ETSI TS 103 544-25 V1.3.1 (2019-10)

TECHNICAL SPECIFICATION

Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 25: Navigation Meta Data Service

CAUTION

The present document has been submitted to ETSI as a PAS produced by CCC and approved by the ETSI Technical Committee Intelligent Transport Systems (ITS).

CCC is owner of the copyright of the document CCC-TS-084 and/or had all relevant rights and had assigned said rights to ETSI on an "as is basis". Consequently, to the fullest extent permitted by law, ETSI disclaims all warranties whether express, implied, statutory or otherwise including but not limited to merchantability, non-infringement of any intellectual property rights of third parties. No warranty is given about the accuracy and the completeness of the content of the present document.

Reference

RTS/ITS-98-25

Keywords

interface, ITS, PAS, smartphone

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

The present document is part 25 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.1].

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document is part of the MirrorLink® specification which specifies an interface for enabling remote user interaction of a mobile device via another device. The present document is written having a vehicle head-unit to interact with the mobile device in mind, but it will similarly apply for other devices, which provide a color display, audio input/output and user input mechanisms.

Current MirrorLink solutions are concentrated on utilization of MirrorLink Client's main display to mirror applications or provide variety services on the MirrorLink Server. However, there are so many MirrorLink Clients which have several other displays, such as cluster display panel, Heads-up Display (HUD) and so on. Instead of applications mirroring, using these displays, the driver and the passenger can be provided with a variety meta information such as turn by turn information, photo or graphic information, meta data information of audio and video clip, text information, etc. Those Meta Information Data Services are based on the SBP (Service Binary Protocol) framework.

The present document specifies navigation meta data service based on SBP (Service Binary Protocol) framework. By receiving this data, the MirrorLink Client (e.g. a car) can provide navigation information to driver and passenger e.g. through the car's cluster display panel, or heads-up display.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 103 544-27 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 27: Basic Meta Data Service".

[2] ETSI TS 103 544-6 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 6: Service Binary Protocol".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TS 103 544-1 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

HUD	Heads-Up Display
ICD	Instrument Cluster Display
SBP	Service Binary Protocol
UTC	Coordinated Universal Time

4 Data Service Definition

4.1 Navigation Meta Data Service Version 1.0

```
/** The present document defines data objects for the Navigation Meta
 *  data service to be carried over the SBP. By receiving this data,
 *  the MirrorLink Client (i.e. car) can provide variety navigation
 *  information to driver and passenger through instrument cluster
 *  display panel, HUD, etc.
 *  The service is based on the Basic Meta Information Data Service.
 *  @version 1.0
 */
SERVICE com.mirrorlink.meta.navigation
 : com.mirrorlink.meta.basic @version 1.0 {
/** Navigation route guidance possible statuses
 */
ENUM<INT> GuidanceState {
  /** Navigation guidance state is unknown
   */
  UNKNOWN = 0x00000000,
  /** Navigation guidance has no destination set
   */
  NO_DESTINATION_SET = 0x00000001,
  /** Navigation guidance system is calculating route
   */
  CALCULATING_ROUTE = 0x00000002,
  /** Navigation guidance system is using a new route
   */
  NEW_ROUTE = 0x00000003,
  /** Navigation guidance system state has no route to destination
   */
  NO_ROUTE = 0x00000004,
  /** Navigation guidance system is in normal operation
   */
  NORMAL_OPERATION = 0x00000005,
  /** Navigation guidance system positioning info is off road.
   *  maneuverDirection DIRECTION_TO_DESTINATION information should be
   *  provided, if available
   */
  OFF_ROAD = 0x00000006,
  /** Navigation guidance system positioning info is off map.
   *  maneuverDirection COMPASS information should be provided, if
   *  available.
   */
  OFF_MAP = 0x00000007,
  /** Navigation guidance system is within the destination area.
   *  maneuverDirection DIRECTION_TO_DESTINATION information should be
   *  provided, if available
   */
  DESTINATION_AREA = 0x00000008,
  /** Navigation guidance system has reached destination.
   *  maneuverDirection FINAL_DESTINATION information shall be
   *  provided.
   */
  DESTINATION_REACHED = 0x00000009
  };
/** Navigation route guidance active possible statuses.
 */
ENUM<INT> GuidanceActive {
  /** route guidance active on sink (usually a head unit)
   */
  GUIDANCE_CLIENT = 0xffffffff,
  /** no active route guidance
   */
  GUIDANCE_NONE = 0x00000000,
  /** route guidance active on source (usually a MirrorLink app)
   */
  GUIDANCE_SERVER = 0x00000001
  };
/** Definitions for NavigationNextManeuver#nextDirection
 */
ENUM<INT> ManeuverDirection {
  /** Next Direction: No symbol defined (blank screen).
   */
  NO_SYMBOL = 0x00000000,
  /** Next Direction: No information available (current direction).
   */
  NO_INFO = 0x00000001,
  /** Next Direction: Follow the street.
   */
  FOLLOW_STREET = 0x00000002,
  /** Next Direction: Turn straight.
   */
  TURN_STRAIGHT = 0x00000003,
  /** Next Direction: Slight right turn.
   */
  TURN_SLIGHT_RIGHT = 0x00000004,
  /** Next Direction: Slight left turn.
   */
  TURN_SLIGHT_LEFT = 0x00000005,
  /** Next Direction: Turn right
   */;
  TURN_RIGHT = 0x00000006,
  /** Next Direction: Turn left;
   */
  TURN_LEFT = 0x00000007,
  /** Next Direction: Sharp right turn.
   */
  TURN_SHARP_RIGHT = 0x00000008,
  /** Next Direction: Sharp left turn.
   */
  TURN_SHARP_LEFT = 0x00000009,
  /** Next Direction: Make a U-turn to the right.
   */
  UTURN_RIGHT = 0x0000000A,
  /** Next Direction: Make a U-turn to the left
   */
  UTURN_LEFT = 0x0000000B,
  /** Next Direction: Keep right.
   */
  KEEP_RIGHT = 0x0000000C,
  /** Next Direction: Keep left.
   */
  KEEP_LEFT = 0x0000000D,
  /** Next Direction: Exit to the right.
   */
  EXIT_RIGHT = 0x0000000E,
  /** Next Direction: Exit to the left.
   */
  EXIT_LEFT = 0x0000000F,
  /** Next Direction: Slight right and slight right again.
   */
  DOUBLE_TURN_SLIGHT_RIGHT_AND_SLIGHT_RIGHT_AGAIN = 0x00000010,
  /** Next Direction: Slight left and slight left again.
   */
  DOUBLE_TURN_SLIGHT_LEFT_AND_SLIGHT_LEFT_AGAIN = 0x00000011,
  /** Next Direction: Slight right and continue straight.
   */
  DOUBLE_TURN_SLIGHT_RIGHT_AND_STRAIGHT = 0x00000012,
  /** Next Direction: Slight left and continue straight.
   */
  DOUBLE_TURN_SLIGHT_LEFT_AND_STRAIGHT = 0x00000013,
  /** Next Direction: Turn right and right again.
   */
  DOUBLE_TURN_RIGHT_AND_RIGHT = 0x00000014,
  /** Next Direction: Turn left and left again.
   */
  DOUBLE_TURN_LEFT_AND_LEFT = 0x00000015,
  /** Next Direction: Turn right and then turn left.
   */
  DOUBLE_TURN_RIGHT_AND_LEFT = 0x00000016,
  /** Next Direction: Turn left and then turn right.
   */
  DOUBLE_TURN_LEFT_AND_RIGHT = 0x00000017,
  /** Next Direction: Merge.
   */
  MERGE = 0x00000018,
  /** Next Direction: Follow the Highway.
   */
  HIGHWAY_FOLLOW = 0x00000019,
  /** Next Direction: On highway, slight right.
   */
  HIGHWAY_SLIGHT_RIGHT = 0x0000001A,
  /** Next Direction: On highway, slight left.
   */
  HIGHWAY_SLIGHT_LEFT = 0x0000001B,
  /** Next Direction: On highway, slight left and then slight right.
   */
  HIGHWAY_DOUBLE_TURN_SLIGHT_RIGHT_AND_SLIGHT_RIGHT = 0x0000001C,
  /** Next Direction: On highway, slight left and then slight left.
   */
  HIGHWAY_DOUBLE_TURN_SLIGHT_LEFT_AND_SLIGHT_LEFT = 0x0000001D,
  /** Next Direction: On highway, slight right and then straight.
   */
  HIGHWAY_DOUBLE_TURN_SLIGHT_RIGHT_AND_STRAIGHT = 0x0000001E,
  /** Next Direction: On highway, slight left and then straight.
   */
  HIGHWAY_DOUBLE_TURN_SLIGHT_LEFT_AND_STRAIGHT = 0x0000001F,
  /** Next Direction: Michigan turn variant 1 to the right.
   */
  MICHIGAN_TURN_VARIANT_1_RIGHT = 0x00000020,
  /** Next Direction: Michigan turn variant 1 to the left.
   */
  MICHIGAN_TURN_VARIANT_1_LEFT = 0x00000021,
  /** Next Direction: Michigan turn variant 2 to the right.
   */
  MICHIGAN_TURN_VARIANT_2_RIGHT = 0x00000022,
  /** Next Direction: Michigan turn variant 2 left.
   */
  MICHIGAN_TURN_VARIANT_2_LEFT = 0x00000023,
  /** Next Direction: Enter tunnel.
   */
  TUNNEL_ENTER = 0x00000024,
  /** Next Direction: Continue in tunnel
   */
  TUNNEL = 0x00000025,
  /** Next Direction: Exit tunnel
   */
  TUNNEL_EXIT = 0x00000026,
  /** Next Direction: Enter ferry.
   */
  FERRY_ENTER = 0x00000027,
  /** Next Direction: Stay on ferry
   */
  FERRY = 0x00000028,
  /** Next Direction: Exit ferry
   */
  FERRY_EXIT = 0x00000029,
  /** Next Direction: Continue using public transportation
   */
  PUBLIC_TRANSPORTATION = 0x0000002A,
  /** Next Direction: Start walking
   */
  WALK = 0x0000002B,
  /** Next Direction: Compass; then angle to the north is given in
   *  nextAngle.
   */
  COMPASS = 0x0000002C,
  /** Next Direction: Destination is at <angle> degree. The angle is
   *  given in nextAngle.
   */
  DIRECTION_TO_DESTINATION = 0x0000002D,
  /** Next Direction: Exit roundabout to the right now.
   */
  ROUNDABOUT_RIGHT_EXIT_NOW = 0x0000002E,
  /** Next Direction: Exit roundabout to the left now.
   */
  ROUNDABOUT_LEFT_EXIT_NOW = 0x0000002F,
  /** Next Direction: Enter roundabout to the right. Exit is not
   *  known.
   */
  ROUNDABOUT_RIGHT_UNKNOWN_EXIT_NUMBER = 0x00000030,
  /** Next Direction: Enter roundabout to the right. Take exit,
   *  provided in navigationNextManeuver#index;
   */
  ROUNDABOUT_RIGHT_KNOWN_EXIT_NUMBER = 0x00000031,
  /** Next Direction: Enter roundabout to the left. Exit is not
   *  known.
   */
  ROUNDABOUT_LEFT_UNKNOWN_EXIT_NUMBER = 0x00000040,
  /** Next Direction: Enter roundabout to the left. Take exit,
   *  provided in navigationNextManeuver#index;
   */
  ROUNDABOUT_LEFT_KNOWN_EXIT_NUMBER = 0x00000041,
  /** Next Direction: Final destination.
   */
  FINAL_DESTINATION = 0x00000050,
  /** Next Direction: Final destination on the right.
   */
  FINAL_DESTINATION_ON_THE_RIGHT = 0x00000051,
  /** Next Direction: Final destination on the left.
   */
  FINAL_DESTINATION_ON_THE_LEFT = 0x00000052,
  /** Next Direction: Intermediate destination. The number of the
   *  intermediate destination is provided in
   * navigationNextManeuver#index.
   */
  INTERMEDIATE_DESTINATION = 0x00000053,
  /** Next Direction: Intermediate destination on the right.
   *  The number of the intermediate destination is provided in
   *  navigationNextManeuver#index.
   */
  INTERMEDIATE_DESTINATION_ON_THE_RIGHT = 0x00000054,
  /** Next Direction: Intermediate destination on the left.
   *  The number of the intermediate destination is provided in
   *  navigationNextManeuver#index.
   */
  INTERMEDIATE_DESTINATION_ON_THE_LEFT = 0x00000055
  };
/** The DistanceUnit enumeration defines the unit of a distance value.
 */
ENUM<INT> DistanceUnit {
  /** distance expressed in meters
   */
  METER = 0x0
  /** distance expressed in kilometer
   */
  KM = 0x1
  /** distance expressed in feet
   */
  FEET = 0x02
  /** distance expressed in yard
   */
  YARDS = 0x03
  /** distance expressed in miles
   */
  MILES = 0x04
  };
/** The LaneGuidanceArrowType enumeration contains the recommendation,
 *  whether the arrow should be shown.
 */
ENUM<BYTE> LaneGuidanceArrowType {
  /** Not shown
   */
  NOT_SHOWN = 0x00,
  /** Not recommended
   */
  NOT_RECOMMENDED = 0x01,
  /** Recommended
   */
  RECOMMENDED = 0x02,
  /** Best recommended
   */
  BEST_RECOMMENDED = 0x03
  };
/** The SpecialPurposeLane enumerations contains the defined special
 *  purpose lane types. A lane may have more than one special purpose.
 *  A regular (non-special purpose lane) shall have the value 0x00.
 */
ENUM<BYTE> SpecialPurposeLane {
  /** Non-special purpose lane: Regular lane, not having any special
   *  purpose as defined below.
   */
  REGULAR_LANE = 0x01,
  /** Special purpose lane: HOV lane for high occupancy vehicles.
   */
  HOV_LANE = 0x02,
  /** Special purpose lane: Toll lane.
   */
  TOLL_LANE = 0x03,
  /** Special purse lane: Temporary-use lane. This indicates a lane,
   *  which can be opened to traffic at certain times, e.g. during
   *  rush-hour.
   */
  TEMPOARY_LANE = 0x04
  };
/** The LaneGuidanceLineType enumeration contains defined type of
 *  lines, which are separating lanes.
 */
ENUM<BYTE> LaneGuidanceLineType {
  /** No line
   */
  NONE = 0x00,
  /** Solid Line
   */
  SOLID = 0x01,
  /** Dashed Line
   */
  DASHED = 0x02,
  /** Double solid Line
   */
  DOUBLE = 0x03,
  /** Barrier (non-crossable) or road limit
   */
  BARRIER = 0x04
  };
/** The enumeration contains the lane guidance arrow types. The values
 *  are bit mask values, i.e. a lane may show a combined arrow of two or
 *  more arrow types, e.g. TURN_RIGHT and TURN_LEFT.
 *  Each arrow type has a value defined in LaneGuidanceArrowType, which
 *  shall be bit shifted to completely fit into the bit mask.
 *  In addition, information on lane types and line types are provided.
 *  Bitfields not covered are reserved for future use.
 */
ENUM<INT> LaneGuidanceBitMask {
  /** Arrow Type: Turn straight; recommendation to show this arrow
   *  type, as defined in LaneGuidanceArrowType.
   */
  TURN_STRAIGHT = 0x00000003,
  /** Arrow Type: Turn slight right; recommendation to show this arrow
   *  type, as defined in LaneGuidanceArrowType << 2 (bit shift).
   */
  TURN_SLIGHT_RIGHT = 0x0000000C,
  /** Arrow Type: Turn slight left; recommendation to show this arrow
   *  type, as defined in LaneGuidanceArrowType << 4 (bit shift).
   */
  TURN_SLIGHT_LEFT = 0x00000030,
```