AdditionalData/>
</openlr:AdditionalInformation>
</openlr:OpenLR>

```

13.1.5. Decoding

This section outlines the decoding of the location reference being prepared in the sections above. The decoding walks along the steps presented in section 12.1.

The step 1 of the decoding process extracts the physical data (binary format) and checks the validity of the location reference. The information extracted from the binary data will not be as accurate as before creating the binary stream. Due to the usage of intervals for the bearing and the distance to next point the concrete value cannot be extracted but a small interval containing the concrete value. The offset value is chosen as the mid point of the interval (calculation: $0.390625 * 68 * 1.5 = 26.76$).

The information being extracted from the binary data example is shown in Table 68, Table 69 and Table 70.

| LRP index | Longitude | Latitude |
|-----------|-----------|-----------|
| 1 | 6.12682° | 49.60850° |
| 2 | 6.12838° | 49.60397° |
| 3 | 6.12817° | 49.60304° |

Table 68: Decoded coordinates

| LRP index | FRC | FOW | Bearing | LFRCNP | DNP |
|-----------|------|----------------------|-------------------|--------|-----------------|
| 1 | FRC3 | MULTIPLE_CARRIAGEWAY | 135.00° - 146.25° | FRC3 | 527.4m – 586.0m |
| 2 | FRC3 | SINGLE_CARRIAGEWAY | 225.00° - 236.25° | FRC5 | 234.4m – 293.0m |
| 3 | FRC5 | SINGLE_CARRIAGEWAY | 281.25° - 292.50° | -- | 0m |

Table 69: Decoded LRP information

| Offset | Value |
|-----------------|-------------------------|
| Positive offset | 26,76% |
| Negative offset | - no offset available - |

Table 70: Decoded offset information

This information is sufficient to resolve the location on the decoder map shown in Figure 30. The map consists of 17 nodes and 26 lines (two-way lines are counted twice).

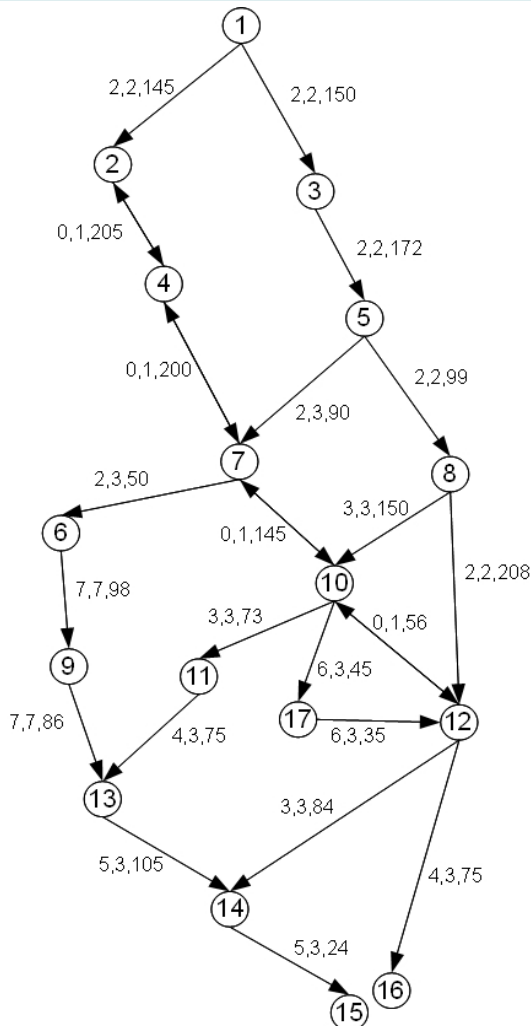


Figure 30: Decoder map

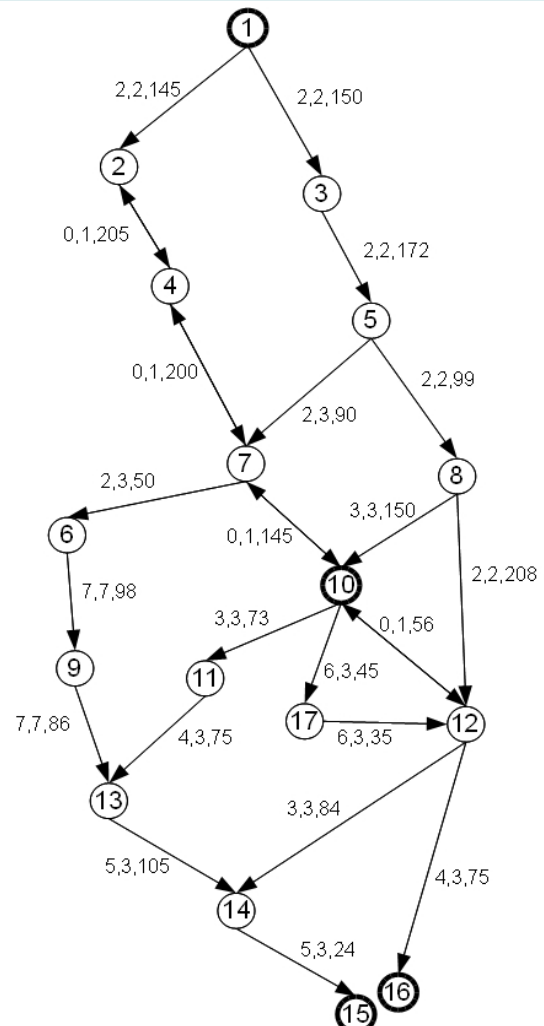


Figure 31: Candidate nodes

This map differs from the encoder map (see Figure 26) in several ways. Some length values are different (e.g. line from node 3 to 5), some functional road class values have changed (e.g. line from node 3 to 5) and there are two more nodes 16 and 17 and also additional lines connecting these new nodes. The challenge of the decoder is to resolve the location in this different map.

After validating the data the decoder starts determining candidate nodes for each location reference point (step 2). Figure 31 shows the candidate nodes (bold circle) which are positioned close by the coordinates of the location reference points. For the location reference point 1 and 2 exist only one candidate node but for the last location reference point two candidate nodes are in the run.

Step 3 deals with determining candidate lines for each location reference point. The bold lines in Figure 32 are the candidate lines for this example. The location reference point 1 has two outgoing lines as candidates, location reference point 2 has three outgoing lines in the run and the last location reference point has two incoming lines (one for each candidate node).

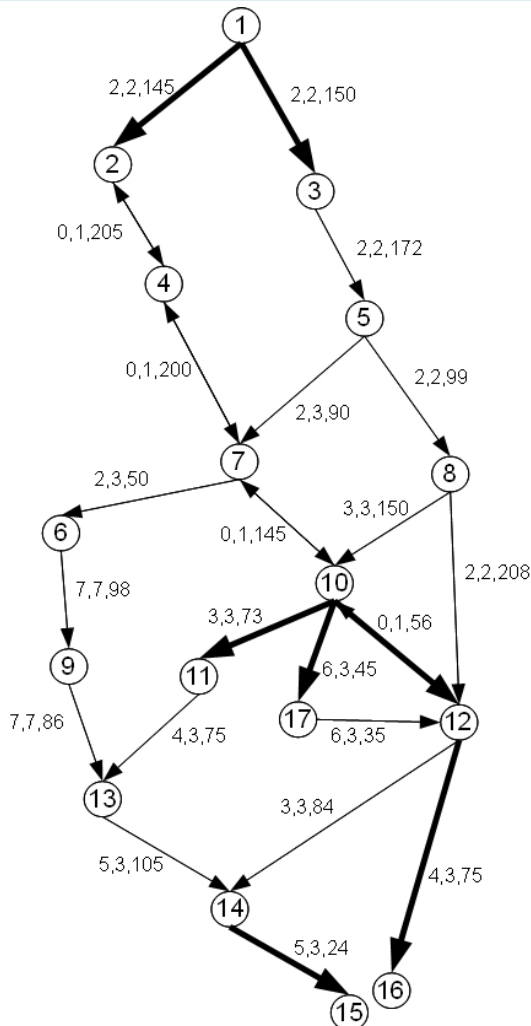


Figure 32: Candidate lines

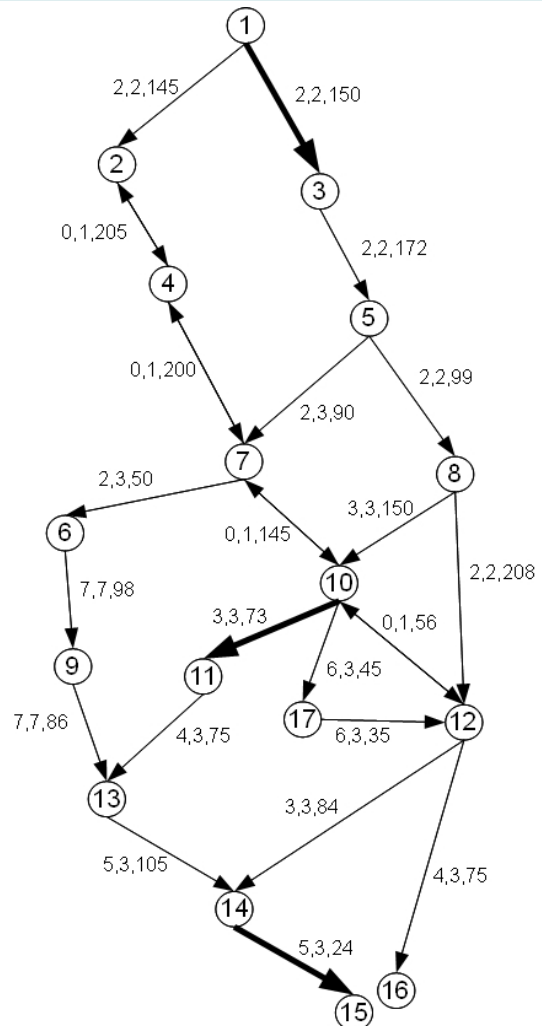


Figure 33: Location reference point - lines

Each candidate line will be rated in step 4 according to its compliance with the attributes of the location reference point and its distance to the location reference coordinate. The bold lines in Figure 33 show the lines which are rated best. These lines will be used for the following shortest-path calculation in step 5 of the decoding process.

The shortest-path calculation will run for each subsequent pair of location reference points and the concatenation of the two shortest-paths in this example are shown in Figure 34. These shortest-paths will be validated in step 6. The length of the first shortest-path (from node 1 to node 10) is 557 meters and this value fits into the distance to next point interval of the first location reference point (527.4 meters – 586.0 meters). The length of the second shortest-path (from node 10 to node 15) is 277 meters and this value also fits into the distance to next point interval of the location reference point 2 (234.4 meters – 293.0 meters). The shortest-paths are validated and the decoder does not fail.

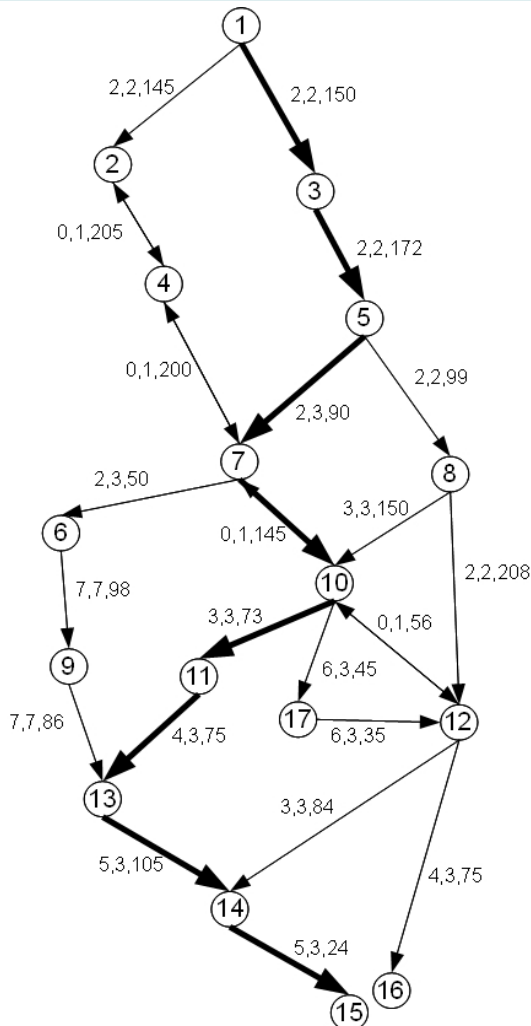


Figure 34: Decoder - shortest path

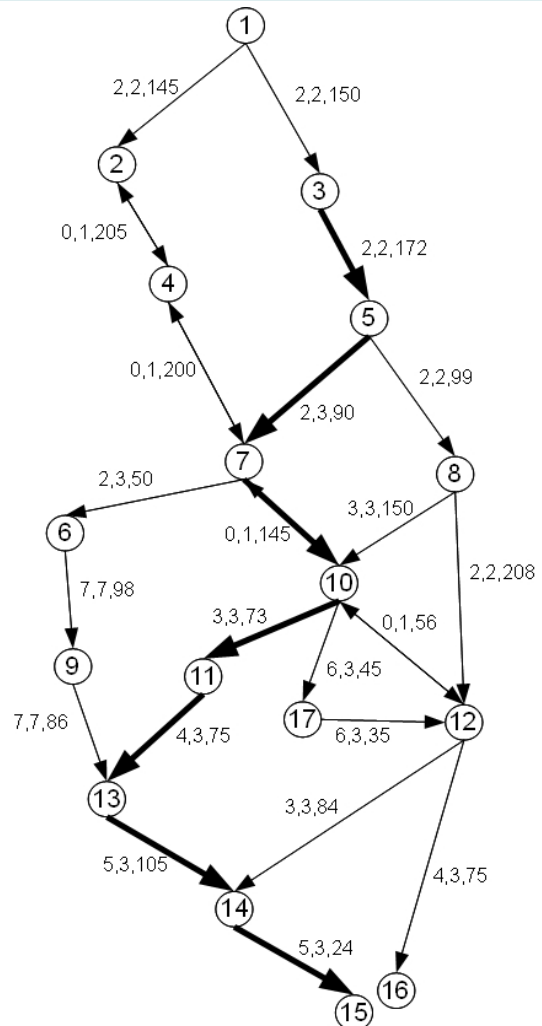


Figure 35: Decoder location

The last step of the decoding process (step 7) now trims the concatenated shortest-path according to the offsets. The example only holds a positive offset and therefore the decoder trims the shortest path at its start. The length between the first two location reference points is 557 meters. The offset value is therefore $557 * 0.2676 = 149$ meters. It points out that node 3 is the real start of the location and the decoded location is finally shown in Figure 35.

13.2 Point location

13.2.1. Example setup

The encoder and decoder maps (see Figure 36 and Figure 37) are extracts of the maps being used for the line location example. Both maps consist of 6 nodes and 8 lines. The necessary line attributes are shown beside every line using the format: <FRC>, <FOW>, <Length in meter>. The arrows indicate the driving direction for each line. The coordinates chosen are exemplary and do not correspond to any real world objects!

All nodes in the network shall be valid even there are only two lines attached to the nodes!!

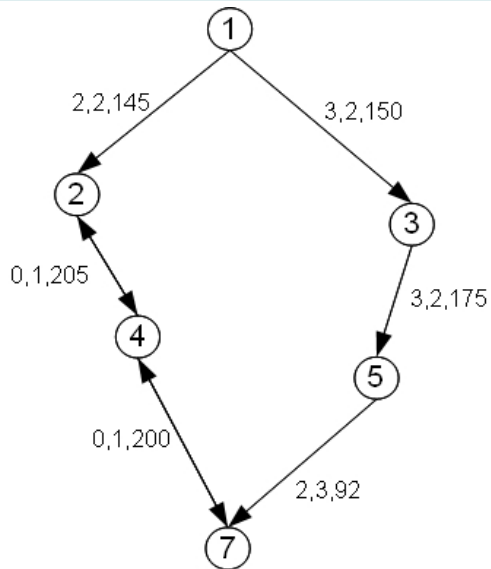


Figure 36: Encoder map

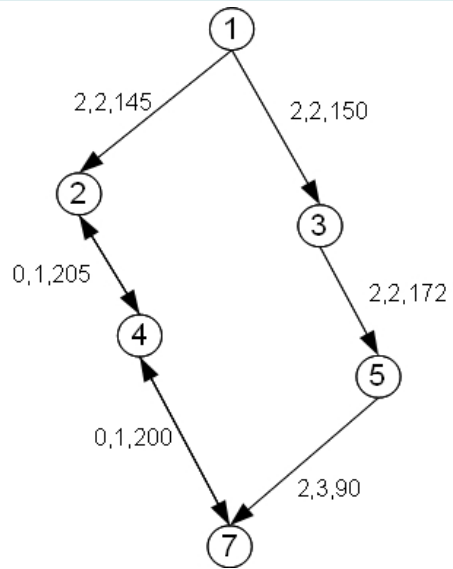


Figure 37: Decoder map

The following examples will need the coordinates of the nodes 5 and 7 shown in Table 71.

| Node ID | Longitude | Latitude |
|---------|-----------|-----------|
| 5 | 6.12829° | 49.60597° |
| 7 | 6.12779° | 49.60521° |

Table 71: Example data

13.2.2. GeoCoordinate

The geo coordinate example shows the simplest type of a node location. Figure 38 shows the geo coordinates to be encoded with a black cross and the coordinates used in this example are shown in Table 72.

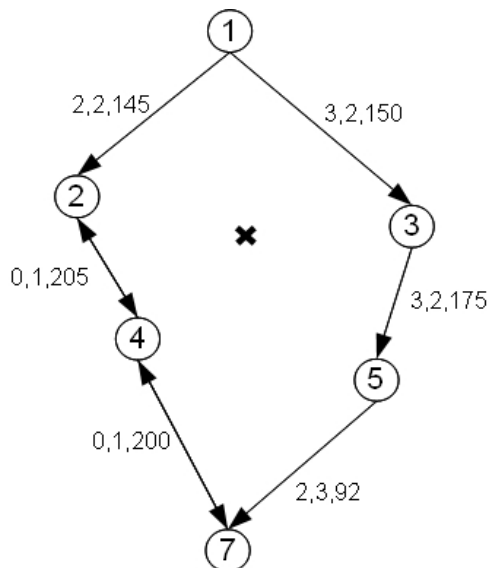


Figure 38: Example - GeoCoordinate (encoder)

| | Longitude | Latitude |
|---|-----------|-----------|
| X | 6.12699° | 49.60728° |

Table 72: Example coordinates

13.2.2.1 Encoding

The encoding of geo coordinates starts with a validation of the coordinates itself. These must fit into the valid ranges for longitude and latitude coordinates. If the geo coordinates are invalid then the encoder shall report an error. If the longitude and latitude values are correct then the encoder can immediately create the physical format.

Binary location reference

The size of the binary location reference for a GeoCoordinate is 7 bytes. The following tables (Table 73 to Table 75) explain the layout for each single byte.

The status byte indicates that the location reference describes a point location using version 3 of the binary format. The geo coordinates (longitude and latitude values) are transformed into integer values according to Equation 1 and then translated into the binary format.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-----|----|-----|----|---------|---|---|
| Description | RFU | RFU | PF | ArF | AF | Version | | |
| Value | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

Table 73: Binary example - GeoCoordinate: status byte

| Byte | First | | | | | | | Second | | | | | | | Third | | | | | | | | | |
|-------|-------|---|---|---|---|---|---|--------|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

Table 74: Binary example - GeoCoordinate: Absolute longitude

| Byte | First | | | | | | | Second | | | | | | | Third | | | | | | | | | |
|-------|-------|---|---|---|---|---|---|--------|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

Table 75: Binary example - GeoCoordinate: Absolute latitude

XML location reference

The XML location reference shows a point location reference which consists of a GeoCoordinate. This special type of point location contains only a pair of longitude and latitude values referring to the geo coordinates.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<openlr:OpenLR xmlns="http://www.openlr.org/openlr">
  <openlr:LocationID>TestID01</openlr:LocationID>
  <openlr:XMLLocationReference>
    <openlr:PointLocationReference>
      <openlr:GeoCoordinate>
        <openlr:Coordinates>
          <openlr:Longitude>6.12699</openlr:Longitude>
          <openlr:Latitude>49.60728</openlr:Latitude>
        </openlr:Coordinates>
      </openlr:GeoCoordinate>
    </openlr:PointLocationReference>
  </openlr:XMLLocationReference>
</openlr:OpenLR>
```

13.2.2.2 Decoding

The decoding starts with the validation of the physical format. If the format is invalid and cannot be read then the decoder should fail.

If the physical format is valid the decoder needs to extract the geo coordinates. For the binary physical format this will result in the longitude and latitude values shown in Table 76. The decoded GeoCoordinate is shown in Figure 39.

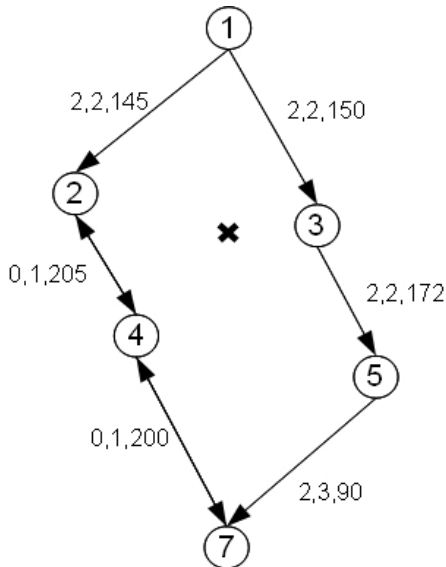


Figure 39: Example: Decoded GeoCoordinate

| | Longitude | Latitude |
|---|-----------|-----------|
| X | 6.12699° | 49.60727° |

Table 76: Decoded GeoCoordinate values

13.2.3. PointAlongLine

The PointAlongLine defines a point location directly on a line or near by a line. The point of interest is identified by a network line an offset value. In this example the point location marked with a black cross in Figure 40 shall be encoded. The distance between the start of the line and the point of interest shall be 28 meter or 30.43% of the line length. The additional information about the side of the road and the orientation are mentioned in Table 77.

| Information type | Value |
|------------------|---------------------------|
| SideOfRoad | LEFT |
| Orientation | NO_ORIENTATION_OR_UNKNOWN |

Table 77: Additional information for point along line

Please note that all nodes in the example shall be valid even if there are only two lines attached to a node!

13.2.3.1 Encoding

The encoding of a PointAlongLine location starts with validating the input data. The point location shown in Figure 40 needs to be translated into a network line and an offset value. The point location is on the determined line and the offset describes the distance between the start node of that line and the point location (see Figure 41).

The line including the offset value will be the input for the OpenLR™ encoder and needs to be checked. The offset value shall not exceed the length of the line. If the validation of the input data fails then the encoder shall report an error.

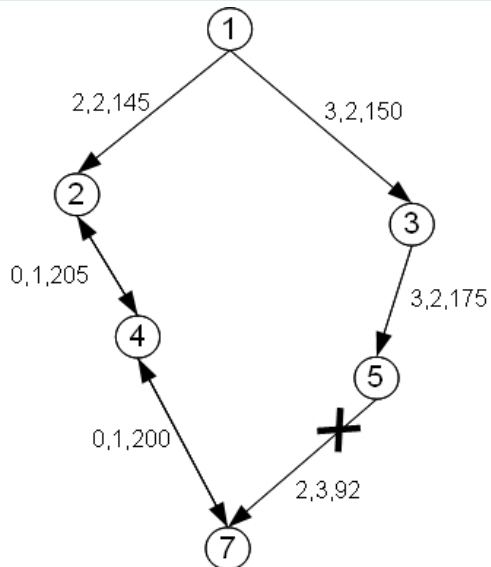


Figure 40: Example - PointAlongLine

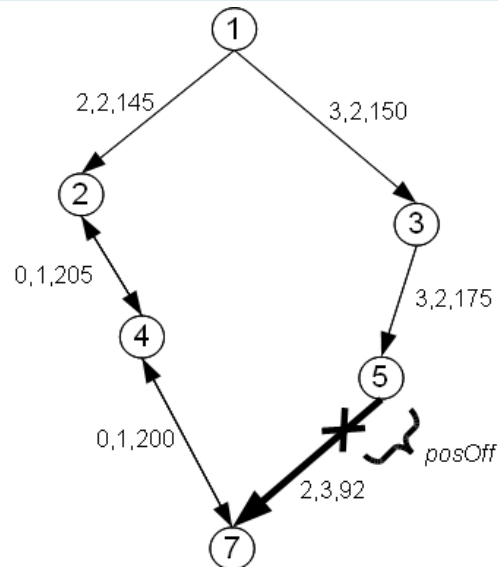


Figure 41: PointAlongLine - Reference line and offset

If the input data is valid then the encoder would try to expand the referenced line in order to find valid nodes as start and end of the location. As all nodes are defined as valid the encoder can directly create two location reference points describing the reference line. The first LRP points to the start of the line (node 5 in Figure 41) and the second LRP (last LRP) points to the end of the reference line (node 7 in Figure 41). The line will therefore be referenced twice but one time in the direction from node 5 to node 7 and the second time against that direction.

Using two LRPs also provides the opportunity to use an artificial location reference point (not to be bound to a network node) in order to reduce the distance between the two location reference points and therefore to increase the accuracy of the offset.

In the last step the two location reference points and the offset value needs to be physically encoded.

Binary location reference

The following tables (Table 78 to Table 88) show the details of the binary physical format (version 3). The data size in this example is 17 bytes. The positive offset value of 30.43% falls into the bucket number 77.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-----|----|-----|----|---------|---|---|
| Description | RFU | RFU | PF | ArF | AF | Version | | |
| Value | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

Table 78: Binary example: status byte

| Byte | First | | | | | | | | Second | | | | | | | | Third | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

Table 79: Binary example: LRP 1 – absolute longitude

| Byte | First | | | | | | | | Second | | | | | | | | Third | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | |

Table 80: Binary example: LRP1 – absolute latitude

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------|-------------|---|---|-----|---|---|-----|---|--|
| Description | Orientation | | | FRC | | | FOW | | |
| Value | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | |

Table 81: Binary example: LRP1 – attribute 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|--------|---|---|---------|---|---|---|---|
| Description | LFRCNP | | | Bearing | | | | |
| Value | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

Table 82: Binary example: LRP1 – attribute 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|---|---|---|---|---|---|---|
| Description | DNP | | | | | | | |
| Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 83: Binary example: LRP1 – attribute 3

| Byte | First | | | | | | | Second | | | | | | | | |
|-------|-------|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

Table 84: Binary example: LRP2 – relative longitude

| Byte | First | | | | | | | Second | | | | | | | | |
|-------|-------|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

Table 85: Binary example: LRP2 – relative latitude

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------------|---|-----|---|---|-----|---|---|
| Description | SideOfRoad | | FRC | | | FOW | | |
| Value | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

Table 86: Binary example: LRP2 – attribute 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-------|-------|---------|---|---|---|---|
| Description | RFU | PoffF | NoffF | Bearing | | | | |
| Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

Table 87: Binary example: LRP3 – attribute 4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------|---|---|---|---|---|---|---|
| Description | POFF | | | | | | | |
| Value | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |

Table 88: Binary example: positive Offset**XML location reference**

The following XML location reference shows a PointAlongLine location consisting of exactly two location reference points and one (positive) offset.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<openlr:OpenLR xmlns="http://www.openlr.org/openlr">
  <openlr:LocationID>TestID02</openlr:LocationID>
  <openlr:XMLLocationReference>
    <openlr:PointLocationReference>
      <openlr:PointAlongLine>
        <openlr:LocationReferencePoint>
          <openlr:Coordinates>
            <openlr:Longitude>6.12829</openlr:Longitude>
            <openlr:Latitude>49.60597</openlr:Latitude>
          </openlr:Coordinates>
          <openlr:LineAttributes>
            <openlr:FRC>FRC2</openlr:FRC>
            <openlr:FOW>MULTIPLE_CARRIAGEWAY</openlr:FOW>
            <openlr:BEAR>202</openlr:BEAR>
          </openlr:LineAttributes>
          <openlr:PathAttributes>
            <openlr:LFRCNP>FRC2</openlr:LFRCNP>
            <openlr:DNP>92</openlr:DNP>
          </openlr:PathAttributes>
        </openlr:LocationReferencePoint>
      </openlr:PointAlongLine>
    </openlr:PointLocationReference>
  </openlr:XMLLocationReference>
</openlr:OpenLR>
```



```

        </openlr:PathAttributes>
    </openlr:LocationReferencePoint>
    <openlr>LastLocationReferencePoint>
        <openlr:Coordinates>
            <openlr:Longitude>6.12779</openlr:Longitude>
            <openlr:Latitude>49.60521</openlr:Latitude>
        </openlr:Coordinates>
        <openlr:LineAttributes>
            <openlr:FRC>FRC2</openlr:FRC>
            <openlr:FOW>MULTIPLE_CARRIAGEWAY</openlr:FOW>
            <openlr:BEAR>42</openlr:BEAR>
        </openlr:LineAttributes>
    </openlr>LastLocationReferencePoint>
    <openlr:Offsets>
        <openlr:PosOff>28</openlr:PosOff>
        <openlr:NegOff>0</openlr:NegOff>
    </openlr:Offsets>
    <openlr:SideOfRoad>LEFT</openlr:SideOfRoad>
    <openlr:Orientation>NO_ORIENTATION_OR_UNKNOWN</openlr:Orientation>
    </openlr:PointAlongLine>
</openlr:PointLocationReference>
</openlr:XMLLocationReference>
</openlr:OpenLR>
    
```

13.2.3.2 Decoding

The decoding process starts with a validation of the physical representation of the location reference. If the input data is not valid the decoder should report an error.

After the validation phase the decoder tries to find network nodes which match the coordinates for the location reference points. The first location reference point should result in node 5 and the second location reference point should result in node 7.

In the next step the decoder resolves for each location reference point a line which is an outgoing line for all location reference points except the last one and an incoming line for the last location reference point whereby these lines match the given LRP attributes. A ranking of these lines will determine a line which matches best. In the example the best matching line for both location reference points is the line between the nodes 5 and 7 (see Figure 42).

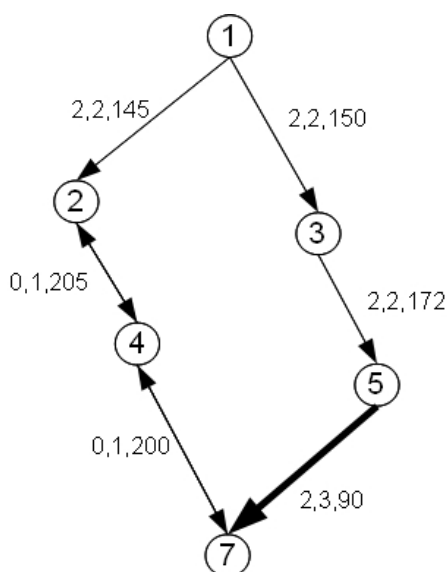


Figure 42: PointAlongLine - decoded reference line

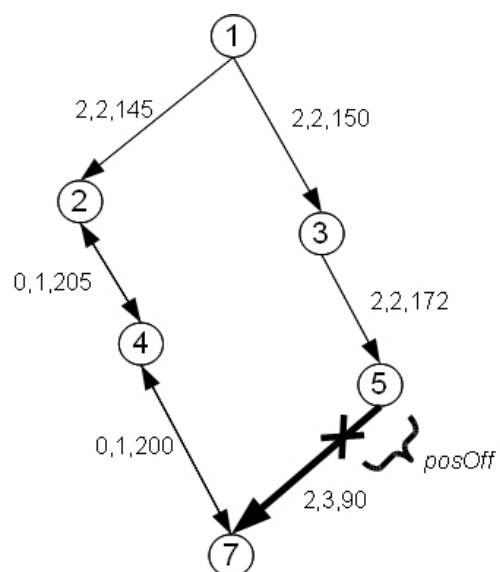


Figure 43: PointAlongLine - decoded offset



If the best matching lines for the first and second location reference point are not the same then the decoder should calculate the shortest path between these two lines. The resulting path shall be used in the next steps.

After having determined the path between the first and second LRP the offset value needs to be calculated. If the binary physical format version 3 is used the decoder would need to determine the length of this path. This length is used to resolve the percentage value into a meter value. The offset percentage value is 30.27% and the resulting offset length is 27.8 meter (see Figure 43).

The values for SideOfRoad and Orientation are read from the physical format and the decoded PointAlongLine location is shown in Figure 44.

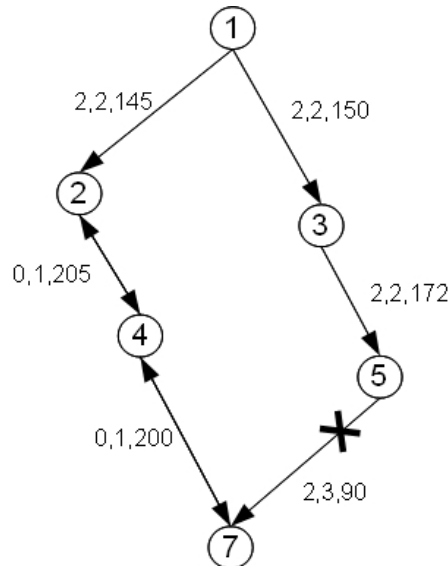


Figure 44: Decoded PointAlongLine location

13.2.4. PoiWithAccessPoint

The PoiWithAccessPoint defines a point of interest as a geo coordinate and an access point to that point directly on a line or near by a line (identified by a network line and an offset value). In the example shown in Figure 45 the point of interest is marked with a small black cross and the access point with a large black cross. The distance between the start of the line and the access point shall be 28 meter or 30.43% of the line length. The coordinates of the point of interest are shown in Table 89. The additional information about the side of the road and the orientation are mentioned in Table 90.

| | Longitude | Latitude |
|---|-----------|-----------|
| X | 6.12699° | 49.60728° |

Table 89: Example coordinates

| Information type | Value |
|------------------|---------------------------|
| SideOfRoad | LEFT |
| Orientation | NO_ORIENTATION_OR_UNKNOWN |

Table 90: Additional information for poi with access point

13.2.4.1 Encoding

The encoding of a PoiWithAccessPoint location starts with validating the input data. The access point of the point location shown in Figure 45 needs to be translated into a network line and an offset value. The access point is on the determined line and the offset describes the distance between the

start node of that line and the point location (see Figure 46). The point of interest is represented by the geo coordinate.

The geo coordinate and the line including the offset value will be the input for the OpenLR™ encoder and needs to be checked. The offset value shall not exceed the length of the line and the coordinate values shall not exceed the valid boundaries for longitude and latitude values. If the validation of the input data fails then the encoder shall report an error.

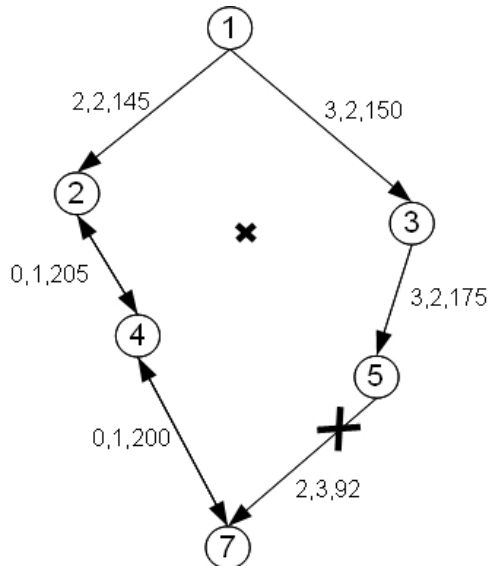


Figure 45: Example - PoiWithAccessPoint

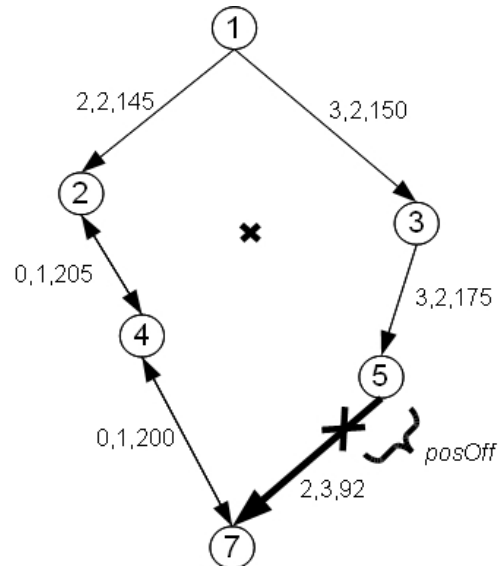


Figure 46: PoiWithAccessPoint - reference line and offset

If the input data is valid then the encoder would try to expand the referenced line in order to find valid nodes as start and end of the location. As all nodes are defined as valid the encoder can directly create two location reference points describing the reference line. The first LRP points to the start of the line (node 5 in Figure 41) and the second LRP (last LRP) points to the end of the reference line (node 7 in Figure 41). The line will therefore be referenced twice but one time in the direction from node 5 to node 7 and the second time against that direction.

Using two LRPs also provides the opportunity to use an artificial location reference point (not to be bound to a network node) in order to reduce the distance between the two location reference points and therefore to increase the accuracy of the offset.

In the last step the two location reference points, the offset value and the geo coordinate needs to be physically encoded.

Binary location reference

The following tables (Table 91 to Table 103) show the details of the binary physical format (version 3). The data size in this example is 21 bytes. The positive offset value of 30.43% falls into the bucket number 77. The geo coordinate (point of interest) will be referenced relatively to the coordinates of the first location reference point.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-----|----|-----|----|---------|---|---|
| Description | RFU | RFU | PF | ArF | AF | Version | | |
| Value | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

Table 91: Binary example: status byte



| Byte | First | | | | | | | | Second | | | | | | | | Third | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

Table 92: Binary example: LRP 1 – absolute longitude

| Byte | First | | | | | | | | Second | | | | | | | | Third | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|-------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | |

Table 93: Binary example: LRP1 – absolute latitude

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------|-------------|---|---|-----|---|---|-----|---|--|
| Description | Orientation | | | FRC | | | FOW | | |
| Value | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | |

Table 94: Binary example: LRP1 – attribute 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|--------|---|---|---|---------|---|---|---|
| Description | LFRCNP | | | | Bearing | | | |
| Value | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

Table 95: Binary example: LRP1 – attribute 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|---|---|---|---|---|---|---|
| Description | DNP | | | | | | | |
| Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 96: Binary example: LRP1 – attribute 3

| Byte | First | | | | | | | | Second | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

Table 97: Binary example: LRP2 – relative longitude

| Byte | First | | | | | | | | Second | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

Table 98: Binary example: LRP2 – relative latitude

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------|------------|---|-----|---|---|-----|---|---|--|
| Description | SideOfRoad | | FRC | | | FOW | | | |
| Value | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | |

Table 99: Binary example: LRP2 – attribute 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----|-------|-------|---------|---|---|---|---|
| Description | RFU | PoffF | NoffF | Bearing | | | | |
| Value | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

Table 100: Binary example: LRP3 – attribute 4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|------|---|---|---|---|---|---|---|
| Description | POFF | | | | | | | |
| Value | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |

Table 101: Binary example: positive Offset

| Byte | First | | | | | | | | Second | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 102: Binary example: POI – relative longitude

| Byte | First | | | | | | | | Second | | | | | | | |
|-------|-------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Table 103: Binary example: POI – relative latitude



XML location reference

The following XML location reference shows a PoiWithAccessPoint location consisting of exactly two location reference points, one (positive) offset and a geo coordinate.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<openlr:OpenLR xmlns="http://www.openlr.org/openlr">
  <openlr:LocationID>TestID03</openlr:LocationID>
  <openlr:XMLLocationReference>
    <openlr:PointLocationReference>
      <openlr:PoiWithAccessPoint>
        <openlr:LocationReferencePoint>
          <openlr:Coordinates>
            <openlr:Longitude>6.12829</openlr:Longitude>
            <openlr:Latitude>49.60597</openlr:Latitude>
          </openlr:Coordinates>
          <openlr:LineAttributes>
            <openlr:FRC>FRC2</openlr:FRC>
            <openlr:FOW>MULTIPLE_CARRIAGEWAY</openlr:FOW>
            <openlr:BEAR>202</openlr:BEAR>
          </openlr:LineAttributes>
          <openlr:PathAttributes>
            <openlr:LFR CNP>FRC2</openlr:LFR CNP>
            <openlr:DNP>92</openlr:DNP>
          </openlr:PathAttributes>
        </openlr:LocationReferencePoint>
        <openlr>LastLocationReferencePoint>
          <openlr:Coordinates>
            <openlr:Longitude>6.12779</openlr:Longitude>
            <openlr:Latitude>49.60521</openlr:Latitude>
          </openlr:Coordinates>
          <openlr:LineAttributes>
            <openlr:FRC>FRC2</openlr:FRC>
            <openlr:FOW>MULTIPLE_CARRIAGEWAY</openlr:FOW>
            <openlr:BEAR>42</openlr:BEAR>
          </openlr:LineAttributes>
        </openlr>LastLocationReferencePoint>
      <openlr:Offsets>
        <openlr:PosOff>28</openlr:PosOff>
        <openlr:NegOff>0</openlr:NegOff>
      </openlr:Offsets>
      <openlr:Coordinates>
        <openlr:Longitude>6.12699</openlr:Longitude>
        <openlr:Latitude>49.60728</openlr:Latitude>
      </openlr:Coordinates>
      <openlr:SideOfRoad>LEFT</openlr:SideOfRoad>
      <openlr:Orientation>NO_ORIENTA TION_OR_UNKNOWN</openlr:Orientation>
    </openlr:PoiWithAccessPoint>
  </openlr:PointLocationReference>
</openlr:XMLLocationReference>
</openlr:OpenLR>
```

13.2.4.2 Decoding

The decoding process starts with a validation of the physical representation of the location reference. If the input data is not valid the decoder should report an error.

After the validation phase the decoder tries to find network nodes which match the coordinates for the location reference points. The first location reference point should result in node 5 and the second location reference point should result in node 7.

In the next step the decoder resolves for each location reference point a line which is an outgoing line for all location reference points except the last one and an incoming line for the last location



reference point whereby these lines matches the given LRP attributes. A ranking of these lines will determine a line which matches best. In the example the best matching line for both location reference points is the line between the nodes 5 and 7 (see Figure 47).

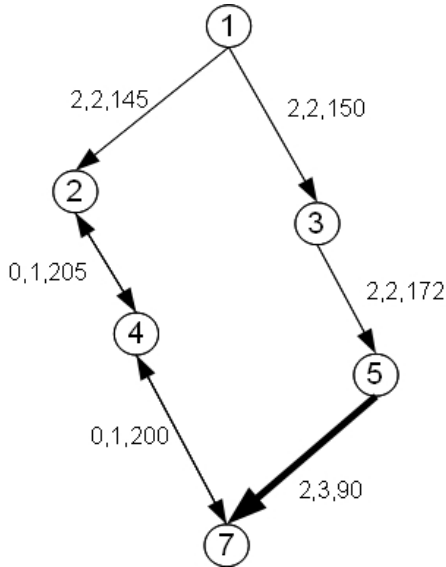


Figure 47: PoiWithAccess - decoded reference line

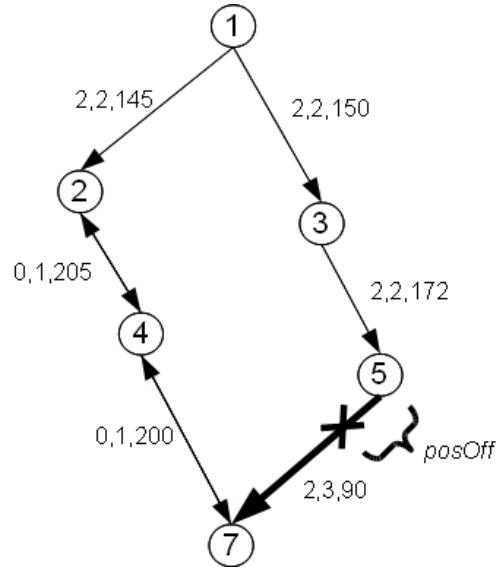


Figure 48: PoiWithAccess - decoded offset

If the best matching lines for the first and second location reference point are not the same then the decoder should calculate the shortest path between these two lines. The resulting path shall be used in the next steps.

After having determined the path between the first and second LRP the offset value needs to be calculated. If the binary physical format version 3 is used the decoder would need to determine the length of this path. This length is used to resolve the percentage value into a meter value. The offset percentage value is 30.27% and the resulting offset length is 27.8 meter (see Figure 48).

The geo coordinates of the point of interest are shown in Table 104. The values for SideOfRoad and Orientation are read from the physical format and the decoded PoiWithAccessPoint location is shown in Figure 49.

| | Longitude | Latitude |
|---|-----------|-----------|
| X | 6.12701° | 49.60727° |

Table 104: Decoded GeoCoordinate values

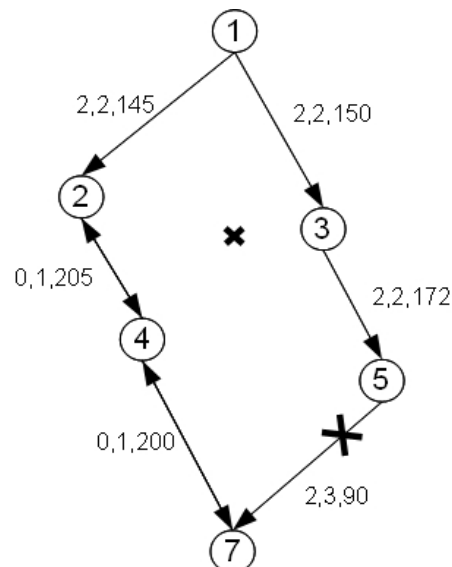


Figure 49: Decoded PoiWithaccessPoint location



Chapter I

Appendix

14. Appendix

List of figures

| | |
|--|-----------|
| Figure 1: Line location without offsets..... | 17 |
| Figure 2: Line location with offsets..... | 18 |
| Figure 3: Example - Geo-coordinate..... | 18 |
| Figure 4: Example - Point along line..... | 19 |
| Figure 5: Example - Poi with access point..... | 20 |
| Figure 6: Connection of LR-points..... | 25 |
| Figure 7: Logical format: Bearing point..... | 27 |
| Figure 8: Logical format: Bearing..... | 28 |
| Figure 9: Logical format: Distance to next point..... | 28 |
| Figure 10: Logical format: Positive and negative offset..... | 30 |
| Figure 11: Relationship: Attributes – LR-point..... | 30 |
| Figure 12: Invalid node (1)..... | 31 |
| Figure 13: Invalid node (2)..... | 31 |
| Figure 14: Offset placement and restrictions..... | 32 |
| Figure 15: Distances between longitudes..... | 36 |
| Figure 16: Physical format: Bearing sectors (Binary)..... | 38 |
| Figure 17: Distribution of the interval lengths..... | 42 |
| Figure 18: Interval lengths (percentage method)..... | 43 |
| Figure 19: XML structure overview..... | 64 |
| Figure 20: Workflow – OpenLR™ encoding (GeoCoordinate locations)..... | 70 |
| Figure 21: Workflow – OpenLR™ encoding (PointAlongLine and PoiWithAccessPoint locations).... | 71 |
| Figure 22: Workflow – OpenLR™ encoding (Line locations)..... | 72 |
| Figure 23: Offset calculation..... | 73 |
| Figure 24: Workflow – OpenLR™ decoding (Geo-Coordinate locations)..... | 77 |
| Figure 25: Workflow – OpenLR™ decoding (except Geo-Coordinate locations)..... | 78 |
| Figure 26: Example network..... | 81 |
| Figure 27: Location..... | 81 |
| Figure 28: Shortest path..... | 82 |
| Figure 29: Location reference points..... | 82 |
| Figure 30: Decoder map..... | 88 |
| Figure 31: Candidate nodes..... | 88 |
| Figure 32: Candidate lines..... | 89 |
| Figure 33: Location reference point - lines..... | 89 |
| Figure 34: Decoder - shortest path..... | 90 |
| Figure 35: Decoder location..... | 90 |
| Figure 36: Encoder map..... | 91 |
| Figure 37: Decoder map..... | 91 |
| Figure 38: Example – GeoCoordinate (encoder)..... | 91 |
| Figure 39: Example: Decoded GeoCoordinate..... | 93 |
| Figure 40: Example - PointAlongLine..... | 94 |
| Figure 41: PointAlongLine - Reference line and offset..... | 94 |
| Figure 42: PointAlongLine - decoded reference line..... | 96 |
| Figure 43: PointAlongLine - decoded offset..... | 96 |
| Figure 44: Decoded PointAlongLine location..... | 97 |
| Figure 45: Example - PoiWithAccessPoint..... | 98 |
| Figure 46: PoiWithAccessPoint - reference line and offset..... | 98 |
| Figure 47: PoiWithAccess - decoded reference line..... | 101 |
| Figure 48: PoiWithAccess - decoded offset..... | 101 |
| Figure 49: Decoded PoiWithaccessPoint location..... | 101 |

List of tables

| | |
|---|----|
| Table 1: Explanation of common terms..... | 14 |
| Table 2: Explanation of common abbreviations..... | 15 |



| | |
|---|----|
| Table 3: Data types (logical data format)..... | 25 |
| Table 4: Logical format: Functional road class | 26 |
| Table 5: Logical Format: Form of way | 27 |
| Table 6: Logical format: Parameter BEARDIST | 27 |
| Table 7: Orientation of a point location | 29 |
| Table 8: Data format rules overview | 32 |
| Table 9: Physical format: Data types (Binary) | 34 |
| Table 10: Physical format: Coordinate format (absolute) (Binary)..... | 35 |
| Table 11: Physical format: Coordinate format (relative) (Binary)..... | 35 |
| Table 12: Physical format: Longitude/Latitude ranges for relative coordinates (Example) | 35 |
| Table 13: Physical format: Functional road class (Binary)..... | 37 |
| Table 14: Physical format: Form of way (Binary)..... | 37 |
| Table 15: Physical format: Bearing (Binary) | 37 |
| Table 16: Physical format: Bearing value definition (Binary) | 38 |
| Table 17: Physical format: Distance to next point (Binary) | 38 |
| Table 18: Physical format: Lowest FRC to next point (Binary) | 39 |
| Table 19: Orientation of a point location | 39 |
| Table 20: Side of the road of a point location | 39 |
| Table 21: Physical format: Version (Binary) | 39 |
| Table 22: Physical format: Attribute flag (Binary) | 40 |
| Table 23: Physical format: Attribute flag values (Binary) | 40 |
| Table 24: Physical format: Area flag (Binary) | 40 |
| Table 25: Physical format: Area flag values (Binary)..... | 40 |
| Table 26: Physical format: Point flag (Binary)..... | 41 |
| Table 27: Physical format: Point flag values (Binary) | 41 |
| Table 28: Physical format: Offset flag (Binary) | 41 |
| Table 29: Physical format: Offset flag values (Binary)..... | 41 |
| Table 30: Physical format: Offset (Binary) | 42 |
| Table 31: Physical format: Offset (Binary) | 42 |
| Table 32: Line location with 2 LR-Points | 44 |
| Table 33: Line location with n location reference points (n>2) | 45 |
| Table 34: Message sizes depending on the number of LR-points | 46 |
| Table 35: Point location – geo-coordinate | 46 |
| Table 36: Point location – Point along line..... | 46 |
| Table 37: Point location – Network node | 47 |
| Table 38: Point location – Point of interest with access point..... | 47 |
| Table 39: Point location – Point of interest with access point at network node..... | 48 |
| Table 40: Identification depending on flags | 48 |
| Table 41: Identification depending on data size | 49 |
| Table 42: Physical format: Data types (XML) | 51 |
| Table 43: Physical format: Functional road class (XML) | 52 |
| Table 44: Physical format: Form of way (XML)..... | 52 |
| Table 45: Physical format: SideOfRoad values (XML) | 53 |
| Table 46: Orientation of a point location | 54 |
| Table 47: Encoding steps overview | 70 |
| Table 48: Decoding steps overview | 77 |
| Table 49: Example coordinates | 83 |
| Table 50: Example offset values..... | 83 |
| Table 51: Location reference points determined during encoding..... | 84 |
| Table 52: Binary example: status byte..... | 84 |
| Table 53: Binary example: LRP 1 – absolute longitude..... | 84 |
| Table 54: Binary example: LRP1 – absolute latitude..... | 84 |
| Table 55: Binary example: LRP1 – attribute 1 | 84 |
| Table 56: Binary example: LRP1 – attribute 2 | 84 |
| Table 57: Binary example: LRP1 – attribute 3 | 84 |
| Table 58: Binary example: LRP2 – relative longitude..... | 84 |
| Table 59: Binary example: LRP2 – relative latitude..... | 84 |
| Table 60: Binary example: LRP2 – attribute 1 | 85 |
| Table 61: Binary example: LRP2 – attribute 2 | 85 |
| Table 62: Binary example: LRP2 – attribute 3 | 85 |
| Table 63: Binary example: LRP3 – relative longitude..... | 85 |



| | |
|--|-----|
| Table 64: Binary example: LRP3 – relative latitude | 85 |
| Table 65: Binary example: LRP3 – attribute 1 | 85 |
| Table 66: Binary example: LRP3 – attribute 4 | 85 |
| Table 67: Binary example: positive Offset | 85 |
| Table 68: Decoded coordinates | 87 |
| Table 69: Decoded LRP information | 87 |
| Table 70: Decoded offset information | 87 |
| Table 71: Example data | 91 |
| Table 72: Example coordinates | 91 |
| Table 73: Binary example - GeoCoordinate: status byte | 92 |
| Table 74: Binary example - GeoCoordinate: Absolute longitude | 92 |
| Table 75: Binary example - GeoCoordinate: Absolute latitude | 92 |
| Table 76: Decoded GeoCoordinate values | 93 |
| Table 77: Additional information for point along line | 93 |
| Table 78: Binary example: status byte | 94 |
| Table 79: Binary example: LRP 1 – absolute longitude | 94 |
| Table 80: Binary example: LRP1 – absolute latitude | 94 |
| Table 81: Binary example: LRP1 – attribute 1 | 95 |
| Table 82: Binary example: LRP1 – attribute 2 | 95 |
| Table 83: Binary example: LRP1 – attribute 3 | 95 |
| Table 84: Binary example: LRP2 – relative longitude | 95 |
| Table 85: Binary example: LRP2 – relative latitude | 95 |
| Table 86: Binary example: LRP2 – attribute 1 | 95 |
| Table 87: Binary example: LRP3 – attribute 4 | 95 |
| Table 88: Binary example: positive Offset | 95 |
| Table 89: Example coordinates | 97 |
| Table 90: Additional information for poi with access point | 97 |
| Table 91: Binary example: status byte | 98 |
| Table 92: Binary example: LRP 1 – absolute longitude | 99 |
| Table 93: Binary example: LRP1 – absolute latitude | 99 |
| Table 94: Binary example: LRP1 – attribute 1 | 99 |
| Table 95: Binary example: LRP1 – attribute 2 | 99 |
| Table 96: Binary example: LRP1 – attribute 3 | 99 |
| Table 97: Binary example: LRP2 – relative longitude | 99 |
| Table 98: Binary example: LRP2 – relative latitude | 99 |
| Table 99: Binary example: LRP2 – attribute 1 | 99 |
| Table 100: Binary example: LRP3 – attribute 4 | 99 |
| Table 101: Binary example: positive Offset | 99 |
| Table 102: Binary example: POI – relative longitude | 99 |
| Table 103: Binary example: POI – relative latitude | 99 |
| Table 104: Decoded GeoCoordinate values | 101 |

List of equations

| | |
|--|----|
| Equation 1: Transformation from decimal coordinates into integer values (Binary) | 34 |
| Equation 2: Transformation from integer values into decimal coordinates (Binary) | 34 |
| Equation 3: Relative coordinates calculation (Binary) | 35 |
| Equation 4: Spherical law of cosines | 36 |
| Equation 5: Calculation of the maximum longitude range | 36 |
| Equation 6: Calculation of the bearing value (Binary) | 38 |
| Equation 7: Calculation of the DNP value (Binary) | 38 |
| Equation 8: Calculation of offset values (Binary) | 42 |
| Equation 9: Bucket index calculation (percentage method) | 43 |



An open standard for
encoding, transmitting
and decoding location
references in digital maps

Internet: <http://www.openlr.org>

Contact:

TomTom (Legal Department)
Email: legal@tomtom.com

TomTom (Technical contact)
Email: openlr@tomtom.com

Address: TomTom International B.V.
Oosterdoksstraat 114
1011DK Amsterdam
the Netherlands

Copyright © 2009, 2010 TomTom International B.V.

OpenLR™ is a trade mark of TomTom International B.V.