

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

**Operation Instructions for API Interface of Bluetooth
Measuring Equipments**

Documented By Yang Liang

Audited By _____

Approved By _____

Issued Date _____

Responsible Dept.	3 rd Party <input checked="" type="checkbox"/>
Revision History	

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

Edition	Description	Date
V0.1	Preliminary Connecting Method and Data Definition	2012-12-14
V0.2		2013-1-15
V0.3	The supported information category of the equipment is handled as an attribute of the device object category.	2013-1-22
V0.4	Merge the two version of the interface repository, add the supported equipment types (pedometer, height gague,kitchen scale)	2013-7-9
V0.5	The new supported equipment type: general BLE version body fat analyzer	2013-7-17

1. Overview

This API interface is mainly used to realize connection between portable equipments, which are installed with iOS, such as iPhone, iPad, and all kinds of measuring equipments produced by Lifesense, including body scale, body fat analyzer, blood pressure monitor, etc., via Bluetooth 4.0 technology, as well as collection of measurement data.

2. The Instruction of the supported Lifesense equipment types

2.1 The supported Lifesense categories of the equipments: body scale(A2 version), body fat analyzer(A2 version), pedometer(A2 version), height gague(A2 version), kitchen scale(general BLE version), body fat analyzer (general BLE version)

3. Service Restrictions

- iOS equipments that support Bluetooth 4.0
- iOS 5.0 or above

4. Instructions for API Interface Method

4.1 General Instructions of Function

LFHardwareConnector class realizes the core communication function of Bluetooth 4.0. You may use an example of LFHardwareConnector to search and connect to Lifesense health equipments, and acquire the measurement data stored in the health equipments as well. Information of Lifesense health equipments as well as the acquired measurement data is returned to the proxy class that has realized LFHardwareConnectorDelegate protocol through the methods in LFHardwareConnectorDelegate protocol. Therefore, when planning to develop applications that can communicate with Lifesense health equipments, LFHardwareConnector class and the methods in realizing LFHardwareConnectorDelegate protocol should be used.

4.2 Detailed Instructions of Class

LFHardwareConnector is derived from NSObject

Brief Instructions

LFHardwareConnector class utilizes CoreBluetooth.framework technological basis provided by iOS platform to realize Bluetooth 4.0 communication function of Lifesense health equipments. The example of LFHardwareConnector class plays the role of manager during the communication with the equipments. It is responsible to handle the process of searching equipments and to manage the searched equipments as well as the communication with the equipments. You may turn ON or OFF

<h1 style="text-align: center;">API Interface of Bluetooth Measuring Equipment</h1>		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

the equipment search function through the methods **startScanning** and **cancelScanning** in LfHardwareConnector class. You may use the method **setConnectorState** to set the status of the manager to inform it of the status of the equipments. You may set the equipment type that should be searched by the manager through the method **enableSensorType**. When an equipment is searched, you may apply the method **pairWithHardwareSensor** for request of pair-up with this equipment. When trying to connect to a searched equipment, the method **enqueueHardwareSensor** can be applied for adding this equipment into the waiting-for-connection list. The manager will connect to one equipment each time. Therefore, when there are several equipments in the queue, the manager will handle them in sequence. In the same time, apply the example of LfHardwareConnector class for Bluetooth communication with Lifesense health equipment. A proxy class that complies with LfHardwareConnectorDelegate protocol should be assigned for this example, for its convenience of feedback of acquired data information during communication by the methods in the protocol.

Task

Create and Initialize the Object

```
+ (LfHardwareConnector*) shareConnector;
-(id)init;
```

Configure the Status of the Connector

```
-(void)setConnectorState:(CONNECTOR_STATE)state;
```

Set the Type of the Equipment Handled by the CONNECTOR

```
-(void)enableSensorType:(LF_SENSOR_TYPE)type;
-(void)disableSensorType:(LF_SENSOR_TYPE)type;
-(void)enableAllSupportedSensorType;
```

Acquire All the Equipment Types Supported by the Current Connector

```
-(NSSet*)supportedSensorTypes;
```

Turn ON and OFF Bluetooth Scanning

```
-(void)startScanning;
-(void)cancelScanning;
```

Request for Pair-up of Equipment

```
-(void)pairWithHardwareSensor:(LfHardwareSensor*)sensor;
```

Place the Equipment into the Waiting Queue

```
-(void)enqueueHardwareSensor:(LfHardwareSensor*)sensor;
```

Set the Proxy Object of the Connector

```
delegate;
```

Acquire the Status Information of the Current Connector

```
isBLEEEable;
currentState;
hasConnectedSensor;
```

Attribute

isBLEEEable

A Boolean Value, used to identify whether the Bluetooth 4.0 of the current system is available.

```
@property(nonatomic,readonly)BOOL isBLEEEable;
```

Detailed Instructions – If this Boolean Value shows “Yes”, the Bluetooth 4.0 of the system is available and can connect to the Bluetooth 4.0 health equipments. If this Boolean Value shows “NO”, the Bluetooth 4.0 of the system is unavailable. The normal communication fails. You may use this attribute to judge whether the Bluetooth 4.0 of the current system is available or not. If not, the current system cannot communicate with Lifesense health equipments via Bluetooth. Then the application may reply to this problem, for example, inform the user that Bluetooth 4.0 is not available.

Available Version: 0.1

currentState

An Enumerated Value, used to indicate the scanning mode of the current connector

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

[@property\(nonatomic,readonly\)CONNECTOR_STATE currentState;](#)

Detailed Instructions – This value indicates the scanning mode of the current connector. When under different scanning modes, the connector chooses to handle the equipments in different situations. The default value is CONNECTOR_ANY_STATE, under which mode the connector will handle the equipments both under normal transmitting status and under pair-up status. Please refer to the instructions of enumerated CONNECTOR_STATE for detailed descriptions of scanning mode. We may know kinds of equipment status that the connector will search via this attribute.

Available Version: 0.1

hasConnectedSensor

A Boolean Value, used to tell whether the current connector is being connected to any equipment.

[@property\(nonatomic,readonly\)BOOL hasConnectedSensor;](#)

Detailed Instruction – This value is used to confirm whether the current connector is being connected to any equipment. If this Boolean Value shows “YES”, there is one equipment connecting to the connector. If this Boolean Value shows “NO”, there are no equipments connecting to the connector. You may know whether the current connector is being connected to an equipment by this attribute.

Available Version: 0.1

delegate

A proxy object, which must conform to LFHardwareConnectorDelegate protocol

[@property\(nonatomic,assign\)id<LFHardwareConnectorDelegate>delegate](#)

Detailed Instructions – This is a proxy object of HardwareConnector. By conforming to and realizing LFHardwareConnectorDelegate protocol, it can respond to the request of the connector and receive the feedback of equipment information and measurement data information from the connector. When applying the examples of HardwareConnector, the proxy class must be set for it.

Available Version: 0.1

Class Method

shareConnector

Return a common singleton object

[+\(LFHardwareConnector*\)shareConnector;](#)

Returned Value – Return an initialized LFHardwareConnector object. Use this class method to acquire a common singleton object and then utilize this object to be responsible for Bluetooth communication with the equipments. When you are using the common singleton object, you may visit this object whenever necessary without management of memory allocation and release of this object.

Instance Method

setConnectorState:

Set the status of the connector

[-\(void\)setConnectorState:\(CONNECTOR_STATE\)state;](#)

Parameter – state: the future status that the connector is going to be under. For the definition of all kinds of status, please refer to the definition of CONNECTOR_STATE.

Detailed Instructions – The default status of the connector is CONNECTOR_ANY_STATE, under which the connector will search the equipments both under the ordinary data transmission status and under the pair-up status. You may use this method to set the scanning mode of the connector as CONNECTOR_NORMAL_STATE, so that the connector only searches the equipments under ordinary data transmission status and ignores those under pair-up status. Vice versa, you may set the scanning mode of the connector as CONNECTOR_PAIRING_STATE, so that the connector only searches the equipments under pair-up status and ignores those under ordinary data transmission status.

Available Version: 0.1

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

enableSensor Type:

Set the type of health equipments which are required to be currently handled by the connector

-(void)enableSensorType:(LF_SENSOR_TYPE)type;

Parameter – type: Type of the health equipment. For the definitions of health equipment type, please refer to the definition of enumerated LF_SENSOR_TYPE.

Detailed Instruction – The default searched health equipment type of the initialized connector is null, that is to say, all kinds of health equipments will be ignored. Then when you apply the connector, you need to use this method to inform the connector the types of health equipments searched. You may use this method for several times to inform the connector of multiples types of health equipments searched.

Available Version: 0.1

disableSensorType:

Delete the type of health equipment which are currently handled by the connector

-(void)disableSensorType:(LF_SENSOR_TYPE)type

Parameter – type: Health equipment type. For the equipment types that are currently supported by the connector, please refer to the definition of enumerated LF_SENSOR_TYPE.

Detailed Instruction – The default searched health equipment type of the initialized connector is null, that is to say, all kinds of health equipments will be ignored. You may use this method to order the connector not to search a certain type of health equipment if necessary. You may also use this method for several times to cancel the search support for multiple types of health equipments.

Available Version: 0.1

enableAllSupportedSensorType:

Set the connector to handle all the supported health equipments.

-(void)enableAllSupportedSensorType;

Detailed Instructions – The default searched health equipment type of the initialized connector is null, that is to say, all kinds of health equipments will be ignored. You may use this method to order the connector to search all currently-supported health equipment type.

Available Version: 0.1

supportedSensorType

acquire all the health equipment types that can be currently supported by the connector

-(NSSet*)supportedSensorType

Returned Value – Return a set of all the currently supported equipment type. The elements included in the set are NSNumber objects transformed by LF_SENSOR_TYPE.

Detailed Instructions – You may apply this method to acquire all the health equipment type supported by the current interface bank. This interface bank can only be used to proceed to Bluetooth 4.0 communication with all supported health equipments. Therefore, this method can help to timely acquire the information of which type of equipment that the interface bank can communicate with.

Available Version: 0.1

startScanning

Turn ON the scanning mode of the connector

-(void)startScanning;

Detailed Instructions – Apply this method to turn on the scanning mode of the connector. However, before applying this method, it is advised to check whether Bluetooth 4.0 is currently available through the attribute isBLEEnable. If isBLEEnable shows NO, the equipments cannot be scanned and searched via Bluetooth 4.0 even applying this method. When the connector is under the scanning mode, it will keep searching all the compatible equipments around. If found, it will transmit the equipment information to the proxy object through the method in the LfHardwareConnectorDelegate protocol.

Available Version: 0.1

cancelScanning

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

Turn off the scanning mode of the connector

-(void)cancelScanning;

Detailed Instruction – Apply this method to turn off the scanning mode of the connector. When the scanning mode of the connector is turned off, the connector will stop searching the equipments until the function is ON again.

Available Version: 0.1

pairWithHardwareSensor:

Request for pair-up with a found health equipment

-(BOOL)pairWithHardwareSensor:(LFHardwareSensor*)sensor;

Parameter – sensor: The health equipment object. This equipment object must be the one that is searched and returned to the proxy object through the method hardwareConnectorDiscoveredSensor in LFHardwareConnectorDelegate protocol. Please refer to LFHardwareSensor class for detailed definitions.

Returned Value – If it returns “YES”, the request is accepted. If it returns “NO”, the request is rejected, maybe due to failure to meet the requirements.

Detailed Instruction – Apply this method to request pair-up with a health equipment. When the request is accepted, the proxy object will be informed of the pair-up result through the method hardwareConnectorPairedSensor:withState in LFHardwareConnectorDelegate protocol. NOTE: the health equipment requesting pair-up should be under the waiting status; otherwise the request will be rejected.

Available Version: 0.1

enqueueHardwareSensor:

Add the searched health equipment object into the waiting queue

-(BOOL)enqueueHardwareSensor:(LFHardwareSensor*)sensor

Parameter – sensor: The health equipment object. The health equipment object. This equipment object must be the one that is searched and returned to the proxy object through the method hardwareConnectorDiscoveredSensor in LFHardwareConnectorDelegate protocol. Please refer to LFHardwareSensor class for detailed definitions.

Returned Value – If it returns “YES”, the equipment object is successfully added to the queue. If it returns “NO”, the equipment object fails to be added to the queue.

Detailed Instruction – Use this method to add the searched health equipment object into the waiting queue. Note: the equipment object added to the waiting queue must be under normal data communication status and contain comprehensive equipment information as well. You’re required to retrieve the corresponding equipment information from the local drive and fill in the health equipment object. Otherwise, the action will fail. The equipment had better finish pair-up. Otherwise, the connection may fail. Since the connector connects to and handles one equipment each time, the equipment object in the queue will be handled in sequence.

Available Version: 0.1

LFHardwareConnectorDelegate Protocol

The proxy object of LFHardwareConnector class object must comply with LFHardwareConnectorDelegate protocol. The method stated in the protocol is used for data communication with the proxy of LFHardwareConnectorDelegate object, and delivery of measurement data and equipment information to the proxy. The method of this protocol must be realized.

Task:

Manage the searched equipment

-hardwareConnectorDiscoveredSensor:

-hardwareConnectorPairedSensor:withState:

Manage the received measurement data

-hardwareConnectorReceiveWeightMeasurementData:

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

-hardwareConnectorReceivePedometerMeasurementData:
 -hardwareConnectorReceiveBloodPressureMeasurementData:
 -hardwareConnectorReceiveKitchenScaleMeasurementData:
 -hardwareConnectorReceiveHeightMeasurementData:
 -hardwareConnectorReceiveGeneralWeightMeasurementData:

Manage the data download to the equipment

-setOfDownloadInfoTypesForSensor:
 -hardwareConnectorGetDownloadInfoForSensor:ofInfoType:
 -hardwareConnectorDidDownloadInfoForSensor:ofInfoType:withResult:

Instance Method

hardwareConnectorDiscoverSensor:

Inform the proxy object of the found health equipment

-(void)hardwareConnectorDiscoveredSensor:(LFHardwareSensor*)sensor

Parameter – sensor: Find the equipment of object by search. For detailed contents, please refer to the definition of class LFHardwareSensor.

Detailed Instruction – This method will submit the newly-found health equipment object to the proxy object. There are two kinds of status for the searched equipments: waiting for pair-up and ordinary data transmission. The newly-found equipment object will include the category information of the equipment and its status (Refer to LFHardwareSensor class). The other related information of the equipment are mostly null. The equipment objects submitted to the proxy by this method are updated real time. You may choose to request pair-up with the equipment or add the equipment into the waiting queue according to the status of the equipment. If you would like not to handle this equipment, you may abandon this equipment object. Do not delay handling it since the delay always face connection failure. For those equipment objects under ordinary data transmission status, you may utilize the attribute **pairSignature** of LFHardwareSensor class to inquire whether the relative equipment information has been stored. If yes, this equipment has been paired; otherwise, this equipment isn't paired and could not proceed to ordinary data communication. This method must be realized.

Available Version: 0.1

hardwareConnectorPairedSensor;withState:

Inform the proxy object of the result of pair-up with the health equipment

-(void)hardwareConnectorPairedSensor:(LFHardwareSensor*)sensor
withState:(BOOL)state

Parameter –

sensor: The health equipment object to proceed to pair-up

state: The pair-up result. If “state” shows “YES”, it indicates successful pair-up with the health equipments. If “state” shows “NO”, it indicates failure to pair-up with the health equipment.

Detailed Instruction – This method informs the proxy object of pair-up result of one health equipment. If the pair-up succeeds, the health equipment object includes all the relative information of health equipment. You're required to store the information of successfully-paired equipment objects in this method. Because when communicating with the equipment under ordinary data transmission status, all kinds of information of the equipment object may be useful. If it is not stored, the communication with this equipment, for example, acquire measurement data, etc. may not continue later. This method must be realized.

Available Version: 0.1

hardwareConnectorReceiveWeightMeasurementData:

Inform the proxy object of the weight measurement data received.

-(void)hardwareConnectorReceiveWeightMeasurementData:(WeightData*)data;

Parameter – data: weight measurement data object. Please refer to WeightData definitions for

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

details.

Detailed Instruction – This method is used to inform the proxy object of weight measurement data object received from the equipment. When you select to acquire measurement data from the weighing devices, this method must be realized.

Available Version: 0.1

hardwareConnectorReceivePedometerMeasurementData:

Inform the proxy object of the received pedometer measurement data

hardwareConnectorReceivePedometerMeasurementData:

-(void)hardwareConnectorReceivePedometerMeasurementData:(PedometerData*)data;

Parameter – data: pedometer measurement data. Please refer to the definition of PedometerData for details.

Detailed Instruction – This method informs the proxy object of the pedometer measurement data object received from the equipment. When you select to acquire measurement data from the pedometer-like equipment, this method must be realized.

Available Version: 0.1

hardwareConnectorReceiveBloodPressureMeasurementData:

Inform the proxy object of the received kitchen scale measurement data

-(void)hardwareConnectorReceiveBloodPressureMeasurementData:(BloodPressureData*)data

Parameter - data: blood pressure monitor measurement data. Please refer to the definition of BloodPressureData for details.

Detailed Instruction - This method informs the proxy object of the blood pressure monitor measurement data object received from the equipment. When you select to acquire measurement data from the blood pressure monitor - like equipment, this method must be realized.

Available Version: 0.1

hardwareConnectorReceiveKitchenScaleMeasurementData:

Inform the proxy object of the received kitchen scale measurement data

-(void)hardwareConnectorReceiveKitchenScaleMeasurementData:(KitchenScaleData*)data

Parameter - data: kitchen scale measurement data. Please refer to the definition of KitchenScaleData for details.

Detailed Instruction - This method informs the proxy object of the kitchen scale measurement data object received from the equipment. When you select to acquire measurement data from the kitchen scale - like equipment, this method must be realized.

Available Version: 0.4

hardwareConnectorReceiveHeightMeasurementData:

Inform the proxy object of the received height measurement data

-(void)hardwareConnectorReceiveHeightMeasurementData:(HeightData*)data

Parameter - data: height measurement data. Please refer to the definition of HeightData for details.

Detailed Instruction - This method informs the proxy object of the height measurement data object received from the equipment. When you select to acquire measurement data from the height measurement - like equipment, this method must be realized.

Available Version: 0.4

hardwareConnectorReceiveGeneralWeightMeasurementData:

Inform the proxy object of the received general BLE version body fat analyzer measurement data

-(void)hardwareConnectorReceiveGeneralWeightMeasurementData:(WeightData*)data

Parameter - data: weight measurement data object. Please refer to WeightData definitions for details.

Detailed Instruction – This method is used to inform the proxy object of weight measurement data object received from the equipment. When you select to acquire measurement data from

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

the equipments of the general BLE version body fat analyzer, this method must be realized.
Available Version: 0.5

-setOfDownloadInfoTypesForSensor:

Request to the proxy object for types of information downloaded to the equipment

-(NSSet *)setOfDownloadInfoTypesForSensor:(LFHardwareSensor*)sensor

Parameter – sensor: the equipment object currently connected; the target equipment requesting for information download. Please refer to the definition of LFHardwareSensor class for details.

Returned Value - Return a set of all the downloaded information type. The objects of this set should be the NSNumber object of corresponding enumerated value of enumerated LF_DOWNLOAD_INFO_TYPE. Please refer to the definition of enumerated LF_DOWNLOAD_INFO_TYPE for the definitions of enumerated values. This returned value can be null. If the returned value is null, you have no information required to be downloaded to the equipment.

Detailed Instruction – This method is used to make an request to the proxy object for the types of information necessary to be downloaded to the equipment. When you have a new kind of information that is required to be downloaded to the equipment, you may add the enumerated value NSNumber object of the corresponding enumerated LF_DOWNLOAD_INFO_TYPE into the set. The returned value can tell the connector which kind of information is required to be downloaded to the equipment. Pay attention to that the type of download information placed into the set must belong to the equipment's supported type of download information. Otherwise, it is invalid. This method is optional to be realized. However, if you do not realize this method, the information cannot be downloaded to the equipment. Also mind that this method should be returned in time. Do not block the return of this method. Otherwise, the equipment may fail to download the information.

-hardwareConnectorGetDownloadInfoForSensor:ofInfoType:

Make an request to the proxy object for the specific type of download information object downloaded to the equipment

-(id)hardwareConnectorGetDowanloadInfoForSensor:(LFHardwareSensor*)sensorofInfoType:(LF_DOWNLOAD_INFO_TYPE)infoType;

Parameter –

sensor: the equipment object currently connected; the target equipment required to download information. Please refer to the definition of LFHardwareSensor class for details.

infoType: Type of download information. Used to label the download type of the download information object. Please refer to the definition of enumerated LF_DOWNLOAD_INFO_TYPE for detailed downloaded information type.

Returned Value – Return the download information object. The download information object contains the message you would like to download to the equipment.

Detailed Instruction – This method is used to make a request to the proxy object for download information object of the specific type of download information. When you have new information content planned to be downloaded to the equipment for the designated download information type of parameter infoType, you can create the corresponding download information object and assign the download content to the corresponding attribute of the object and then return the object. When for the designated download information type by the parameter infoType, no information is required to be downloaded, please return null. The returned download information object by this method must be paired with infoType's designated download information type. Otherwise, the download fails. The realization of this method is optional. However, if you do not realize this method, the information cannot be downloaded to the equipment. Mind to return this method in time. Do not block the return of this method. Otherwise, it may result in failure to download information to the equipment.

-hardwareConnectorDidDownloadInfoForSensor: ofInfoType:withResult:

Inform the proxy object

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

-(void)hardwareConnectorDidDownloadInfoForSensor:(LFHardwareSensor*)sensorofInfoType:(LF_DOWNLOAD_INFO_TYPE)infoTypewithResult:(BOOL)result;

Parameter –

sensor: the equipment object currently connected; the target equipment required to download information. Please refer to the definition of LFHardwareSensor class for details.

infoType: Type of download information. Please refer to the definition of enumerated LF_DOWNLOAD_INFO_TYPE for the type of download information.

result: The result of information download to the equipment. When the “results” shows “YES”, the information is successfully downloaded to the equipment; when the “results” shows “NO”, the information fails to be downloaded to the equipment.

Detailed Instruction – This method is used to inform the proxy object of the result of downloading the designated type of download information by infoType When “result” shows “YES”, download succeeds; when “result” shows “NO”, download fails. Through this method, you can timely know whether a certain type of download information is downloaded to your equipment. If the download fails, you can choose to download the information next time the equipment is connected. The realization of this method is optional. If you choose not to realize this method, the result of download information will be invisible.

4.3 Instructions of Data Object and Structure

Class Definition of LFHardwareSensor:

Attribute	Class	Version in Service	Description
deviceId	NSString	0.1	Device ID, the marking to identify a device
deviceSn	NSString	0.1	Serial Number of a device. Transformed from Device ID, same effect as Device ID.
preparePair	BOOL	0.1	If it shows “YES”, it indicates that the device is under waiting status; if it shows “NO”, it indicates that the device is under data transmission status. Different operation modes to devices may lead to the devices under different kinds of status. The devices under waiting status can only proceed to pair-up

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			procedure while the devices under data transmission status can only proceed to data transmission procedure (e.g. acquiring measurement data or download user information). The device must proceed to pair-up first and then data transmission after successful pair-up.
sensorType	LF_SENSOR_TYPE	0.1	Refer to device type. Please refer to enumerated LF_SENSOR_TYPE for detailed definitions.
modelName	NSString	0.1	Refer to model number of the device. Those are the information retrieved from the device.
sensorName	NSString	0.1	Name of the device. No more than 12 bytes. The adopted coding method is UTF8. When the device is under pair-up announcement status, if the device has a device name, this piece of information can be retrieved from the device; if the device has no names, the value of this attribute will be empty. When the device is under data transmission status, the device name cannot be retrieved.
pairSignature	NSUInteger	0.1	Pair-up signature is the only symbol that labels a device. The signature written on the device via interface bank during pair-up is used to confirm the labeled device belongs to my pairing device. This value is created automatically by the interface bank during pair-up. The value is different every time pair-up proceeds. When the device is under data transmission status, this value can be retrieved from the device and written in the device object. You may use this value to inquire whether the information of this device is stored locally so as to confirm whether this device is paired. If yes, the device information

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			can be retrieved locally and written in the device object.
password	NSUInteger	0.1	The password uploaded by the device during pair-up is created by the device and used by the interface bank during communication with the device. It is used during data transmission to go through security verification so as to proceed to data communication with the device. If the security verification fails, the connection with the device will be terminated immediately.
supportDownloadInfoFeature	NSUInteger	0.2	The download characteristics supported by the device shows the supported types of download information of the device. The information can be acquired during the pair-up process and its value is the calculated value of enumerated LF_DOWNLOAD_INFO_TYPE. If the device supports multiple types of information, this attribute value is the calculated result of multiple enumerated LF_DOWNLOAD_INFO_TYPE. For example, when the device supports download of the device's setting information and the user's status information, the attribute value supportDownloadInfoFeature = LF_DOWNLOAD_TYPE_SENSORINFO LF_DOWNLOAD_TYPE_CURRENT_STATE. You may calculate this value and the enumerated value LF_DOWNLOAD_INFO_TYPE to judge whether the equipment supports download of a certain kind of information. For the types of download information not supported by the device, you cannot download that type of information to the equipment successfully.

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			Please refer to the definition instructions of enumerated LF_DOWNLOAD_INFO_TYPE for more information of enumerated value's represented download information.
maxUserQuantity	NSInteger	0.1	The maximum users supported by the device; the information retrieved from the device.
softwareVersion	NSString	0.1	The software version of the device, which is retrieved from the device. It's of no effect so far. You may choose to display the information of this device when required.
hardwareVersion	NSString	0.1	The hardware version of the device, which is retrieved from the device. It's of no effect so far. You may choose to display the information of this device when required.
manufactureName	NSString	0.1	Manufacturer of the device which is retrieved from the device. It's of no effect so far. You may choose to display the information of this device when required.
firmwareVersion	NSString	0.1	The firmware version of the device, which is retrieved from the device. It's of no effect so far. You may choose to display the information of this device when required.
systemId	NSString	0.1	The system ID of the device which is retrieved from the device. It's of no effect so far. You may choose to display the information of this device when required.

Definition of SensorDownloadInfo Class (setting information class of the device):

This class is used to seal the setting information downloaded to the device. When this type of information is required to download to this equipment, it's necessary to use object of this class to seal the information. Assign the value of the corresponding attributes according to the type of download information and the devices for information download.

Attribute	Class	Version in Service	Description
sensorName	NSString	0.1	The name of the device should be no more than 12

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			bytes. The adopted coding method is UTF8.
--	--	--	---

Definition of ExerciseTargetDownloadInfo Class (exercise target information of peometer):

This class is used to seal the exercise target information downloaded to the pedometer, that is to say, set the exercise target. When this type of information is required to download to this equipment, it's necessary to use object of this class to seal the information. Assign the value of the corresponding attributes according to the type of download information and the devices for information download.

Attribute	Class	Version in Service	Description
userNo	NSInteger	0.1	The user ID in the device, starts from 1 and ends at 254, 255 refers to the invalid user ID. When the device supports multiple users, you may switch among different users through the devices. A specific user ID can be utilized to assign the specific setting information to a certain user. If the value of userNo, is invalid, these information cannot be assigned to the correct user.
targetType	EXERCISE_TARGET_TYPE	0.1	Type of exercise target. The different values indicate the exercise target of different time slots. Please refer to the definition of enumerated EXERCISE_TARGET_VALUE.
targetStep	NSInteger	0.1	Indicates the step goals of the exercise.
targetCalories	double	0.1	Indicates the targeted consumed energy value. Unit: kilogram calories.
targetDistance	double	0.1	Indicates the targeted distance of the exercise. Unit: m.
targetExercieseAmount	double	0.1	Indicates the targeted exercise volume of the exercise.

Definition of CurrentStateDownloadInfo Class (Current Status Information class of the user)

This class is used to seal the current status information of a specific user downloaded to the device. When this type of information is required to download to this equipment, it's necessary to use object of this class

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

to seal the information. Assign the value of the corresponding attributes according to the type of download information and the devices for information download.

Definition of TargetStateDownloadInfo class (targeted status information class of the user)

Attribute	Class	Version in Service	Description
userNo	NSInteger	0.1	The user ID in the device, starts from 1 and ends at 254, 255 refers to the invalid user ID. When the device supports multiple users, you may switch among different users through the devices. A specific user ID can be utilized to assign the specific setting information to a certain user. If the value of userNo, is invalid, these information cannot be assigned to the correct user.
weight	double	0.1	Current weight of the user. Unit: kg
fatRatio	double	0.1	Current fat rate of the user.
height	double	0.1	Current height of the user. Unit: m
waistline	double	0.1	Current waistline of the user. Unit: m
stride	double	0.1	Current stride of the user. Unit: m
age	NSInteger	0.1	Current age of the user.

This class is used to seal the targeted status information of a specific user downloaded to the device. When this type of information is required to download to this equipment, it's necessary to use object of this class to seal the information. Assign the value of the corresponding attributes according to the type of download information and the devices for information download.

Attribute	Class	Version in Service	Description
userNo	NSInteger	0.1	The user ID in the device, starts from 1 and ends at 254, 255 refers to the invalid user ID. When the device supports multiple users, you may switch among different users through the

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			devices. A specific user ID can be utilized to assign the specific setting information to a certain user. If the value of userNo, is invalid, these information cannot be assigned to the correct user.
targetWeight	double	0.1	The targeted weight set by the user. Unit: kg
targetHeight	double	0.1	The targeted height set by the user. Unit: m.
targetDate	NSString	0.1	The targeted date set by the user (Time Format: yyyy-MM-dd HH:mm:ss), e.g. 2012-08-01 16:55:22

Definition of UserMessageDownloadInfo class (personal detailed information class of the user)

This class is used to seal the personal detailed information of a specific user downloaded to the device. When this type of information is required to download to this equipment, it's necessary to use object of this class to seal the information. Assign the value of the corresponding attributes according to the type of download information and the devices for information download.

Attribute	Class	Version in Service	Description
userNo	NSInteger	0.1	The user ID in the device, starts from 1 and ends at 254, 255 refers to the invalid user ID. When the device supports multiple users, you may switch among different users through the devices. A specific user ID can be utilized to assign the specific setting information to a certain user. If the value of userNo, is invalid, these information cannot be assigned to the correct user.
userId	NSInteger	0.1	User ID, the attribute that can be set by you via application terminal, is used to

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			label the personal information and assign to a specific user.
username	NSString	0.1	User Name, coded by UTF8, should be no more than 16 bytes.
sex	NSInteger	0.1	Indicates the gender of the user.
activityLevel	NSInteger	0.1	Indicates the exercise level of the user.
birthdayYear	NSInteger	0.1	Indicates the year of birth of the user.
birthdayMonth	NSInteger	0.1	Indicates the month of birth of the user.
birthdayDay	NSInteger	0.1	Indicates the day of birth of the user.
weekStart	NSInteger	0.1	Indicates the start of a week set by the user. The numeral “1” refers to start from Sunday; the numeral “2” refers to start from Monday. Other value are not applicable yet.

Definition of WeightData class (weight measurement data)

Attribute	Class	Version in Service	Description
date	NSString	0.1	The created date of the measurement data. It will be created simultaneously when the measurement data is produced by the device. (Time Format: yyyy-MM-dd HH:mm:ss), e.g. 2012-08-01 16:55:22
deviceId	NSString	0.1	The device ID of the device that produces the measurement data. You may tell the specific device which creates this piece of measurement data from this ID.
userNo	NSInteger	0.1	The selected user number during measurement, starts from 1 and ends at 254 while 255 refers to invalid user number. When the device supports multiple users, you may switch among different users through the devices. A specific user number

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			can be utilized to assign the specific setting information to a certain user. If the value of userNo, is invalid, these information cannot be assigned to the correct user.
weight	double	0.1	The measured weight value.
pbf	double	0.1	The measured body fat rate. It' s applicable when the device supports direct measurement of body fat rate.
resistance_1	double	0.1	The measured bio-impedance value (5 kHz, the output impedance value of the signal) If the device supports to measure body fat rate, it may output this bio-impedance value. This bio-impedance value can calculate the body fat rate according to the specific arithmetic.
Resistance_2	double	0.1	The measured bio-impedance value (output bio-impedance value of 50 kHz signal). If the device supports measured body fat rate, this bio-impedance value may be outputted, which can calculate the body fat rate according to the specific arithmetic. (For the arithmetic of body fat rate calculation, some adopts 5 kHz bio-impedance value while some other adopts 50 kHz bio-impedance value.)
unit	NSString	0.1	Measurement unit of weight measurement (St, LB, or Kg)

Definition of PedometerData class (measurement data of pedometer)

Attribute	Class	Version in Service	Description
date	NSString	0.1	The created date of measurement data (Time Format: yyyy-MM-dd HH:mm:ss), e.g. 2012-08-01 16:55:22
deviceId	NSString	0.1	Device ID of the device that outputs measurement

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			data. You may tell the which device produces the measurement data according this ID.
userNo	NSInteger	0.1	The user number during measurement, starts from 1 and ends at 254 while 255 refers to invalid user number. When the device supports multiple users, you may switch among different users through the devices. A specific user number can be utilized to assign the specific setting information to a certain user. If the value of userNo, is invalid, these information cannot be assigned to the correct user.
walkSteps	NSInteger	0.1	Walking steps.
runSteps	NSInteger	0.1	Running steps.
examount	double	0.1	
calories	double	0.1	Consumed energy. Unit: kilogram calories
exerciseTime	NSInteger	0.1	Exercise time. Unit: minute
distance	NSInteger	0.1	Exercise distance. Unit: m
battery	NSInteger	0.1	The battery power class of the equipment

Definition of BloodPressureData class (measurement data of blood pressure monitor)

Attribute	Class	Version in Service	Description
date	NSString	0.1	The created date of measurement data(Time Format:yyyy-MM-dd HH:mm:ss) e. g. : 2012-08-01 16:55:22
deviceId	NSString	0.1	The device ID of the device that produces the measurement data. You may tell the specific device which creates this piece of measurement data from this ID.
userNo	NSInteger	0.1	The user number during measurement, starts from 1

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			and ends at 254 while 255 refers to invalid user number. When the device supports multiple users, you may switch among different users through the devices. A specific user number can be utilized to assign the specific setting information to a certain user. If the value of userNo, is invalid, these information cannot be assigned to the correct user.
systolic	double	0.2	The measured systolic pressure(This value is calculated with the unit mmHg)
diastolic	double	0.2	The measured diastolic pressure(This value is calculated with the unit mmHg)
pluseRate	double	0.1	The measured heart rate
deviceSelectedUnit	NSString	0.2	The selected setting unit on the equipment while measuring the blood pressure (mmHg or kPa)
battery	NSInteger	0.1	The battery power class of the equipment

Definition of HeightData class (measurement data of height guage)

Attribute	Class	Version in Service	Description
date	NSString	0.4	The created date of measurement data(Time Format:yyyy-MM-dd HH:mm:ss) e.g.2012-08-01 16:55:22
deviceId	NSString	0.4	The device ID of the device that produces the measurement data. You may tell the specific device which creates this piece of measurement data from this ID.
userNo	NSInteger	0.4	The user number during measurement, starts from 1 and ends at 254 while 255 refers to invalid user number. When the device supports multiple users,

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

			you may switch among different users through the devices. A specific user number can be utilized to assign the specific setting information to a certain user. If the value of userNo, is invalid, these information cannot be assigned to the correct user.
height	double	0.4	The measured height value.
unit	NSString	0.4	Measurement unit of height measurement
battery	NSInteger	0.4	The battery power class of the equipment

Definition of KitchenScaleData Class (measurement data of kitchen scale)

Attribute	Class	Version in Service	Description
weight	double	0.4	While the unit is kg, it indicates the weight data of the LB part.
sectionWeight	double	0.4	While the unit is LB, it indicates the weight data of the OZ part.
time	NSInteger	0.4	The set regular time intervals of the equipment
unit	NSString	0.4	The current selected unit of the equipment
Battery	NSInteger	0.4	The battery power class of the equipment

4.3 Definition of enumerated structure

Definition of enumerated CONNECTOT_STATE

enum

```
{
    CONNECTOR_NORMAL_STATE=0X01,
    CONNECTOR_PAIRING_STATE=0X02,
    CONNECTOR_ANY_STATE
CONNECTOR_NORMAL_STATE|CONNECTOR_PAIRING_STATE
}CONNECTOR_STATE
```

Explanation of enumerated value:

CONNECTOR_NORMAL_STATE

Scanning and searching status of the connector, only scans and searches the devices under data transmission status.

CONNECTOR_PAIRING_STATE

Pair-up scanning and searching status of the connector, only scans and searches the devices under

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

waiting status.

[CONNECTOR_ANY_STATE](#)

Random scanning status of the connector, searches the devices both under data transmission status and waiting status.

Definition of enumerated LF_SENSOR_TYPE

```
enum
{
    LF_SENSOR_TYPE_UNKONW = 0x00,
    LF_SENSOR_TYPE_WEIGHT_SCALE = 0x01,
    LF_SENSOR_TYPE_PEDOMETER = 0x04
    LF_SENSOR_TYPE_BLOODPRESSURE = 0x05,
    LF_SENSOR_TYPE_KITCHEN_SCALE = 0x06,
    LF_SENSOR_TYPE_HEIGHT = 0x07
    LF_SENSOR_TYPE_GENERAL_WEIGHT_SCALE=0x08
```

[}LF_SENSOR_TYPE](#)

Explanations of enumerated value:

[LF_SENSOR_TYPE_UNKONW](#)

Refers to the unknown type of devices.

[LF_SENSOR_TYPE_WEIGHT_SCALE](#)

Refers to the weighing scale.

[LF_SENSOR_TYPE_PEDOMETER](#)

Refers to the pedometer.

[LF_SENSOR_TYPE_BLOODPRESSURE](#)

Refers to the blood pressure monitor.

[LF_SENSOR_TYPE_KITCHEN_SCALE](#)

Refers to the kitchen scale.

[LF_SENSOR_TYPE_HEIGHT](#)

Refers to the height gague.

[LF_SENSOR_TYPE_GENERAL_WEIGHT_SCALE=0x08](#)

Refers to the general BLE version body fat analyzer.

Definition of enumerated LF_DOWNLOAD_INFO_TYPE

```
enum
{
    LF_DOWNLOAD_TYPE_UNKONW = 0x00,
    LF_DOWNLOAD_TYPE_SENSORINFO = 0x01,
    LF_DOWNLOAD_TYPE_EXERCISE_DAY_TARGET= 0x02,
    LF_DOWNLOAD_TYPE_EXERCISE_WEEK_TARGET=0x04,
    LF_DOWNLOAD_TYPE_CURRENT_STATE = 0x08,
    LF_DOWNLOAD_TYPE_TARGET_STATE = 0x10,
    LF_DOWNLOAD_TYPE_USER_MESSAGE = 0x20
```

[}LF_DOWNLOAD_INFO_TYPE](#)

Explanations of enumerated values:

[LF_DOWNLOAD_TYPE_SENSORINFO](#)

Indicates that the downloaded data category is the setting information of the devices. The corresponding download information object is sealed by SensorDownloadInfo class.

[LF_DOWNLOAD_TYPE_EXERCISE_DAY_TARGET](#)

Indicates that the downloaded data category is the daily exercise goal information of the pedometer. The corresponding download information object is sealed by ExerciseTargetDownloadInfo class.

[LF_DOWNLOAD_TYPE_EXERCISE_WEEK_TARGET](#)

API Interface of Bluetooth Measuring Equipment		File No.:	
		Issued Edition: 0.4	
Product Name:		Product Model:	

Indicates that the downloaded data category is the weekly exercise goal information of the pedometer. The corresponding download information object is sealed by ExerciseTargetDownloadInfo class.
[LF_DOWNLOAD_TYPE_CURRENT_STATE](#)

Indicates that the downloaded data category is the current status information of the user. The corresponding download information object is sealed by CurrentStateDownloadInfo class.

[LF_DOWNLOAD_TYPE_TARGET_STATE](#)

Indicates that the downloaded data category is the targeted status information of the user. The corresponding download information object is sealed by TargetStateDownloadInfo class.

[LF_DOWNLOAD_TYPE_USER_MESSAGE](#)

Indicates that the downloaded data category is the detailed personal information of the user. The corresponding download information object is sealed by UserMessageDownloadInfo class.

Enumerated EXERCISE_TARGET_TYPE

```
enum
{
    EXERCISE_DAY_TARGET = 0x01,
    EXERCISE_WEEK_TARGET
}EXERCISE_TARGET_TYPE
```

Explanations of enumerated values:

[EXERCISE_DAY_TARGET](#)

Indicates that the type of exercise goal is daily goal.

[EXERCISE_WEEK_TARGET](#)

Indicates that the type of exercise goal is weekly goal.