ETSI TS 103 544-28 V1.3.0 (2017-10)

TECHNICAL SPECIFICATION

**Publicly Available Specification (PAS);
Intelligent Transport Systems (ITS);
MirrorLink®;
Part 28: Weather 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).

Reference

DTS/ITS-88-28

Keywords

interface, ITS, PAS, smartphone

ETSI

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 only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

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 28 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 colour display, audio input/output and user input mechanisms.

The present document specifies weather data service based on SBP (Service Binary Protocol) framework. The service is used to provide weather data in car environments.

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 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 27: Basic Meta Data Service".

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 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".

3   Abbreviations

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

UV   Ultra Violet
SBP   Service Binary Protocol

4 Data Service Definition

4.1 Weather Data Service Version 1.0

```
/** This specification defines data objects for the weather data
 *  service to be carried over by SBP. By receiving this data, the
 *  weather data sink can provide weather information to the driver
 *  through instrument cluster display panel. The Weather Source may
 *  be implemented in a Mirrorlink Client or a Mirrorlink Server.
 *  @version 1.0
 */
SERVICE com.mirrorlink.meta.weather

    : com.mirrorlink.meta.basic version 1.0 {

/** The City object capture the city information for which weather
 *  data is captured
 *  @mandatory, @writable, @uid 0x18da0af0
 */
OBJECT City {

  /** Geographic latitude in degree; a positive value indicates
   *  north, a negative value south. Allowed values are -90 to 90.
   *  @mandatory, @uid 0x64f8f3f1
   */
  DOUBLE latitude;

  /** Geographic longitude in degree; a positive value indicates
   *  east, a negative value west. Allowed values are -180 to 180.
   *  @mandatory, @uid 0x7581892a
   */
  DOUBLE longitude;

  /** Height of the position above the coordinate system's ellipsoid
   *  in meter.
   *  @mandatory, @uid 0x970e9047
   */
  DOUBLE altitude;

  /** name of the city where weather data was captured.
   *  @optional, @writable, @uid 0xe9b422d0
   */
  STRING city;

  /** name of the country where weather data was captured.
   *  @optional, @writable, @uid 0x554bca71
   */
  STRING country;

  /** zipcode of the place where weather data was captured.
   *  @optional, @uid 0xb82fcc49
   */
  STRING zipcode;
  };

/** The temperature object captures the temperature information. When
 *  this object is supported all the parameters shall be supported.
 *  @optional, @uid 0xca0e732f
 */
OBJECT Temperature {

  /** temperature in degrees
```

```
 *   @mandatory, @uid 0x1ac4b311
 */
SHORT degree;

/** defines if temperature measured is in Celsius or not
 *   @mandatory, @uid 0xfa52da6f
 */
BOOLEAN isCelsius;

/** Maximum temperature of the day
 *   @mandatory, @uid 0x6ac64539
 */
SHORT maximumTemperature;
/** Minimum temperature of the day
 *   @mandatory, @uid 0x5c159fcb
 */
SHORT minimumTemperature;
};

/** The humidity object captures the humidity information in %
 *   @optional, @uid 0xa85a2c78
 */
OBJECT Humidity {

/** measured humidity value
 *   @mandatory, @uid 0x0a906458
 */
SHORT humidity;
};

/** The pressure object captures the pressure information in hPa
 *   @optional, @uid 0xf40c8faa
 */
OBJECT Pressure {

/** Air pressure value in hpa
 *   @mandatory, @uid 0x5642c78a
 */
SHORT pressure;
};

/** The wind object captures the wind information. When this object
 *   is supported at least one of the parameters shall be supported.
 *   @optional, @uid 0xbb5c584d
 */
OBJECT Wind {

/** Wind speed in km per hour or miles per hour
 *   @optional, @uid 0xda62001a
 */
SHORT windSpeed;

/** Wind speed type: 0 - km per hour; 1 - miles per hour
 *   @optional, @uid 0x709511c7
 */
BOOLEAN windType;

/** Wind direction in degrees ranging from 0 - 360, with 0
 *   pointing to North
 *   @optional, @uid 0xbc406cf0
 */
SHORT windsFrom;
};

/** The Rain object captures the rain information. When this object
```

```
 *  is supported at least one of the parameters shall be supported.
 *  @optional, @uid 0xcec4c759
 */
OBJECT Rain {

  /** Probability of rainfall with range 0 ~ 100
   *  @optional, @uid 0xcb15db4c
   */
  SHORT rainfallProbablity;

  /** If there is Rain or Snow expected
   *  @optional, @uid 0x7c0cac89
   */
  BOOLEAN isRainorSnow;

  /** rainfall rate in inch or mm
   *  @optional, @uid 0x8a055334
   */
  FLOAT rainfallRate;

  /** rainfall rate type: mm (false), Inch (true)
   *  @optional, @uid 0x54178dce
   */
  BOOLEAN rainfallRateType;
  };

/** The Misc_Weather object captures the misc. weather information
 *  @optional, @uid 0x8cdb2d46
 */
OBJECT Misc_Weather {

  /** measured UV Index value ranging between 0~11
   *  @optional, @uid 0x4af8e8cc
   */
  SHORT uvIndex;

  /** Percentage of cloud cover with range 0 ~ 100
   *  @optional, @uid 0x9ab38d87
   */
  SHORT cloudCover;
  };

/** The Alerts object captures the extreme weather alerts
 *  @mandatory, @uid 0x912b62bc
 */
OBJECT Alerts {

  /** Extreme weather Level ranging from 1 ~ 5, with 5 being extreme
   *  @mandatory, @uid 0x4ba58f8d
   */
  SHORT alertLevel;

  /** Descriptive text related to extreme weather
   *  @optional, @uid 0xf9181f24
   */
  STRING alertText;
  };

/** The WeatherConfig object is set from data sink, to configure the
 *  behavior of data source for limiting the return results. Data Sink
 *  shall choose either maxNoOfItems or startTime and endTime.
 *  @mandatory, @uid 0x91f7a331
 */
```

```
OBJECT WeatherConfig {

  /** Maximum number of items to be returned
   *  @optional, @uid 0x577142c9
   */
  INT maxNoOfItems;
  /** UTC start time of capture
   *  @optional, @uid 0x92b68a6a
   */
  TIME startTime;
  /** UTC end time of capture
   *  @optional, @uid 0xa9a217e3
   */
  TIME endTime;
  };

/** The TimeRange object is set to identify real-time data or
 *  historical data. Depending on type either time is mandatory or
 *  startTime and endTime is mandatory.
 *  @mandatory, @uid 0x447c6a8b
 */
OBJECT TimeRange {

  /** UTC time when the weather data was acquired
   *  @optional, @uid 0x00a0fdb2
   */
   TIME time;

  /** UTC start time of capture
   *  @optional, @uid 0x92b68a6a
   */
  TIME startTime;
  /** UTC end time of capture
   *  @optional, @uid 0xa9a217e3
   */
  TIME endTime;
  };

};
```

5   SBP Binding

A SBP Sink endpoint shall be able to access the weather service objects using SBP *Subscribe* and *Get* commands. A SBP Source endpoint shall support both the REGULAR interval and ON_CHANGE SBP subscription types depending on the object.

If the SBP Source endpoint is not able to retrieve weather information temporarily, the SBP Source endpoint shall have the following behaviour within the SBP protocol:

- The SBP Source endpoint shall return an SBP *response* message with a "Not available" SBP error code in response to a SBP *Get* command to the weather related objects for e.g. Temperature, Humidity etc.

- The SBP Source endpoint shall send an SBP *response* message with a "Not available" SBP error code in regular intervals when the SBP Sink endpoint has subscribed to the weather service objects. The SBP Source shall provide a valid objects, as soon as the weather information becomes available again.

- The SBP Source endpoint shall send an SBP *response* message with a "Not available" SBP error code in response to a SBP *Subscribe* command to the weather service objects. The SBP Sink endpoint should then send a new *Subscribe* command again not earlier than 5s and not later than 30s to receive notifications.

The Weather Meta Data Services uses the following objects and their access capabilities, as defined in [1]:

name / uid	accessType	subscriptionType	minIntervalTime	MaxIntervalTime
City	WRITABLE	NONE	N/A	N/A
Temperature	READABLE	REGULAR		
Humidity	READABLE	REGULAR		
Pressure	READABLE	REGULAR		
Wind	READABLE	REGULAR		
Rain	READABLE	REGULAR		
Misc_Weather	READABLE	REGULAR		
Alert	READABLE	ON_CHANGE	NA	NA
WeatherConfig	WRITABLE	NONE	NA	NA
TimeRange	WRITABLE	NONE	NA	NA

6   Theory of Operations

6.1   Getting Weather updates

The following sequence diagram shows how a weather data sink retrieves metadata about the current weather condition from the weather data source.
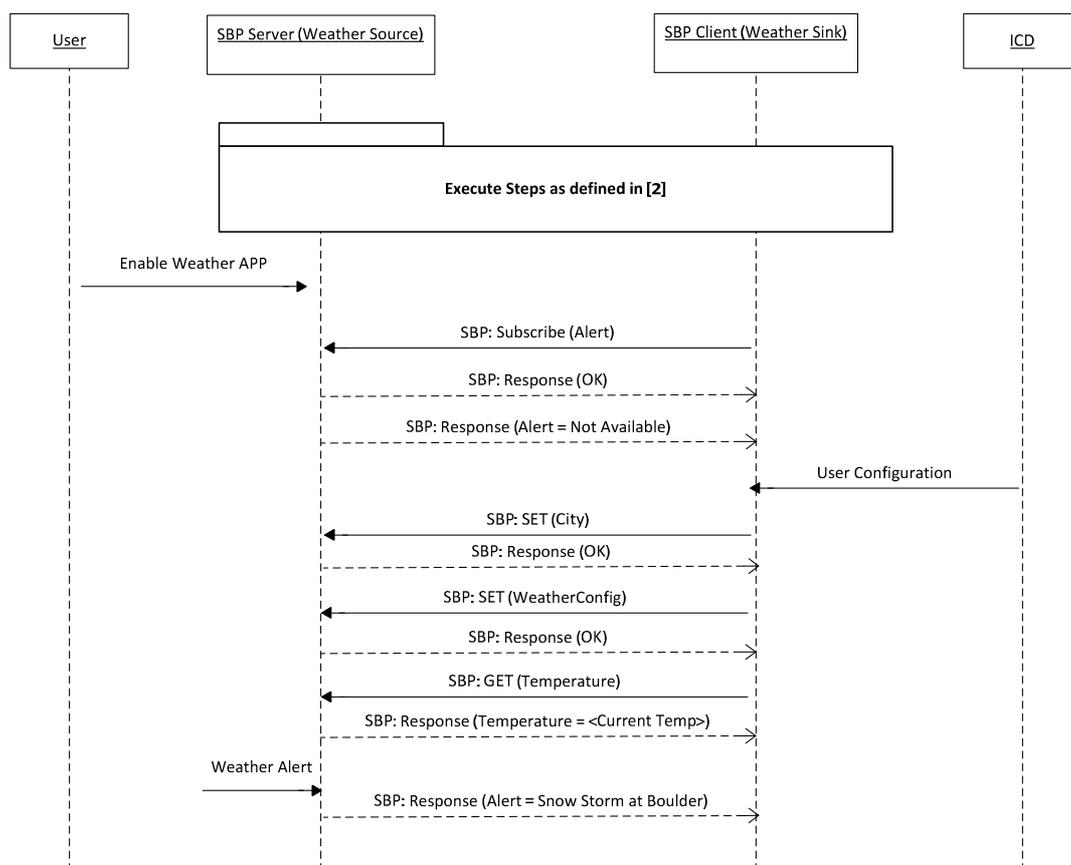
Figure 1: Message Sequence Diagram – Retrieve weather metadata

It consists of the following steps, after the data service has been setup as defined in [1]:

1) Weather Sink sends an SBP: *Subscribe* message for the *Alert* object; Subscription is ON_CHANGE. The Weather Source responds first with an SBP: *Response* message confirming the *Subscribe* message, followed by a second SBP: *Response* message containing a "Not Available" error code, as no alert is currently available.

NOTE: In case an alert is already available, the Weather Sink will return a valid Alert object, representing the current Alert.

2) User wants to see weather at a particular city en-route. Weather Sink sends SBP: *SET* message to the *City* object. The Weather Source responds with an SBP *Response* message confirming the setting.

3) Weather Sink further sends SBP: *SET* message to the *WeatherConfig* object to set filtering conditions on retrieved data. The Weather Source responds with an SBP: *Response* message confirming the setting.

4) Weather Sink sends SBP: *GET* message to the *Temperature* object. The Weather Source responds with the SBP: *Response* message, containing the Temperature object, representing the current temperature.

5) In the meanwhile, there is an alert for severe snow storm en-route. Weather Source sends an SBP: *Response* message for the *Alert*, containing the Alert object, representing the current alert.

The Weather Sink can subscribe to all weather related objects e.g. *Temperature*, *Humidity* etc., which will contain the current weather information. Any updates in the objects are sent on a regular basis to the Weather Sink. It is also possible for a Weather Sink to retrieve Historical data by setting the range of such historical data in the *TimeRange* object.

Annex A (informative):
Authors and Contributors

The following people have contributed to the present document:

Rapporteur:	Dr. Jörg Brakensiek, E-Qualus (for Car Connectivity Consortium LLC)
Other contributors:	Kiran Vedula, Samsung Electronics
	Lee Poong-Seok, Samsung Electronics

History

Document history		
V1.3.0	October 2017	Publication