

# RW BLE Time Profile (TIP) Interface Specification

---

Interface Specification

RW-BLE-PRF-TIP-IS

Version 8.0

2015-07-29

---

## 1 Revision History

Version	Date	Revision Description	Author
0.1	May 30 <sup>nd</sup> 2012	Initial Release	LT
0.2	June 21 <sup>th</sup> 2012	API Messages Update	LT
1.0	August 13th 2012	API Messages Update	LT
2.0	December 3 <sup>rd</sup> 2012	Client Multi-Instance API	LT
7.0	December 1 <sup>st</sup> 2014	Update to BLE 4.1	CM
8.0	July 29 <sup>th</sup> 2015	Update to BLE 4.2	CM



## 2 Table of Contents

1	Revision History .....	2
2	Table of Contents.....	3
3	Overview .....	4
3.1	Document Overview .....	4
3.2	Protocol Overview .....	4
3.3	Firmware Implementation Overview .....	4
4	Time Profile Server .....	5
4.1	INITIALIZATION/DATA BASE CREATION .....	5
4.2	TIPS_ENABLE_REQ.....	6
4.3	TIPS_ENABLE_RSP.....	6
4.4	TIPS_UPD_CURR_TIME_REQ.....	6
4.5	TIPS_UPD_CURR_TIME_RSP.....	7
4.6	TIPS_RD_REQ_IND.....	8
4.7	TIPS_RD_CFM .....	8
4.8	TIPS_CURRENT_TIME_CCC_IND .....	9
4.9	TIPS_TIME_UPD_CTLN_PT_IND .....	9
5	Time Profile Client .....	10
5.1	TIPC_ENABLE_REQ .....	10
5.2	TIPC_ENABLE_RSP .....	11
5.3	TIPC_RD_CHAR_REQ .....	11
5.4	TIPC_RD_CHAR_RSP .....	12
5.5	TIPC_CT_NTF_CFG_REQ.....	12
5.6	TIPC_CT_NTF_CFG_RSP .....	13
5.7	TIPC_WR_TIME_UPD_CTLN_PT_REQ.....	13
5.8	TIPC_WR_TIME_UPD_CTLN_PT_RSP.....	13
5.9	TIPC_CT_IND.....	14
6	Miscellaneous .....	15
6.1	Error Codes .....	15
6.2	Services Structure .....	15
6.3	Reading Codes .....	16
6.4	Types .....	16
7	Abbreviations .....	20
8	References .....	21

## 3 Overview

### 3.1 Document Overview

This document describes the non-standard interface of the RW BLE Time Profile implementation. Along this document, the interface messages will be referred to as API messages for the profile block(s).

Their description will include their utility and reason for implementation for a better understanding of the user and the developer that may one day need to interface them from a higher application.

### 3.2 Protocol Overview

The Bluetooth Low Energy Time profile enables the user to obtain the current date and time, and related information such as time zone as exposed by the Current Time service in the peer device. Information of when next change of daylight savings time (DST) will occur can be retrieved from the peer exposed by the Next DST Change service. This profile also enables a device to request updating the time on the peer device as exposed by the Reference Time Update Service.

Within the profile, two roles can be supported: **Client** and **Server**. The Client must support the GAP Peripheral Role and the Server, the GAP Central role. The profile requires a connection to be established between the two devices for its functionality.

The functionality of a profile requires the presence of certain services and attributes on one of the two devices, which the other device can manipulate. In this case, the Time Server device must have one instance of the Current Time Service (CTS) in its attribute database; the Next DST Change Service (NDCS) and the Reference Time Update Service (RTUS) are optional. The Time Profile Client (TIPC) will discover these services and their characteristics, and it may then configure them to cause the Time Server (TIPS) to notify the Current Time value to the Client or to require a time update.

The various documents edited by the Bluetooth SIG present different use cases for this profile, their GATT, GAP and security, mandatory and optional requirements. The TIP profile and CTS, NDCS, RTUS services specifications have been adopted by the Bluetooth SIG on September 15<sup>th</sup> 2011 ([1], [2], [3], [4]). Their related Test Specifications have been released at the same time and are referenced in [5], [6], [7] and [8].

The profile is implemented in the RW-BLE software stack as two tasks, one for each role. Each task has an API decided after the study of the profile specifications and test specifications, and it is considered to be minimalistic and designed for a future application which would combine the profile functionality with the device connectivity and security procedures.

### 3.3 Firmware Implementation Overview

Basically, if a device needs only be Time Profile Server, the firmware should be compiled with this role only, and inversely for the Client role. The role enables the part of the DB, which, important to know, will be hidden by the Time Server until its role is enabled post-connection establishment.

The Applications which will control the roles on end-products are responsible with creating the connection between the devices, using suggested advertising intervals and data, connection intervals, security levels, etc. The Profile implementation allows modulating the behavior depending on the final needs. Profile role enabling should be immediate after connection creation in order to allow correct profile behavior with the peer device.

## 4 Time Profile Server

This role is meant to be activated on the device that acts as Time Server and sends time values to the Client. It implies it is a GAP Client. Please refer to “tips\_task.h” for implementation of this API.

This task only has two states, IDLE and BUSY.

**Important Note:** The TASK\_TIPS task is multi-instantiated, one instance is created for each connection for which the profile will be enabled and each of these instances will have a different task ID. Thus, it is very important for the application to keep the source task ID of the first received PASPS\_CMP\_EVT message to be able to communicate with the peer device linked to this task ID once it has been enabled.

The term TASK\_TIPS\_IDX will be used in the rest of the document to refer to any instance of the Time profile Server Role Task. The term TASK\_TIPS will refer to the first instance of this task.

### 4.1 INITIALIZATION/DATA BASE CREATION

During the initialization phase of the Time Sensor, the memory for this task must be allocated using the message GAPM\_PROFILE\_TASK\_ADD\_CMD provided by the GAPM interface. Apart from the security level, the following parameters should be filled:

**Parameters:**

Type	Parameters	Description
uint8_t	features	Indicate if optional features are supported or not

**Response:** GAPM\_PROFILE\_TASK\_ADDED\_IND

**Description:** This API message shall be used to add one instance of the Current Time Service, optionally one instance of the Next DST Change Service and one instance of the Reference Time Update Service. This should be done during the initialization phase of the device.

The feature parameter shall be used to indicate if an optional feature is supported or not. Value of this parameter shall be set using the following masks:

Name	Mask value	Description
TIPS_CTS_LOC_TIME_INFO_SUP	0x01	Indicate if CTS supports the Local Time Information characteristic
TIPS_CTS_REF_TIME_INFO_SUP	0x02	Indicate if CTS supports the Reference Time Information characteristic
TIPS_NDCS_SUP	0x04	Indicate if NDCS is supported
TIPS_RTUS_SUP	0x08	Indicate if RTUS is supported

## 4.2 TIPS\_ENABLE\_REQ

**Source:** TASK\_APP

**Destination:** TASK\_TIPS

**Required state:** IDLE

**Parameters:**

Type	Parameters	Description
uint16_t	current_time_ntf_en	Value stored for Current Time Notification Client Configuration Char.

**Response:** TIPS\_ENABLE\_RSP

### Description:

This API message is used for enabling the Time Server role of the Time profile. Before sending this message, a BLE connection shall exist with peer device. Application shall provide connection handle in order to activate the profile. Connection handle and Application task ID are saved within the role's environment.

Connection Type will determine if client configuration should be applied to the corresponding CTS Attributes in the database:

- Normal connection: Peer device is known and client configuration characteristics shall be restored.
- Discovery connection: Peer device is unknown and peer collector will manage client configuration characteristics.

## 4.3 TIPS\_ENABLE\_RSP

**Source:** TASK\_TIPS

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
uint8_t	status	Status of the operation

### Description:

This API message is used by the Time Server role to inform the application of the enable status operation.

## 4.4 TIPS\_UPD\_CURR\_TIME\_REQ

**Source:** TASK\_APP

**Destination:** TASK\_TIPS

**Required state:** IDLE

**Parameters:**

Type	Parameters	Description
struct tip_curr_time	current_time	Current Time characteristic value (see Current Time Structure (struct tip_curr_time))
uint8_t	enable_ntf_send	Define if a notification of new current time value will be send. See the 'Application' part below for more information 0: Disable, 1: Enable

**Response:** TIPS\_UPD\_CURR\_TIME\_RSP

**Description:**

This API message is used by the application for requesting an update of the Current Time characteristic value. If the connection handle is wrong, a TIPS\_UPD\_CURR\_TIME\_RSP with the error code is sent to the application. Else, the value is saved so that it can be read by the peer client; in this case and if requirements are met, this value will be notified to the peer device. The purpose of the enable\_ntf\_send parameters is exposed below.

**Application:**

The enable\_ntf\_send parameter shall be used to conform to the following CTS Specification requirement:

*"If the time of the Current Time Server is changed because of reference time update, then no notification shall be sent to Current Time Service Client within the 15 minutes from the last notification, unless one of both of the two statements below are true:*

- *The new time information differs by more than 1 minute from the Current Time Server time previous to the update.*
- *The update was caused by the client (interacting with another service)."*

The responsibility to respect this rule is left to the designer of the application through the enable\_ntf\_send parameter. Its role is to inform the handler of the TIPS\_UPD\_CURR\_TIME\_REQ if it is allowed to notify the Time Client about a new Current Time value in case where the Client Characteristic Configuration descriptor of the Current Time Characteristic would have been configured to allow notifications to be sent.

As shown in the figure below, the value of enable\_ntf\_send will be taken in account **only** if both of the two following features are fulfilled:

- The value of the Client Characteristic Configuration descriptor has been set to 0x01 by the Time Client.
- The 'External Reference Time Update' bit of adjust\_reason (bit 2) is set to 1.

## 4.5 TIPS\_UPD\_CURR\_TIME\_RSP

**Source:** TASK\_TIPS

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
uint8_t	status	Status of the operation

**Description:**

This API message is used by the Time Server role to inform the application of the enable status operation.

## 4.6 TIPS\_RD\_REQ\_IND

**Source:** TASK\_TIPS

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
uint8_t	char_code	Characteristic code to be read: <ul style="list-style-type: none"> <li><i>TIPS_CTS_CURR_TIME_SUP</i> = 0x00</li> <li><i>TIPS_CTS_LOC_TIME_INFO_SUP</i> = 0x01</li> <li><i>TIPS_CTS_REF_TIME_INFO_SUP</i> = 0x02</li> <li><i>TIPS_NDCS_SUP</i> = 0x04</li> <li><i>TIPS_RTUS_SUP</i> = 0x08</li> <li><i>TIPS_CTS_CURRENT_TIME_CFG</i> = 0x10</li> </ul>

**Description:** This API message informs the application that a peer device wants to read one, the application must answer to this request using the TIPS\_RD\_REQ\_CFM message followed by the correct parameters.

## 4.7 TIPS\_RD\_CFM

**Source:** TASK\_APP

**Destination:** TASK\_TIPS

**Parameters:**

Type	Parameters	Description
uint8_t	op_code	Operation code
union	value	
struct tip_curr_time	curr_time	Current Time
struct tip_loc_time_info	loc_time_info	Local Time Information
struct tip_ref_time_info	ref_time_info	Reference Time Information
struct tip_time_with_dst	time_with_dst	Time With DST
struct tip_time_upd_state	time_upd_state	Time Update State

**Response:** None

**Description:** This API message is used to send the requested data after the read command was correctly received by the device



## 4.8 TIPS\_CURRENT\_TIME\_CCC\_IND

**Source:** TASK\_TIPS

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
uint16_t	cfg_val	Value stored for Current Time Notification Client Configuration Char.

**Response:** None

**Description:**

This API message is used to inform the application about a modification of the Current Time Client Configuration characteristic value.

## 4.9 TIPS\_TIME\_UPD\_CTLN\_PT\_IND

**Source:** TASK\_TIPS

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
tip_time_upd_contr_pt	value	Time Update Control Point value written by the peer client

**Response:** None

**Description:**

This API message is used to inform the application about a modification of the Time Update Control Point Characteristic value.

## 5 Time Profile Client

This role is meant to be activated on the device that will collect the time values and information from the Time Server. It implies it is a GAP Central. The FW task for this role will discover the CTS (Mandatory), the NDCS (Optional) and the RTUS (Optional) present on the peer Server, after establishing connection, and will allow configuration of the CTS and RTUS attributes if so required. Please refer to “tipc\_task.h” for implementation of this API.

This task has 3 possible states: **IDLE, CONNECTED, DISCOVERING**.

**Important Note:** The TASK\_TIPC task is multi-instantiated, one instance is created for each connection for which the profile will be enabled and each of these instances will have a different task ID. Thus, it is very important for the application to keep the source task ID of the TIPC\_ENABLE\_CFM message to be able to communicate with the peer device linked to this task ID once it has been enabled.

The term TASK\_TIPC\_IDX will be used in the rest of the document to refer to any instance of the Battery Service Client Role Task. The term TASK\_TIPC will refer to the first instance of this task.

### 5.1 TIPC\_ENABLE\_REQ

**Source:** TASK\_APP

**Destination:** TASK\_TIPC

**Parameters:**

Type	Parameters	Description
uint8_t	con_type	Connection type
struct tipc_cts_content	cts	Existing handle values CTS (see Current Time Service Structure (struct tipc_cts_content))
struct tipc_ndcs_content	ndcs	Existing handle values NDCS (see Next DST Change Service Structure (struct tipc_ndcs_content))
struct tipc_rtus_content	rtus	Existing handle values RTUS (see Reference Time Update Service Structure (struct tipc_ndcs_content))

**Response:** TIPC\_ENABLE\_RSP

**Description:**

This API message is used for enabling the Client role of the Time profile. This Application message contains BLE connection handle, the connection type and the previously saved discovered CTS, NDCS and RTUS details on peer. The connection type may be PRF\_CON\_DISCOVERY for discovery/initial connection or PRF\_CON\_NORMAL for normal connection.

For a discovery connection, discovery of the peer CTS, NDCS and RTUS is started and the response will be sent at the end of the discovery with the discovered attribute details. Application shall save those information to reuse them for other connections. During normal connection, previously discovered device information can be reused.

For a normal connection, the response to this request is sent right away after saving the CTS, NDCS and RTUS content in the environment and registering TIPC in GATT to receive the notifications for the known attribute handle in CTS (Current Time) that would be notified.

## 5.2 TIPC\_ENABLE\_RSP

**Source:** TASK\_TIPC\_IDX

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
uint8_t	status	Enable status: discovery error code if anything goes wrong during a configuration type connection. (see RW Profiles Error Codes)
struct tipc_cts_content	cts	Existing handle values CTS (see Current Time Service Structure (struct tipc_cts_content))
struct tipc_ndcs_content	ndcs	Existing handle values NDCS (see Next DST Change Service Structure (struct tipc_ndcs_content))
struct tipc_rtus_content	rtus	Existing Handle values RTUS (see Reference Time Update Service Structure (struct tipc_ndcs_content))

**Description:**

This API message is used by the Client to either send the discovery results of CTS, NDCS or RTUS and confirm enabling of the Client role (status = PRF\_ERR\_OK), or to simply confirm enabling of Client role if it is a normal connection and the attribute details are already known (status = PRF\_ERR\_OK), or to inform the application that the discovery process has been stopped because of a missing attribute (status = PRF\_ERR\_STOP\_DISC\_CHAR\_MISSING).

## 5.3 TIPC\_RD\_CHAR\_REQ

**Source:** TASK\_APP

**Destination:** TASK\_TIPC\_IDX

**Parameters:**

Type	Parameters	Description
uint16_t	char_code	Code for which characteristic to read. (see Reading codes)

**Response:** TIPC\_RD\_CHAR\_RSP

**Description:**

This API message is used by the application to request sending of a GATT\_READ\_CHAR\_REQ with the parameters deduced from the char\_code. Upon reception of this message, TIPC checks whether the parameters are correct, then if the handle for the characteristic is valid (not 0x0000), the request is sent to GATT.

## 5.4 TIPC\_RD\_CHAR\_RSP

**Source:** TASK\_TIPC\_IDX

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
uint8_t	op_code	Operation code
uint8_t	status	Status of the operation
union	value	
uint16_t	ntf_cfg	Notification configuration
struct tip_curr_time	curr_time	Current Time
struct tip_loc_time_info	loc_time_info	Local Time Information
struct tip_ref_time_info	ref_time_info	Reference Time Information
struct tip_time_with_dst	time_with_dst	Time With DST
struct tip_time_upd_state	time_upd_state	Time Update State

**Description:**

This API message is used by the Client role to inform the application that the read operation was performed successfully. Note that the parameter will depend on the op\_code.

## 5.5 TIPC\_CT\_NTF\_CFG\_REQ

**Source:** TASK\_APP

**Destination:** TASK\_TIPC\_IDX

**Parameters:**

Type	Parameters	Description
uint16_t	cfg_val	Configuration value.

**Response:** TIPC\_CT\_NTF\_CFG\_RSP

**Description:**

This API message is used by the application to send a GATT\_WRITE\_CHAR\_REQ to the Current Time Client Configuration Characteristic Descriptor with the parameter cfg\_val.

When the peer has responded to GATT, and the response is routed to TIPC, the TIPC\_WR\_CHAR\_RSP message will be generically built and sent to Application. An error status is also possible either for the Write procedure or for the application request, in the second case, the TIPC\_CT\_NTF\_CFG\_RSP message is sent to Application with the appropriated error code.

## 5.6 TIPC\_CT\_NTF\_CFG\_RSP

**Source:** TASK\_TIPC\_IDX

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
uint8_t	status	Status of the operation

**Description:**

This API message informs the app about the status of the operation.

## 5.7 TIPC\_WR\_TIME\_UPD\_CTLN\_PT\_REQ

**Source:** TASK\_APP

**Destination:** TASK\_TIPC\_IDX

**Parameters:**

Type	Parameters	Description
tip_time_upd_contr_pt	value	Time Update Control Point value to write

**Response:** TIPC\_WR\_TIME\_UPD\_CTLN\_PT\_RSP

**Description:** This API message is used by the application to send a GATT\_WRITE\_CHAR\_REQ to the Time Update Control Point Characteristic. Upon reception of this message, TIPC checks whether the parameters are correct, then if the handle for the characteristic is valid (not 0x0000) and whether it is writable or not - if all OK, the request is sent to GATT, otherwise a TIPC\_WR\_TIME\_UPD\_CTLN\_PT\_RSP message is built for the Application with the error code. When the peer has responded to GATT, and the response is routed to TIPC, the TIPC\_WR\_CHAR\_RSP message will be generically built and sent to the Application. An error status is also possible for the Write procedure, it will be sent through the same message.

## 5.8 TIPC\_WR\_TIME\_UPD\_CTLN\_PT\_RSP

**Source:** TASK\_TIPC\_IDX

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
uint8_t	status	Error code: (see RW Profiles Error Codes)

**Description:** This API message is used by the Client role to inform the Application of a received write response. The status and the data from the write response are passed directly to Application, which must interpret them based on

the request it made.

## 5.9 TIPCT\_IND

**Source:** TASK\_TIPC\_IDX

**Destination:** TASK\_APP

**Parameters:**

Type	Parameters	Description
struct tip_curr_time	ct_val	Current Time Structure (see Current Time Structure (struct tip_curr_time))

**Description:**

This API message is used by the Client role to inform the Application of a notified current time value.

## 6 Miscellaneous

### 6.1 Error Codes

Name	Value	Description
PRF_ERR_OK	0x00	No error
PRF_ERR_INVALID_PARAM	0x80	Invalid parameter in request
PRF_ERR_INEXISTENT_HDL	0x81	Inexistent handle for sending a read/write characteristic request
PRF_ERR_STOP_DISC_CHAR_MISSING	0x82	Discovery stopped due to missing attribute according to specification
PRF_ERR_MULTIPLE_SVC	0x83	Too many SVC instances found-> protocol violation
PRF_ERR_STOP_DISC_WRONG_CHAR_PROP	0x84	Discovery stopped due to found attribute with incorrect properties
PRF_ERR_MULTIPLE_CHAR	0x85	Too many instance of a characteristic have been found -> protocol violation
PRF_ERR_NOT_WRITABLE	0x86	Attribute write not allowed
PRF_ERR_NOT_READABLE	0x87	Attribute read not allowed
PRF_ERR_REQ_DISALLOWED	0x88	Request not allowed
PRF_ERR_NTF_DISABLED	0x89	Notification Not Enabled
PRF_ERR_IND_DISABLED	0x8A	Indication Not Enabled
PRF_ERR_FEATURE_NOT_SUPPORTED	0x8B	Feature not supported by profile

Table 1: RW Profiles Error Codes

### 6.2 Services Structure

Type	Parameters	Description
struct prf_svc	svc	Service Info (see Service Handle Structure (struct prf_svc))
struct prf_char_inf	chars[0]	Current Time Characteristic (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[1]	Local Time Information Characteristic (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[2]	Reference Time Information Characteristic (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_desc_inf	descs[0]	Current Time Client Configuration Descriptor (see Descriptor Info Structure (struct prf_char_desc_inf))

Table 2: Current Time Service Structure (struct tipc\_cts\_content)

Type	Parameters	Description
struct prf_svc	svc	Service Info (see Service Handle Structure (struct prf_svc))
struct prf_char_inf	chars[0]	Time With DST Characteristic (see Characteristic Info Structure (struct prf_char_inf))

Table 3: Next DST Change Service Structure (struct tipc\_ndcs\_content)

Type	Parameters	Description
struct prf_svc	svc	Service Info (see Service Handle Structure (struct prf_svc))
struct prf_char_inf	chars[0]	Time Update Control Point Characteristic (see Characteristic Info Structure (struct prf_char_inf))
struct prf_char_inf	chars[1]	Time Update State Characteristic (see Characteristic Info Structure (struct prf_char_inf))

Table 4: Reference Time Update Service Structure (struct tipc\_ndcs\_content)

## 6.3 Reading Codes

Read Char. code	Description
TIPC_RD_CTS_CURR_TIME	Read CTS Current Time characteristic value
TIPC_RD_CTS_LOCAL_TIME_INFO	Read CTS Local Time Information characteristic value
TIPC_RD_CTS_REF_TIME_INFO	Read CTS Reference Time Information characteristic value
TIPC_RD_CTS_CURR_TIME_CLI_CFG	Read CTS Current Time characteristic Client Characteristic Configuration descriptor value
TIPC_RD_NDCS_TIME_WITH_DST	Read NDCS Time with DST characteristic value
TIPC_RD_RTUS_TIME_UPD_STATE	Read RTUS Time Update State characteristic value

Table 5: Reading codes

## 6.4 Types

Type	Parameters	Description
struct tip_exact_time_256	exact_time_256	Exact Time 256 Characteristic Structure (see Exact Time 256 Structure (struct tip_exact_time_256))
uint8_t	adjust_reason	Bit 0 : Manual Time Update Bit 1 : External Reference Time Update Bit 2 : Change of Time Zone Bit 3 : Change of DST 0 if no, 1 otherwise.

Table 6: Current Time Structure (struct tip\_curr\_time)

Type	Parameters	Description
struct tip_day_date_time	day_date_time	Day Date Time Characteristic Structure (see Day Date



		Time Structure (struct tip_day_date_time))
uint8_t	fraction_256	1/256 <sup>th</sup> of a second

**Table 7: Exact Time 256 Structure (struct tip\_exact\_time\_256)**

Type	Parameters	Description
struct prf_date_time	date_time	Date Time Characteristic Structure (see Time Stamp Structure (struct prf_date_time))
tip_day_of_week	day_of_week	Day of week Characteristic Structure (see Day of Week Type (tip_day_of_week))

**Table 8: Day Date Time Structure (struct tip\_day\_date\_time)**

Type	Declaration	Description
tip_day_of_week	typedef tip_day_of_week uint8_t	Day of the week (0-7) 0: Unknown, 1: Monday, 7: Sunday

**Table 9: Day of Week Type (tip\_day\_of\_week)**

Type	Parameters	Description
uint16_t	year	Year (1582, 9999) 0: Year is not known
uint8_t	month	Month (1, 12) 0: Month is not known 1: January, 12: December
uint8_t	day	Day (1, 31) 0: Day of moth is not known
uint8_t	hour	Hour (0, 23)
uint8_t	min	Minutes (0, 59)
uint8_t	sec	Seconds (0, 59)

**Table 10: Time Stamp Structure (struct prf\_date\_time)**

Type	Parameters	Description
tip_time_zone	time_zone	Time Zone Characteristic Structure (see Time Zone Type (tip_time_zone))
tip_dst_offset	dst_offset	DST Offset Characteristic Structure (see DST Offset Type (tip_dst_offset))

**Table 11: Local Time Information Structure (struct tip\_loc\_time\_inf)**

Type	Declaration	Description
------	-------------	-------------

tip_time_zone	typedef tip_time_zone int8_t	Time Zone (-48, 56) -128: Time Zone Offset is not known -48: UTC-12:00 56: UTC+14:00
---------------	------------------------------	---

**Table 12: Time Zone Type (tip\_time\_zone)**

Type	Declaration	Description
tip_dst_offset	typedef tip_dst_offset uint8_t	DST Offset (0, 8) 0: Standard 2: +0.5h 4: +1h 8: +2h 255: DST is not known

**Table 13: DST Offset Type (tip\_dst\_offset)**

Type	Parameters	Description
tip_time_source	time_source	Time Source Characteristic Structure (see Time Source Type (tip_time_source))
tip_time_accuracy	time_accuracy	Time Accuracy Characteristic Structure (see Time Accuracy Type (tip_time_accuracy))
uint8_t	days_update	Days since update of Reference Time Information (0, 254) 255: 255 or more days
uint8_t	hours_update	(0, 23) 255: 255 or more days

**Table 14: Reference Time Information Structure (struct ref\_time\_info)**

Type	Declaration	Description
tip_time_source	typedef tip_time_source uint8_t	(0, 6) 0: Unknown, 1: Network Time Protocol, 2: GPS, 3: Radio Time Signal, 4: Manual, 5: Atomic Clock, 6: Cellular Clock

**Table 15: Time Source Type (tip\_time\_source)**

Type	Declaration	Description
tip_time_accuracy	typedef tip_time_accuracy uint8_t	Accuracy (drift) of time information in steps of 1/8 of a second (125ms) compared to a reference time source. Valid range from 0 to 253 (0s to 31.5s). 254: Accuracy out of range, 255: Accuracy unknown

**Table 16: Time Accuracy Type (tip\_time\_accuracy)**

Type	Parameters	Description
struct prf_date_time	date_time	Date/Time of the next DST change
tip_dst_offset	dst_offset	DST offset to take in account at the next DST change

**Table 17: Time with DST Structure (struct tip\_time\_with\_dst)**

Type	Declaration	Description
time_upd_contr_pt	typedef tip_time_upd_contr_pt uint8_t	0x01: Get Reference Update 0x02: Cancel Reference Update

**Table 18: Time Update Control Point Type (time\_upd\_contr\_pt)**

Type	Parameters	Description
uint8_t	current_state	0x00: Idle 0x01: Update Pending
uint8_t	result	0x00: Successful 0x01: Canceled 0x02: No connection to reference 0x03: Reference responded with an error 0x04: Timeout 0x05: Update not attempted after reset

**Table 19: Time Update State Structure (struct tip\_upd\_state)**

Type	Parameters	Description
uint16_t	shdl	Start handle
uint16_t	ehdl	End handle

**Table 20: Service Handle Structure (struct prf\_svc)**

Type	Parameters	Description
uint16_t	char_hdl	Characteristic handle
uint16_t	val_hdl	Value handle
uint8_t	prop	Characteristic properties

**Table 21: Characteristic Info Structure (struct prf\_char\_inf )**

Type	Parameters	Description
uint16_t	desc_hdl	Descriptor handle

**Table 22: Descriptor Info Structure (struct prf\_char\_desc\_inf)**

## 7 Abbreviations

Abbreviation	Original Terminology
API	Application Programming Interface
BLE	Bluetooth Low Energy
CTS	Current Time Service
NDCS	Next DST Change Service
RTUS	Reference Time Update Service
RW	RivieraWaves
TIP	Time Profile
TIPC	Time Profile Client
TIPS	Time Profile Server

## 8 References

[1]	<b>Title</b>	Time Profile		
	<b>Reference</b>	TIP_SPEC_V10		
	<b>Version</b>	V10	<b>Date</b>	September 15 <sup>th</sup> 2011
	<b>Source</b>	Bluetooth SIG		

[2]	<b>Title</b>	Current Time Service		
	<b>Reference</b>	CTS_SPEC_V10		
	<b>Version</b>	V10	<b>Date</b>	September 15 <sup>th</sup> 2011
	<b>Source</b>	Bluetooth SIG		

[3]	<b>Title</b>	Next DST Change Service		
	<b>Reference</b>	NDCS_SPEC_V10		
	<b>Version</b>	V10	<b>Date</b>	September 15 <sup>th</sup> 2011
	<b>Source</b>	Bluetooth SIG		

[4]	<b>Title</b>	Reference Time Update Service		
	<b>Reference</b>	RTUS_SPEC_V10		
	<b>Version</b>	V10	<b>Date</b>	September 15 <sup>th</sup> 2011
	<b>Source</b>	Bluetooth SIG		

[5]	<b>Title</b>	Time (TIP) Profile Test Specification 1.0		
	<b>Reference</b>	TIP.TS.1.0.0		
	<b>Version</b>	1.0.0	<b>Date</b>	September 15 <sup>th</sup> 2011
	<b>Source</b>	Bluetooth SIG		

[6]	<b>Title</b>	Current Time Service (CTS) Test Specification 1.0		
	<b>Reference</b>	CTS.TS.1.0.0		
	<b>Version</b>	1.0.0	<b>Date</b>	September 15 <sup>th</sup> 2011
	<b>Source</b>	Bluetooth SIG		

[7]	<b>Title</b>	Next DST Change Service (NDCS) Test Specification 1.0		
	<b>Reference</b>	NDCS.TS.1.0.0		
	<b>Version</b>	1.0.0	<b>Date</b>	September 15 <sup>th</sup> 2011
	<b>Source</b>	Bluetooth SIG		

[8]	<b>Title</b>	Reference Time Update Information (RTUS) Test Specification 1.0		
	<b>Reference</b>	RTUS.TS.1.0.0		
	<b>Version</b>	1.0.0	<b>Date</b>	September 15 <sup>th</sup> 2011
	<b>Source</b>	Bluetooth SIG		