

## Table of Contents

<b>I. Introduction.....</b>	<b>3</b>
<b>II. Class .....</b>	<b>3</b>
1. OmronPeripheralManager .....	3
1.1 sharedManager .....	3
1.2 libVersion .....	3
1.3 startManager .....	3
1.4 stopManager.....	3
1.5 retrieveManagerConfiguration.....	4
1.6 setConfiguration .....	4
1.7 getConfiguration .....	4
1.8 setAPIKey .....	4
1.9 getAPIKey .....	5
1.10 getBluetoothState.....	5
1.11 startScanPeripheralsWithCompletionBlock.....	5
1.12 stopScanPeripherals .....	5
1.13 stopScanPeripheralsWithCompletionBlock .....	6
1.14 connectPeripheral .....	6
1.15 connectPeripheral .....	7
1.16 resumeConnectPeripheralWithUser .....	8
1.17 resumeConnectPeripheralWithUsers .....	8
1.18 endConnectPeripheralWithCompletionBlock.....	9
1.19 disconnectPeripheral .....	9
1.20 onConnectStateChangeWithCompletionBlock .....	10
1.21 startDataTransferFromPeripheral.....	10
1.22 startDataTransferFromPeripheral.....	11
1.23 startDataTransferFromPeripheral.....	12
1.24 startDataTransferFromPeripheral.....	13
1.25 endDataTransferFromPeripheralWithCompletionBlock .....	14
1.26 updatePeripheral.....	14
1.27 updatePeripheral.....	15
1.28 updatePeripheral.....	16
1.29 startRecording .....	17
1.30 stopRecordingWithCompletionBlock.....	17
2. OmronPeripheralManagerConfig.....	18
2.1 List of public variables .....	18
2.2 retrievePeripheralConfigurationWithGroupID.....	19
3. OmronPeripheral .....	19
3.1 initWithLocalName .....	19
3.2 getVitalDataWithUser .....	20
3.3 getVitalDataWithUser .....	20
3.4 getVitalData .....	21
3.5 getVitalDataWithCompletionBlock .....	21
3.6 getDeviceInformation .....	21
3.7 getDeviceInformationWithCompletionBlock .....	22
3.8 getDeviceSettings.....	22
3.9 getDeviceSettingsWithUser .....	22
3.10 getDeviceSettingsWithUser .....	23
<b>III. OmronDefines.....</b>	<b>24</b>
1. OmronDefines.....	24
1.1 OMRONVitalDataMeasurementModeTypeKey.....	24
1.2 OMRONWheezeTypeKey.....	24
1.3 OMRONTemperatureUnitTypeKey.....	24
1.4 OMRONTemperatureLevelTypeKey.....	25

1.5	OMRONDevicePersonalSettingsUserGenderType .....	25
1.6	OMRONDevicePersonalSettingsBloodPressureTruReadStatus .....	25
1.7	OMRONDevicePersonalSettingsBloodPressureTruReadInterval .....	25
1.8	OMRONDeviceAlarmStatus .....	26
1.9	OMRONDeviceAlarmType .....	26
1.10	OMRONDeviceSleepAutomatic .....	26
1.11	OMRONDeviceTimeFormat .....	26
1.12	OMRONDeviceDateFormat .....	27
1.13	OMRONDeviceDistanceUnit .....	27
1.14	OMRONDeviceNotificationStatus .....	27
1.15	OMRONDeviceWeightDisplay .....	27
1.16	OMRONDeviceWeightUnit .....	27
1.17	OMRONDeviceScanSettingsMode .....	28
1.18	OMRONBLEConnectionState .....	28
1.19	OMRONBLEBluetoothState .....	28
1.20	OMRONBLEDeviceCategory .....	29
<b>IV.</b>	<b>ErrorCodes .....</b>	<b>30</b>
1.	ErrorCode .....	30
2.	NSError .....	30

## I. Introduction

This document is an API reference for the communication interface between Omron Connected Devices and OmronConnectivityLibrary for iOS.

## II. Class

### 1. OmronPeripheralManager

#### 1.1 sharedManager

*(id)sharedManager*

Obtain a shared instance of OmronConfigManager.

Returns	<i>OmronPeripheralManager *</i>
	Returns an instance of OmronConfigManager.

OmronConfigManager API must always be called via SharedManager.

code example:

```
[[OmronPeripheralManager sharedManager] libVersion]
```

#### 1.2 libVersion

*(NSString \*)libVersion*

Method to get the library version.

Returns	<i>NSString *</i>
	Obtain OmronConnectivityLibrary version.

#### 1.3 startManager

*(void)startManager*

Starts Omron Peripheral Manager.

By starting Omron Peripheral Manager, Bluetooth will be set up and connection will be enabled.

#### 1.4 stopManager

*(void)stopManager*

Stop Omron Peripheral Manager.

## 1.5 retrieveManagerConfiguration

*(id)retrieveManagerConfiguration*

Get the supported device information data list. This data is required to connect to Omron devices.

Returns	<i>NSDictionary *</i>
	List of devices supported by the OmronConnectivityLibrary.

## 1.6 setConfiguration

*(void)setConfiguration:(OmronPeripheralManagerConfig \*)config*

Set up Omron device settings such as filters, timeout intervals, etc

Parameters	<i>OmronPeripheralManagerConfig * config</i>
	Use the values obtained from <code>getConfiguration</code> to update the settings. Please refer to the details of Omron Peripheral Manager Config below. <a href="#">II.2 OmronPeripheralManagerConfig</a>

## 1.7 getConfiguration

*(OmronPeripheralManagerConfig \*)getConfiguration*

Get existing Omron device settings such as filters, timeout intervals, etc.

Returns	<i>OmronPeripheralManagerConfig *</i>
	Existing Peripheral Configuration. Please refer to the details of Omron Peripheral Manager Config below. <a href="#">II.2 OmronPeripheralManagerConfig</a>

## 1.8 setAPIKey

*(void)setAPIKey:(NSString \*)APIKey options:(NSDictionary \*)options*

Set the partner API key.

Setting the partner API key determines the list of supported Omron device information data. It must be done immediately after the application starts.

Parameters	<i>NSString * apiKey</i>
	Partner's API key.
	<i>NSDictionary * options</i>
	Always set to null as it is not currently used.

## 1.9 getAPIKey

*(NSString \*)getAPIKey*

Get the Partner's API key, set with setAPIKey.

Returns	<i>NSString *</i>
	Return the set Partner's API key.

## 1.10 getBluetoothState

*(OMRONBLEBluetoothState)getBluetoothState*

Returns the Bluetooth state of smartphone.

Returns	<i>OMRONBLEBluetoothState</i>
	Bluetooth state. See <a href="#">III.1.19 OMRONBLEBluetoothState</a>

## 1.11 startScanPeripheralsWithCompletionBlock

*(void)startScanPeripheralsWithCompletionBlock:(void (^)(NSArray \*retrievedPeripherals, NSError \*error))completionBlock*

Starts scanning for Omron devices.

Returns a list of devices found to the completionBlock in real time. If no devices are found, a timeout error object is returned.

Parameters	<i>(void (^)(NSArray *retrievedPeripherals, NSError *error))completionBlock</i>
	Set up a completionBlock to notify the results of the scan.

completionBlock

Parameters	<i>NSArray *retrievedPeripherals</i>
	Information on scan results is returned.
	<i>NSError *error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.12 stopScanPeripherals

*(void)stopScanPeripherals*

Stops scanning of Omron devices.

Use II.1.13 stopScanPeripheralsWithCompletionBlock if you want to stop receiving results.

## 1.13 stopScanPeripheralsWithCompletionBlock

*(void)stopScanPeripheralsWithCompletionBlock:(void (^)(NSError \*error))completionBlock*

Stops scanning of Omron devices.

Parameters	<i>(void (^)(NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of the scan stop.

completionBlock

Parameters	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.14 connectPeripheral

*(void)connectPeripheral:(OmronPeripheral \*)peripheral  
withCompletionBlock:(void (^)(OmronPeripheral \*peripheral, NSError  
\*error))completionBlock*

Starts the connection with the detected Omron device. This will result in a Bluetooth pairing request between the Omron device and the Smartphone.

Once the Bluetooth pairing request is granted by the smartphone user, the device is connected to the smartphone and OmronPeripheralManager can communicate with the device.

OmronPeripheralManager returns the connected Omron device information to the application in the form of an OmronPeripheral object.

If a failure occurs during the connection, an error object is returned to the application.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral to Connect to using Manager.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of connect to device.

completionBlock

Parameters	<i>OmronPeripheral *peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError *error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.15 connectPeripheral

```
(void)connectPeripheral:(OmronPeripheral *)peripheral  
        withWait:(BOOL)isWait withCompletionBlock:(void (^)(OmronPeripheral  
        *peripheral, NSError *error))completionBlock
```

Starts the connection with the detected Omron device. This initiates a Bluetooth pairing request between the Omron device and the smartphone.

If isWait is set to True, user selection can be performed after device connection.

Specify the user using resumeConnectPeripheral in onConnectCompleted.

This allows you to determine if the user you wish to select is already in use at the time of pairing.

Please refer to Implementation Guide OmronConnectivityLibraryIOS for more information on determining when a user is in use.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral to Connect to using Manager.
	<i>BOOL isWait</i>
	Setting YES will wait for user selection after the connection is established.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of connect to device.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	An Omron device object is returned in the completionBlock.. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.16 resumeConnectPeripheralWithUser

```
(void)resumeConnectPeripheralWithUser:(int)currentUser withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Select a user for the OMRON device connected with II.1.15 connectPeripheral.

Parameters	<i>int currentUser</i>
	Current selected User to begin pair.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of connect to device.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.17 resumeConnectPeripheralWithUsers

```
(void)resumeConnectPeripheralWithUsers:(NSArray *)users withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Select multiple users for OMRON devices connected with II.1.15 connectPeripheral.

Parameters	<i>NSArray * users</i>
	User list to connect.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of connect to device.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>



## 1.18 endConnectPeripheralWithCompletionBlock

```
(void)endConnectPeripheralWithCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock)
```

Terminate the connection of OMRON devices connected by II.1.15 connectPeripheral.

Parameters	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock)</i>
	Sets a completionBlock to notify the results of connect to device.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.19 disconnectPeripheral

```
(void)disconnectPeripheral:(OmronPeripheral *)peripheral withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Explicitly disconnects any currently connected Omron device independent of connectPeripheral.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral to Connect to using Manager.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of device disconnection.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.20 onConnectStateChangeWithCompletionBlock

*(void)onConnectStateChangeWithCompletionBlock:(void (^)(int state))completionBlock*

Sets a completionBlock to be notified of connection status changes.

Parameters	<i>(void (^)(int state))completionBlock</i>
	Sets a completionBlock that notifies the connection status to the device.

completionBlock

Parameters	<i>int state</i>
	The connection status is returned in the completionBlock. See <a href="#">III.1.19 OMRONBLEBluetoothStatell</a>

## 1.21 startDataTransferFromPeripheral

*(void)startDataTransferFromPeripheral:(OmronPeripheral \*)peripheral  
withUser:(int)currentUser withWait:(BOOL)isWait withCompletionBlock:(void  
(^)(OmronPeripheral \*peripheral, NSError \*error))completionBlock*

Starts acquiring measurement data for the paired Omron device.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral Object.
	<i>int currentUser</i>
	User number to connect.
	<i>BOOL isWait</i>
	Only YES is supported. Please set it to YES.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of data transfer.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.22 startDataTransferFromPeripheral

```
(void)startDataTransferFromPeripheral:(OmronPeripheral *)peripheral withUsers:(NSArray *)users withWait:(BOOL)isWait withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Starts acquiring measurement data for the paired Omron device.  
Data from multiple users can be acquired simultaneously.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral Object.
	<i>NSArray * users</i>
	User list to connect.
	<i>BOOL isWait</i>
	Only YES is supported. Please set it to YES.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of data transfer.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.23 startDataTransferFromPeripheral

```
(void)startDataTransferFromPeripheral:(OmronPeripheral *)peripheral  
withUser:(int)currentUser withWait:(BOOL)isWait  
withType:(OMRONVitalDataTransferCategory)type withCompletionBlock:(void  
(^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Starts acquiring measurement data from a paired Omron device by specifying the category of measurement data.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral Object.
	<i>int currentUser</i>
	User selection for transferring data.
	<i>BOOL isWait</i>
	Only YES is supported. Please set it to YES.
	<i>OMRONVitalDataTransferCategory type</i>
	Categories of data to transfer
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of data transfer.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.24 startDataTransferFromPeripheral

```
(void)startDataTransferFromPeripheral:(OmronPeripheral *)peripheral withUsers:(NSArray *)users withWait:(BOOL)isWait withType:(OMRONVitalDataTransferCategory)type withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Starts acquiring measurement data from a paired Omron device by specifying the category of measurement data.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral Object.
	<i>NSArray * users</i>
	User list to connect.
	<i>BOOL isWait</i>
	Only YES is supported. Please set it to YES.
	<i>OMRONVitalDataTransferCategory type</i>
	Categories of data to transfer
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of data transfer.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.25 endDataTransferFromPeripheralWithCompletionBlock

```
(void)endDataTransferFromPeripheralWithCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Terminates data transfer and clears the unsent flag of the OMRON device.

Since clearing the unsent flag prevents reacquisition of acquired data, please execute this API after saving the acquired data.

Parameters	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of data transfer.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.26 updatePeripheral

```
(void)updatePeripheral:(OmronPeripheral *)peripheral withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Updates Omron device settings such as time format.

API for single user devices only.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral Object.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of device settings update.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.27 updatePeripheral

```
(void)updatePeripheral:(OmronPeripheral *)peripheral withUser:(int)currentUser  
withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError  
*error))completionBlock
```

Updates the settings of Omron devices, such as time format.  
API for devices that can be used by multiple users.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral Object.
	<i>int currentUser</i>
	User selection for updating device.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of device settings update.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

# 1.28 updatePeripheral

```
(void)updatePeripheral:(OmronPeripheral *)peripheral withUsers:(NSArray *)users  
withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError  
*error))completionBlock
```

Update settings for Omron devices, including time format.  
Update settings for devices that can be used by multiple users at once.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral Object.
	<i>NSArray * users</i>
	User list to perform device update.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of device settings update.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>



## 1.29 startRecording

```
(void)startRecording:(OmronPeripheral *)peripheral onSignalStrength:(void (^)(double signal))signalStrengthBlock withCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Starts recording data from the specified sonic communication-compatible Omron device.

Parameters	<i>OmronPeripheral * peripheral</i>
	Omron Peripheral Object.
	<i>(void (^)(double signal))signalStrengthBlock</i>
	Sets a completionBlock that notifies the connection status to the device.
	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of recording.

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 1.30 stopRecordingWithCompletionBlock

```
(void)stopRecordingWithCompletionBlock:(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock
```

Stops recording data from the specified sonic communication-compatible Omron device.

Parameters	<i>(void (^)(OmronPeripheral *peripheral, NSError *error))completionBlock</i>
	Sets a completionBlock to notify the results of recording

completionBlock

Parameters	<i>OmronPeripheral * peripheral</i>
	The completionBlock returns an Omron device object. See <a href="#">II.3 OmronPeripheral</a>
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 2. OmronPeripheralManagerConfig

### 2.1 List of public variables

Edit the Omron device configuration obtained with `getConfiguration` and set it with `setConfiguration`.

See [II.1.6 setConfiguration](#) and [II.1.7 getConfiguration](#)

Variables	<i>NSMutableArray * deviceFilters</i>
	Device Filters (Optional) Allows the partner application to filter devices to specific Omron device models when detecting Omron connected
	<i>NSTimeInterval timeoutInterval</i>
	Determines the scan timeout interval when searching for Omron connected devices. The default timeout is 60 seconds
	<i>NSString * userHashId</i>
	This is used to authenticate the connection between the Omron device and the application during pair transfer. This should preferably be the login e-mail address of the user used for application authentication purposes. The application should ensure that this property is properly configured and identical when using the pair and transfer functions. It should also be consistent between application upgrades. If different inputs are provided, the device will fail to connect or transfer due to encryption issues.
	<i>BOOL enableAllDataRead</i>
	Provides the ability to read all measurements from the selected user's device. Once this flag is used, the application must update this flag according to subsequent data transfers.
	<i>NSMutableArray * deviceSettings</i>
	This is used to set configuration in device.  For more information, please refer to the following documents Implementation GuideOmronConnectivityLibrary iOS and Appendix-DataStructuresOverview iOS.

## 2.2 retrievePeripheralConfigurationWithGroupId

```
(NSMutableDictionary *)retrievePeripheralConfigurationWithGroupId:(NSString *)groupId  
andGroupIncludedId:(NSString *)groupIncludedId
```

This function can be used to retrieve a specific device configuration.

Parameters	<i>NSString * groupId</i>
	Device Category Type from OmronPeripheral.
	<i>NSString * groupIncludedId</i>
	Device Model Type from OmronPeripheral.
Returns	<i>NSMutableDictionary *</i>
	specific device configuration. For more information, please refer to the following documents Implementation GuideOmronConnectivityLibraryIOS and Appendix- DataStructuresOverviewIOS

## 3. OmronPeripheral

### 3.1 initWithLocalName

```
(id)initWithLocalName:(NSString *)localName andUUID:(NSString *)UUID
```

Generate an OMRON peripheral object using the BLE local name and BLE UUID.

Parameters	<i>NSString * localName</i>
	BLE local name of the Omron device to be connected
	<i>NSString * UUID</i>
	BLE UUID of the target Omron device to be connected
Returns	<i>OmronPeripheral *</i>
	Instance of OMRON peripheral object

## 3.2 `getVitalDataWithUser`

```
(void)getVitalDataWithUser:(int)user withCompletionBlock:(void (^)(NSMutableDictionary *vitalData, NSError *error))completionBlock
```

Return vital data.

This vital data is the user's data set in user parameter.

Parameters	<i>int user</i>
	Target users you want to acquire
	<i>(void (^)(NSMutableDictionary *vitalData, NSError *error))</i>
	Set a completionBlock to be notified of the recording results.

completionBlock

Parameters	<i>NSMutableDictionary * vitalData</i>
	Vital data For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix-DataStructuresOverviewIOS
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## 3.3 `getVitalDataWithUser`

```
(id)getVitalDataWithUser:(int)user
```

Return vital data.

This vital data is the user's data set in user parameter.

Parameters	<i>int user</i>
	Target users you want to acquire
Returns	<i>NSMutableDictionary *</i>
	Vital data For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix-DataStructuresOverviewIOS

### 3.4 `getVitalData`

*(id)getVitalData*

Returns user's vital data specified in `startDataTransferFromPeripheral`.

Returns	<i>NSMutableDictionary *</i>
	Vital data For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix-DataStructuresOverviewIOS

### 3.5 `getVitalDataWithCompletionBlock`

*(void)getVitalDataWithCompletionBlock:(void (^)(NSMutableDictionary \*vitalData, NSError \*error))completionBlock*

Asynchronously retrieves the vital data of the user specified in `startDataTransferFromPeripheral`.

Parameters	<i>(void (^)(NSMutableDictionary *vitalData, NSError *error)) completionBlock</i>
	CompletionBlock for vital data acquisition

*completionBlock*

Parameters	<i>NSMutableDictionary * vitalData,</i>
	Vital data For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix-DataStructuresOverviewIOS
	<i>NSError *error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

### 3.6 `getDeviceInformation`

*(NSMutableDictionary \*)getDeviceInformation*

Returns Omron device information.

Returns	<i>NSMutableDictionary *</i>
	Omron device information. For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix-DataStructuresOverviewIOS

### 3.7 `getDeviceInformationWithCompletionBlock`

```
(void)getDeviceInformationWithCompletionBlock:(void (^)(NSMutableDictionary *deviceInfo, NSError *error))completionBlock
```

Obtains OMRON device information asynchronously.

Parameters	<i>(void (^)(NSMutableDictionary *deviceInfo, NSError *error)) completionBlock</i>
	CompletionBlock for acquiring OMRON device information

completionBlock

Parameters	<i>NSMutableDictionary * deviceInfo</i>
	Omron device information data is returned to the completionBlock. For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix-DataStructuresOverviewIOS
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

### 3.8 `getDeviceSettings`

```
(id)getDeviceSettings
```

Return all the device settings.

Returns	<i>NSMutableDictionary *</i>
	Omron device settings. For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix-DataStructuresOverviewIOS

### 3.9 `getDeviceSettingsWithUser`

```
(id)getDeviceSettingsWithUser:(int)user
```

Return personal settings of a particular user.

Parameters	<i>int user</i>
	Target users you want to acquire
Returns	<i>NSMutableDictionary *</i>
	Omron device settings. For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix-DataStructuresOverviewIOS

### 3.10 `getDeviceSettingsWithUser`

```
(void)getDeviceSettingsWithUser:(int)user withCompletionBlock:(void (^)(id deviceSettings,
NSError *error))completionBlock
```

Return personal settings of a particular user.

Parameters	<i>int user</i>
	Target users you want to acquire
	<i>(void (^)(id deviceSettings, NSError *error))</i>
	Set a completionBlock to be notified of the recording results.

completionBlock

Parameters	<i>NSMutableDictionary * deviceSettings</i>
	Omron device settings. For more information, please refer to the following documents Implementation Guide OmronConnectivityLibraryIOS and Appendix- DataStructuresOverviewIOS
	<i>NSError * error</i>
	Error object is returned in the completionBlock See <a href="#">IV.ErrorCodes</a>

## III. OmronDefines

### 1. OmronDefines

#### 1.1 OMRONVitalDataMeasurementModeTypeKey

Constants	<i>OMRONVitalDataMeasurementModeTypeNormal = 0</i>
	Normal Mode
	<i>OMRONVitalDataMeasurementModeTypeAFib= 1</i>
	Afib Mode
	<i>OMRONVitalDataMeasurementModeTypeNightTimeDesignation = 2</i>
	Night + Time designation (2 am, 4 am, so on)
	<i>OMRONVitalDataMeasurementModeTypeNightElapsedTime = 3</i>
	Night + Elapsed time (4 hours after setting Nocturnal mode)
	<i>OMRONVitalDataMeasurementModeTypeNightElapsedTimeTimeDesignation = 4</i>
	Night + Elapsed time + Time designation (Night mode measurement is started by two settings at the same time)
	<i>OMRONVitalDataMeasurementModeTypeTruRead = 5</i>
	TruRead Mode

#### 1.2 OMRONWheezeTypeKey

Constants	<i>OMRONWheezeTypeUndetected = 0</i>
	Wheeze Not Detected
	<i>OMRONWheezeTypeDetected = 1</i>
	Wheeze detected
	<i>OMRONWheezeTypeError = 2</i>
	Measurement error

#### 1.3 OMRONTemperatureUnitTypeKey

Constants	<i>OMRONTemperatureUnitTypeCelsius = 0</i>
	Celsius
	<i>OMRONTemperatureUnitTypeFahrenheit = 1</i>
	Fahrenheit



## 1.4 OMRONTemperatureLevelTypeKey

Constants	<i>OMRONTemperatureLevelTypeLow = 0</i>
	Low Temperature
	<i>OMRONTemperatureLevelTypeHigh = 1</i>
	High Temperature

## 1.5 OMRONDevicePersonalSettingsUserGenderType

Constants	<i>OMRONDevicePersonalSettingsUserGenderTypeFemale = 0</i>
	Female
	<i>OMRONDevicePersonalSettingsUserGenderTypeMale = 1</i>
	Male

## 1.6 OMRONDevicePersonalSettingsBloodPressureTruReadStatus

Constants	<i>OMRONDevicePersonalSettingsBloodPressureTruReadOff = 0</i>
	TruRead setting is off.
	<i>OMRONDevicePersonalSettingsBloodPressureTruReadOn = 1</i>
	TruRead setting is on.

## 1.7 OMRONDevicePersonalSettingsBloodPressureTruReadInterval

Constants	<i>OMRONDevicePersonalSettingsBloodPressureTruReadInterval15 = 0</i>
	TruRead Interval Time is 15 sec.
	<i>OMRONDevicePersonalSettingsBloodPressureTruReadInterval30 = 1</i>
	TruRead Interval Time is 30 sec.
	<i>OMRONDevicePersonalSettingsBloodPressureTruReadInterval60 = 2</i>
	TruRead Interval Time is 60 sec.
	<i>OMRONDevicePersonalSettingsBloodPressureTruReadInterval120 = 3</i>
	TruRead Interval Time is 120 sec.

## 1.8 OMRONDeviceAlarmStatus

Constants	<i>OMRONDeviceAlarmStatusOff = 0</i>
	Alarm setting is off.
	<i>OMRONDeviceAlarmStatusOn = 1</i>
	Alarm setting is on.

## 1.9 OMRONDeviceAlarmType

Constants	<i>OMRONDeviceAlarmTypeNormal = 0</i>
	Normal alarm
	<i>OMRONDeviceAlarmTypeMeasure = 1</i>
	Blood pressure measurement alarm
	<i>OMRONDeviceAlarmTypeMedication = 2</i>
	Medication alarm
	<i>OMRONDeviceAlarmTypeNone = 3</i>
	None alarm

## 1.10 OMRONDeviceSleepAutomatic

Constants	<i>OMRONDeviceSleepAutomaticOff = 0</i>
	Auto sleep setting is off.
	<i>OMRONDeviceSleepAutomaticOn = 1</i>
	Auto sleep setting is on.

## 1.11 OMRONDeviceTimeFormat

Constants	<i>OMRONDeviceTimeFormat24Hour = 0</i>
	24 Hour Format
	<i>OMRONDeviceTimeFormat12Hour = 1</i>
	12 Hour Format

## 1.12 OMRONDeviceDateFormat

Constants	<i>OMRONDeviceDateFormatMonthDay = 0</i>
	MM/DD Date Format
	<i>OMRONDeviceDateFormatDayMonth = 1</i>
	DD/MM Date Format

## 1.13 OMRONDeviceDistanceUnit

Constants	<i>OMRONDeviceDistanceUnitKilometer = 0</i>
	The distance unit is kilometer.
	<i>OMRONDeviceDistanceUnitMile = 1</i>
	The distance unit is miles.

## 1.14 OMRONDeviceNotificationStatus

Constants	<i>OMRONDeviceNotificationStatusOff = 0</i>
	Notification setting is off.
	<i>OMRONDeviceNotificationStatusOn = 1</i>
	Notification setting is on.

## 1.15 OMRONDeviceWeightDisplay

Constants	<i>OMRONDeviceWeightDisplayOff = 0</i>
	Weight display is disable.
	<i>OMRONDeviceWeightDisplayOn = 1</i>
	Weight display is enabled.

## 1.16 OMRONDeviceWeightUnit

Constants	<i>OMRONDeviceWeightUnitKg = 0</i>
	The weight unit is kilogram.
	<i>OMRONDeviceWeightUnitLbs = 1</i>
	The weight unit is Lbs.
	<i>OMRONDeviceWeightUnitSt = 2</i>
	The weight unit is Stone.

## 1.17 OMRONDeviceScanSettingsMode

Constants	<i>OMRONDeviceScanSettingsModePairing = 0</i>
	Scan for devices in pairing mode.
	<i>OMRONDeviceScanSettingsModeMismatchSequence = 1</i>
	Scan for devices in transfer mode.
	<i>OMRONDeviceScanSettingsModeInvalidTime = 2</i>
	Device scan invalid mode.

## 1.18 OMRONBLEConnectionState

Constants	<i>OMRONBLEConnectionStateUnknown = 0</i>
	Unknown State
	<i>OMRONBLEConnectionStateScanning = 1</i>
	Scanning State
	<i>OMRONBLEConnectionStateConnecting = 2</i>
	Connecting State
	<i>OMRONBLEConnectionStateConnected = 3</i>
	Connected State
	<i>OMRONBLEConnectionStateDisconnecting = 4</i>
	Disconnecting State
	<i>OMRONBLEConnectionStateDisconnect = 5</i>
	Disconnected State

## 1.19 OMRONBLEBluetoothState

Constants	<i>OMRONBLEBluetoothStateUnknown = 0</i>
	Unknown status
	<i>OMRONBLEBluetoothStateOff = 1</i>
	The Bluetooth function of the smartphone is disabled.
	<i>OMRONBLEBluetoothStateOn = 2</i>
	The Bluetooth function of the smartphone is enabled.
	<i>OMRONBLEBluetoothStateUnauthorized = 3</i>
	Bluetooth Unauthorized State by the application

## 1.20 OMRONBLEDeviceCategory

Constants	<i>OMRONBLEDeviceCategoryBloodPressure = 0</i>
	Blood Pressure Category
	<i>OMRONBLEDeviceCategoryBodyComposition = 1</i>
	Body Composition Category
	<i>OMRONBLEDeviceCategoryActivity = 2</i>
	Activity Category
	<i>OMRONBLEDeviceCategoryPulseOximeter = 6</i>
	Pulse Oximeter Category
	<i>OMRONBLEDeviceCategoryWheeze = 14</i>
	Wheeze Category
	<i>OMRONBLEDeviceCategoryTemperature = 19</i>
	Temperature Category

## IV. ErrorCodes

### 1. ErrorCode

CODE	GENERAL DESCRIPTION
6000	Partner key not verified
8002	User hash parameter missing error
8003	Unsupported device error
6011,6017,6100,6206	Sequence error
6020	Multiple connection error
6002,6018,6038,6095,8000,8001,9000	Parameter error
6016	Bluetooth function OFF
6015	Not compatible with Bluetooth
60161	Bluetooth permission error
4353,4354,4356,4357,4358,4359,4360,4361,4362,4363,4364,8193	Audio error
4355,4365	Audio resource contention
6207,6221	Communication conflict error
4366	Microphone permission error
6021,6128	Pairing failure (pairing information error)
6022	Pairing failure (device communication mode)
6024	Pairing failure (user cancellation)
6029	Scan timeout
6030	Communication timeout
6019,6223	Unintentional disconnection
All Other Error Codes	Other errors

See below for error handling.

Implementation Guide OmronConnectivityLibraryIOS

### 2. NSError

The NSError object is used to capture and return error information.

For more information, see [Foundation Framework Documentation].

See <https://developer.apple.com/documentation/foundation/nSError>

code example:

```
if(error == nil) {  
    // Processing in the absence of errors  
} else {  
    ErroCodeText = [NSString stringWithFormat:@"Code: %ld", (long)error.code];  
    DescriptionText = [error localizedDescription];  
}
```