

RW BLE Weighing Scale Profile Interface Specification

Interface Specification

RW-BLE-ENVP-IS

Version 1

2016-05-11

Revision History

Version	Date	Revision Description	Author
1.0	May 11 th 2016	First Version	GF



Table of Contents

1	Overview.....	4
1.1	Document Overview	4
1.2	BLE Weighing Scale Profile Overview	4
1.3	Services of the Weighing Scale Profile	4
1.4	Weighing Scale Service Characteristics	5
1.4.1	Weigh Scale Feature	5
1.4.2	Weigh Scale Measurement.....	5
1.4.3	Client Characteristic Configuration (CCC) for Weighing Scale Measurements	6
2	API Messages	7
2.1	Server API.....	7
2.1.1	Initialization	7
2.1.2	WSCS_ENABLE_REQ.....	7
2.1.3	WSCS_ENABLE_RSP.....	7
2.1.4	WSCS_WR_CCC_IND	7
2.1.5	WSCS_MEAS_INDICATE_CMD.....	8
2.1.6	WSCS_CMP_EVT	8
2.2	Client API.....	9
2.2.1	WSCC_ENABLE_REQ	9
2.2.2	WSCC_ENABLE_RSP	9
2.2.3	WSCC_RD_FEATURE_REQ	10
2.2.4	WSCC_RD_FEATURE_RSP	10
2.2.5	WSCC_RD_CCC_REQ	10
2.2.6	WSCC_RD_CCC_RSP	10
2.2.7	WSCC_WR_CCC_CMD	11
2.2.8	WSCC_MEAS_IND.....	11
2.2.9	WSCC_CMP_EVT.....	12
3	Message Sequence Charts (MSCs).....	13
4	Abbreviations.....	15
5	References.....	16



1 Overview

1.1 Document Overview

This document describes the non-standard interface of the RivieraWaves (RW) Bluetooth Low Energy (BLE) Weight Scale Profile (WSP) implementation. Throughout 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.

1.2 BLE Weighing Scale Profile Overview

The WSP enables a collector device to connect and interact with a Bluetooth LE enabled Weighing Scale sharing its current and stored weight data.

This service has been implemented as a profile. Within this profile, two roles can be supported: Sensor role (WSPS) and Collector role (WSPC). The Collector role must support the GAP Central Role and the Sensor role, the GAP Peripheral role. The profile requires a connection to be established between the two devices for its functionality.

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 Weighing Scale Profile specifications have been adopted by the Bluetooth SIG on November 18th 2014 ([1] and [3]). Their related Test Specifications have been released and are referenced in [2] and [4].

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.

1.3 Services of the Weighing Scale Profile

The weighing scale profile includes a number of mandatory and optional services listed below :

- **Weighing Scale Service - mandatory**
- **Device Information Service – mandatory**
- **Current Time Service – optional**
- **Battery Service – optional**
- *Body Composition Service – optional*
- *User Data Service – optional, required for multi-user.*

The initial version of the RW weighing scale profile does not include the Body Composition and User Data services. These will be provided in further revisions.

The remainder of this document focuses on the support of the weighing scale service in the Collector and the Server. For details of the DIS, CTS and battery service support please read [5],[6],[7].

1.4 Weighing Scale Service Characteristics

The weighing scale service itself only supports two characteristics :- The Weight Scale Feature and The Weight Scale Measurement. The Weight Scale Measurement characteristic has an associated Client Characteristic Configuration (CCC). Overview of these characteristics is provided in the following sections.

1.4.1 Weigh Scale Feature

This is a 32bit number of which the first 9 bits are relevant. They describe the capabilities of the server . Table 1 shows the interpretation of bits of the Weigh Scale Feature Characteristic.

Name	Id in Code	Bit / Value
Time Stamp Supported	WSC_TIME_STAMP_SUPPORTED	0 (0 False, 1 True)
Multiple Users Supported	WSC_MULTIPLE_USERS_SUPPORTED	1 (0 False, 1 True)
BMI Supported	WSC_BMI_SUPPORTED_SUPPORTED	2 (0 False, 1 True)
Weight Measurement Resolution	WSC_WGHT_RESOLUTION_NOT_SPECIFIED WSC_WGHT_RESOLUTION_05kg_1lb WSC_WGHT_RESOLUTION_02kg_05lb WSC_WGHT_RESOLUTION_01kg_02lb WSC_WGHT_RESOLUTION_005kg_01lb WSC_WGHT_RESOLUTION_002kg_005lb WSC_WGHT_RESOLUTION_001kg_002lb WSC_WGHT_RESOLUTION_0005kg_001lb	bits 3-5 0 - none 1 - 0.5kg or 1 lb 2 - 0.2kg or .5 lb 3 - 0.1kg or .2 lb 4 - 0.05kg or .01lb 5 - 0.02kg or .005lb 6 - 0.01kg or .002lb 7 - 0.005kg or .001lb
Height Measurement Resolution	WSC_HGHT_RESOLUTION_NOT_SPECIFIED WSC_HGHT_RESOLUTION_001mtr_1inch WSC_HGHT_RESOLUTION_0005mtr_05inch WSC_HGHT_RESOLUTION_0001mtr_01inch	Bits 6-8 0 – none 1 - .001m or 1 inch 2 - .0005m or .5 inch 3 – .0001m or .1 inch

Table 1 - Format of the Weighing Scale Features.

The support of some of these capabilities on a server, is not dependent on support of additional services (with the exception of multi-user). The Time-Stamp feature can be supported without the server providing the Current Time Service. In this case, the management of update of the time is a local issue on the server. Similarly the BMI can be supported in the feature set without the server supporting the Body Composition Service.

1.4.2 Weigh Scale Measurement

This characteristic consists of a weight plus a set of optional fields. A flags field is included to indicate which optional components are valid in the message. The structure of the Weigh Scale Measurement is shown in Table 2 :

Name	Type	Mandatory Optional	Field on API messages
Flags	<i>uint8_t</i>	M	<i>flags</i>
Weight	<i>uint16_t</i>	M	<i>weight</i>
Time Stamp	<i>struct prf_date_time</i>	O	<i>time_stamp</i>
User Id	<i>uint8_t</i>	O	<i>user_id</i>
BMI	<i>uint16_t</i>	O	<i>bmi</i>
Height	<i>uint16_t</i>	O	<i>height</i>

Table 2 - Format of the Weighing Scale Measurement.

The flags field is complementary to the weighing scale features, and allows the server application to control what information it indicates to the client.

The format of the flags field is shown below in Table 3.

Name	Bit	Name on API
Measurement Units	0	(0 – Metric – Mass in Kg, Height in Meters) (1 – Imperial – Mass in Lb, Height in Feet)
Time stamp present	1	0 – Absent 1 – Present
User Id Present	2	0 – Absent 1 – Present
BMI and Height Present	3	0 – Absent 1 – Present

Table 3 :- Format of the Flags Field

1.4.3 Client Characteristic Configuration (CCC) for Weighing Scale Measurements

The Client Characteristic Configuration for Weighing Scale Measurements is used to enable/disable indications to the client of weight measurement value(s) on the server. Only indications are allowed for the measurement characteristic it cannot be explicitly read from the client.

2 API Messages

2.1 Server API

2.1.1 Initialization

During the initialization phase of the Weighing Scale Server, the memory for this task must be allocated using the message GAPM_PROFILE_TASK_ADD_CMD provided by the GAPM interface. This should provide the

Type	Parameters	Description
uint32_t	feature	Weigh Scale supported features. (see Table 1)

Description: This API message shall be used to add one instance of the Weigh Scale Service in the database.

2.1.2 WSCS_ENABLE_REQ

Source: TASK_APP

Destination: TASK_WSCS

Parameters:

Type	Parameters	Description
uint16_t	Ind_cfg	The initial value for the CCC for the Weigh Scale Characteristic.

Response: WSCS_ENABLE_RSP

Description: This message is responsible for restoring the Client Characteristic Configuration of the Weight Scale Characteristic to a previous value. If the CCC was configured on previous links, the application should track it value, and restore the value on establishment of subsequent links.

2.1.3 WSCS_ENABLE_RSP

Source: TASK_WSCS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Indicates success/failure of the previous WSCS_ENABLE_REQ

Response: none

Description :- Indicates the outcome of the previous WSCS_ENABLE_REQ

2.1.4 WSCS_WR_CCC_IND

Source: TASK_WSCS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
------	------------	-------------



uint16_t	Ind_cfg	The new value for the CCC for the Weigh Scale Characteristic
----------	---------	--

Response :- none

Description :- This message informs the Application that the peer Client has updated the CCC for the Weigh Scale Characteristic. The value assigned determines if Weight Measurements should be send to the peer client.

2.1.5 WSCS_MEAS_INDICATE_CMD

Source: TASK_APP

Destination: TASK_WSCS

Parameters:

Type	Parameters	Description
uint8_t	<i>flags</i>	Selects the valid contents of the message – if field is not selected in the flags field - it will not be transmitted to the collector. (for more info see Table 3)
uint16_t	<i>weight</i>	The weight value (in Kg or lb)
<i>struct prf_date_time</i>	<i>time_stamp</i>	Current time and date
uint8_t	<i>user_id</i>	Identified current user.. Only relevant if multi-user is support (not supported in the initial version of RW weigh scale profile)
uint16_t	<i>Bmi</i>	Body mass index
uint16_t	<i>Height</i>	Height Value (in Meters or Feet/inches)

Response :- WSCS_CMP_EVT

Description:

:-

This message is used by the application to provide an updated weight measurement to the collector/client. If the CCC has been enabled for indications to the peer then the contents of the message (selected by the flags field) will be sent to the peer. If the message is sent by the Application when the CCC is not configured for indications, and error will be returned in the WSCS_CMP_EVT.

The contents of the message transmitted are dependent on the flags field, which selects which fields of the message should be transmitted to the peer. However, if the flags field indicates an unsupported feature (according to the value of Weigh Scale Feature Characteristic), the corresponding field will not be transmitted and the flags field will be adjusted.

2.1.6 WSCS_CMP_EVT

Source: TASK_WSCS

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	operation	WSCS_MEAS_INDICATE_CMD_OP_CODE = 1
uint8_t	status	Indicates success or failure of the operation.

Response :- none

Description :- This message is used inform the user of the outcome of a preceding command. As currently the only command is the WSCS_MEAS_INDICATE_CMD this indicates completion of the transmission of a Weigh Scale Measurement Indication to the peer. If the user sends this message, when the CCC is not configured for indications, this event will be returned indicating error.

2.2 Client API

WSSC_MEAS_IND

WSSC_CMP_EVT

2.2.1 WSSC_ENABLE_REQ

Source: TASK_APP

Destination: TASK_ENVS

Parameters:

Type	Parameters	Description
uint8_t	con_type	Connection type: 1st discovery – 0 normal connection - 1
struct wssc_wss_content	wss	Existing handle values for WSS. (see Table 4)

Type	Parameters	Description
struct prf_svc	svc	The start and end handles for the Weigh Scale service.
struct prf_char_inf	chars[2]	Array of two structures which describe the Weigh Scale Feature Characteristic and the Weigh Scale Measurement Characteristic
struct prf_char_desc_inf	descs[0]	Weight Measurement Client Characteristic Configuration descriptor.

Table 4 :- Weigh Scale content structure (struct wssc_wss_content)

Response :- WSSC_ENABLE_RSP

Description :-

This API message is used for enabling the Collector role of the Weigh Scale profile. This Application message contains BLE connection handle, the connection type and the previously saved discovered WSS details on peer.

The connection type may be 0 = Connection for discovery/initial configuration or 1 = Normal connection. This parameter is used by Application to discover peer device services once at first connection. Application shall save information to reuse them for other connections. During normal connection, previously discovered information can be reused.

This is useful since most use cases allow Weigh Scale to disconnect the link once all measurements have been sent to Collector.

If it is a discovery /configuration type of connection, the WSS parameters are useless; they will be filled with 0's.

Otherwise the WSS will contain pertinent data which will be kept in the Collector environment while enabled. It allows for the Application operate transparently to attribute details.

For a normal connection, the response to this request is sent straight after saving the WSS content in the environment and registering WSS in GATT to receive the indications for the Weigh Scale Measurement handle in WSS that would be indicated. For a discovery connection, discovery of the peer WSS is started and the response will be sent at the end of the discovery with the discovered attribute details

2.2.2 WSSC_ENABLE_RSP

Source: TASK_WSSC

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Outcome of the WSCC_ENABLE_RSP, indicating success failure
struct wscw_wss_content	wss	Existing handle values for WSS. (see Table 4). Relevant if con_type in the WSCC_ENABLE_RSP was 0, otherwise WSS is all 0s.

Response :- none

Description :- Response to the WSCC_ENABLE_REQ which indicates the outcome of the request. If the con_type indicated a discovery connection (con_type = 0) then this message may contain wss field which contains the service handles and the information on the characteristics and descriptors of the service which can be reused on subsequent link.

2.2.3 WSCC_RD_FEATURE_REQ

Source: TASK_APP

Destination: TASK_WSCC

Parameters: none

Response :- WSCC_RD_FEATURES_RSP

Description

This message is used to read the Weigh Scale Features from the server.

2.2.4 WSCC_RD_FEATURE_RSP

Source: TASK_WSCC

Destination: TASK_APP

Parameters:

Type	Parameters	Description
Uin32_t	features	Only the first 9 bits are relevant – the format of the features is given in Table 1

Response :- none

Description

This message is used to deliver the Weigh Scale Features retrieved from the peer to the application. More information on the format of the features is given in section 1.4.1

2.2.5 WSCC_RD_CCC_REQ

Source: TASK_APP

Destination: TASK_WSCC

Parameters: none

Response :- WSCC_RD_CCC_RSP

Description

This message is used to read the current value of the Client Characteristic Configuration (CCC) Descriptor for the Weigh Scale Measurement characteristic from the server.

2.2.6 WSCC_RD_CCC_RSP

Source: TASK_WSCC

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint16_t	ccc	Only valid values are: 0x0000 - indications disabled 0x0002 - indications enabled

Response :- none

Description

:-

This message is used to deliver the value of the CCC for the Weigh Scale Measurement Characteristic retrieved from the peer to the application. The value of the CCC determines if the Weigh Scale server will indicate weight measurement values.

2.2.7 WSCC_WR_CCC_CMD

Source: TASK_APP

Destination: TASK_WSCC

Parameters:

Type	Parameters	Description
uint16_t	ccc	Only valid values are: 0x0000 - indications disabled 0x0002 - indications enabled

Response :- none

Description :- This message is used to change the CCC setting for the Weigh Scale Measurement Characteristic on the server. It is used to enable / disable weight measurement indications from the server.

2.2.8 WSCC_MEAS_IND

Source: TASK_WSCC

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	<i>Flags</i>	Selects the valid contents of the message – if field is not selected in the flags field. (for more info see Table 3)
uint16_t	<i>Weight</i>	The weight value (in Kg or lb)
<i>struct prf_date_time</i>	<i>time_stamp</i>	Current time and date
uint8_t	<i>user_id</i>	Identified current user.. Only relevant if multi-user is support (not supported in the initial version of RW weigh scale profile)
uint16_t	<i>Bmi</i>	Body mass index
uint16_t	<i>Height</i>	Height Value (in Meters or Feet/inches)

Response :- none

Description: :-

This message provides the application with an updated weight measurement from the server. This message will only be received if the Weigh Scale Measurement CCC on the server has been enabled for indications.

The contents of the message transmitted are dependent on the flags field, which selects which fields of the message should be transmitted to the peer.

2.2.9 WSCC_CMP_EVT

Source: TASK_WSCC

Destination: TASK_APP

Parameters:

Type	Parameters	Description
uint8_t	operation	WSCC_WRITE_CCC_OP_CODE= 3
uint8_t	status	Indicates success or failure of the operation.

Response :- none

Description :- This message is used to inform the user of the outcome of a preceding command. As currently the only command is the WSCC_WR_CCC_CMD this indicates completion of the command, indicating success or failure.

3 Message Sequence Charts (MSCs)

Following are a set of 2 simple message sequence charts which overview the operation of the Weigh Scale service and profile.

The first MSC is shown in figure 1 and show the setup of the Collector and Server following initialization. Following ACL setup the server is enabled using the WSCS_ENABLE_REQ/RSP messages. The WSCC_ENABLE_REQ is called by the Collector Application, which triggers discovery of the Weigh Scale Service if the con_type = 0 (first connection).

Following discovery, the collector reads the Weigh Scale Features using WSCC_RD_FEATURES_REQ/RSP. Next the Collector is shown reading the Client Characteristic Configuration (CCC) from the server, this may not be required as the value of the CCC may already be known, or can be explicitly written to the required value.

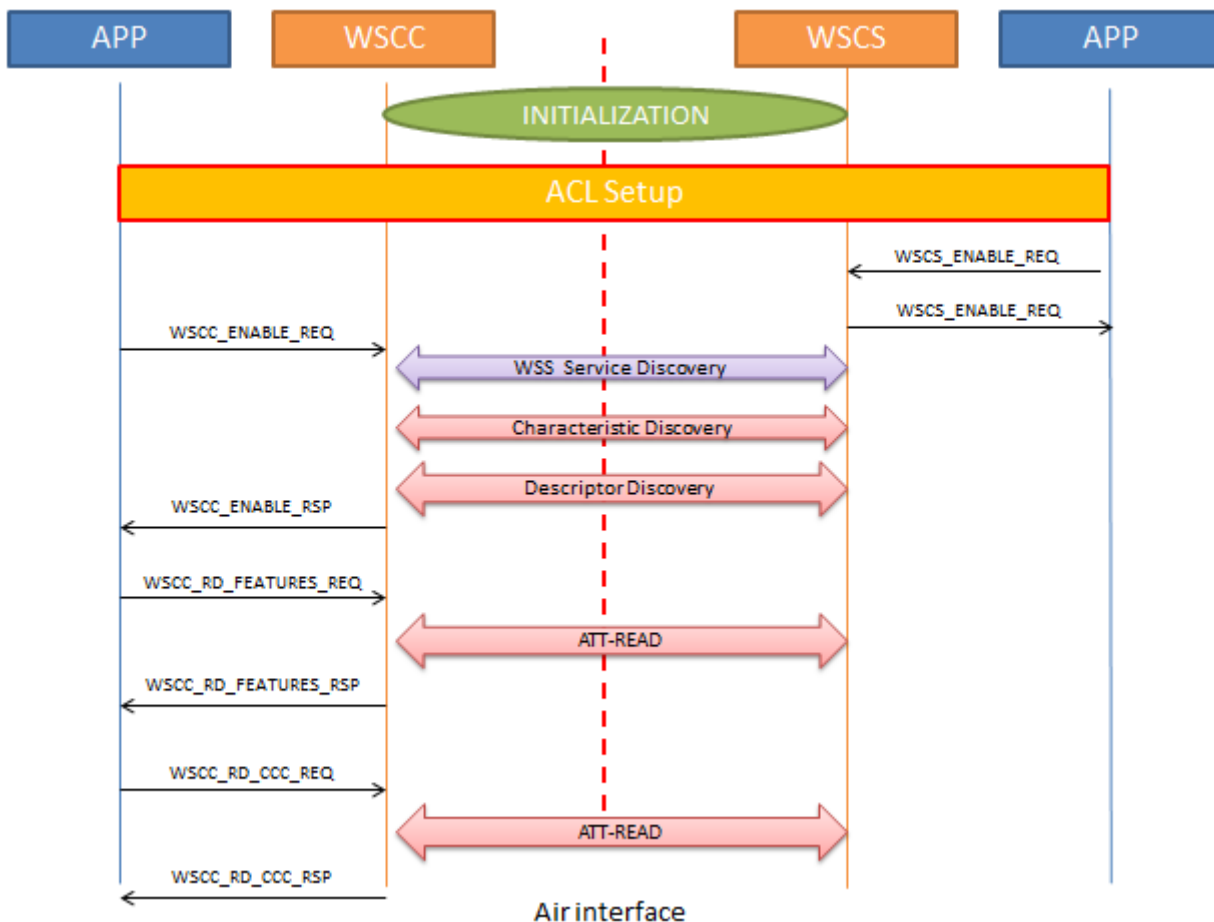
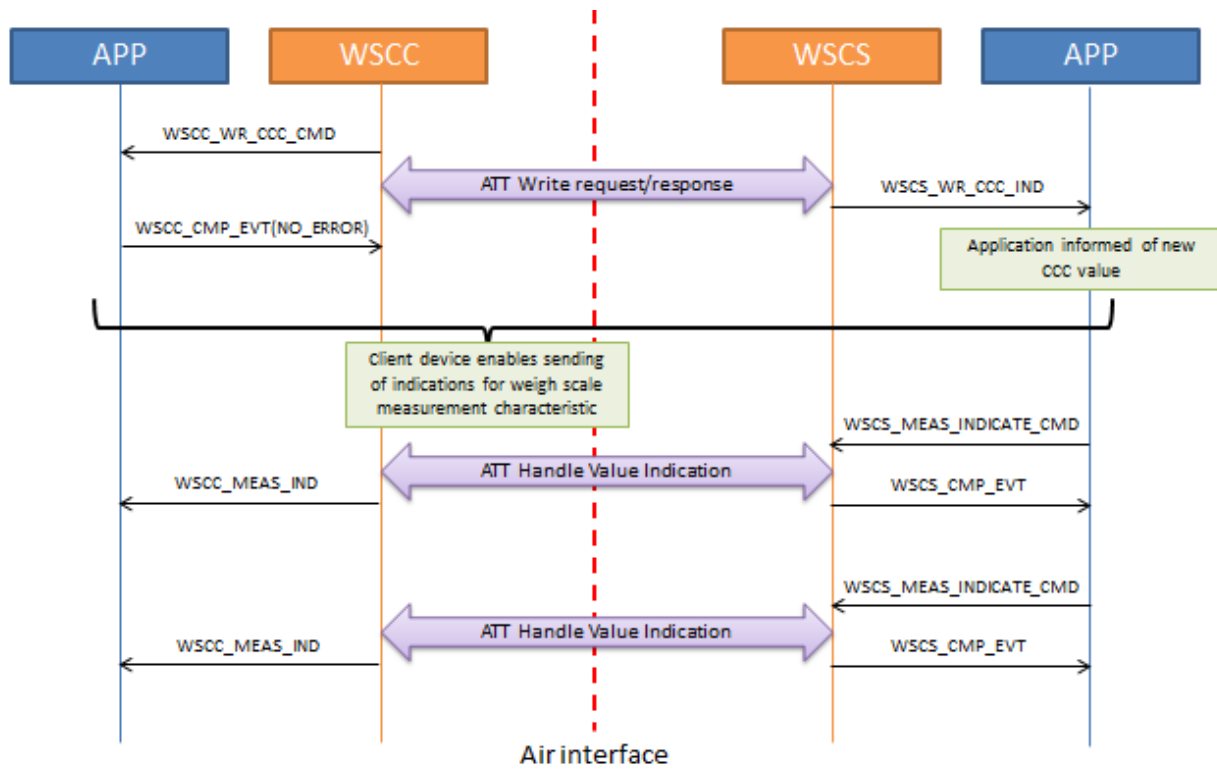


Figure 1 :- Setup of Weigh Scale Collector and Server

Figure 2 shows a continuation of the sequence. The Collector writes to the CCC for the Weigh Scale Measurement Characteristic to enable indications. The server application is informed of the new CCC value so it can store it in non-volatile memory for restoration on subsequent links.

Following the setting of the CCC, the Server Application is shown sending two weight measurements to the Collector/Client.





4 Abbreviations

Abbreviation	Original Terminology
API	Application Programming Interface
BLE	Bluetooth Low Energy
GAP	Generic Access Profile
GATT	Generic Attribute Profile
MSC	Message Sequence Chart
WSS	Weigh Scale Service
WSP	Weigh Scale Profile
WSCC	Weigh SCAle Collector
WSCS	Weigh SCAle Service
RW	RivieraWaves



5 References

[1]	Title	WEIGHT SCALE SERVICE		
	Reference	WSS_SPEC_V10		
	Version	1.0.0	Date	2014-10-21
	Source	Bluetooth SIG		

[2]	Title	WEIGHT SCALE PROFILE		
	Reference	WSP_SPEC_V10		
	Version	1.0.0	Date	2014-10-21
	Source	Bluetooth SIG		

[3]	Title	WEIGHT SCALE SERVICE – Test Suite Structure and Test Purpose		
	Reference	WSS_TEST_SPEC_V10		
	Version	1.0.0	Date	2014-10-21
	Source	Bluetooth SIG		

[4]	Title	WEIGHT SCALE PROFILE – Test Suite Structure and Test Purpose		
	Reference	WSS_TEST_SPEC_V102		
	Version	1.0.2	Date	2015-12-22
	Source	Bluetooth SIG		

[5]	Title	RW BLE Battery Service Interface Specification		
	Reference	RW_BLE_PRF_BAS_IS		
	Version	8.0.0	Date	2015-07-29
	Source	RivieraWaves		

[6]	Title	RW BLE Time Profile Interface Specification		
	Reference	RW_BLE_PRF_TIP_IS		
	Version	8.0.0	Date	2015-07-29
	Source	RivieraWaves		

[7]	Title	RW BLE Device Information Service Interface Specification		
	Reference	RW_BLE_PRF_DIS_IS		
	Version	1.0.2	Date	2015-12-22
	Source	RivieraWaves		