

## Android OPN2002i / OPN3002i Series SDK API Reference Manual



OPN2002i

OPN3002i

## OPN2002i/OPN3002i Series

All information subject to change without notice.

## Document History

<b>Model Number:</b>	OPN2002 Series / OPN3002 Series	<b>Specification Number:</b>	SI12025
<b>Edition:</b>	1.2	<b>Original Spec Number:</b>	
<b>Date:</b>	2/12/2013		

Copyright 2013 Opticon. All rights reserved.

This manual may not, in whole or in part, be copied, photocopied, reproduced, translated or converted to any electronic or machine readable form without prior written consent of Opticon.

## Limited Warranty and Disclaimers

PLEASE READ THIS MANUAL CAREFULLY BEFORE INSTALLING OR USING THE PRODUCT.

### Serial Number

A serial number appears on all Opticon products. This official registration number is directly related to the device purchased. Do not remove the serial number from your Opticon device. Removing the serial number voids the warranty.

### Warranty

Unless otherwise agreed in a written contract, all Opticon products are warranted against defects in materials and workmanship for two years after purchase. Opticon will repair or, at its option, replace products that are defective in materials or workmanship with proper use during the warranty period. Opticon is not liable for damages caused by modifications made by a customer. In such cases, standard repair charges will apply. If a product is returned under warranty and no defect is found, standard repair charges will apply. Opticon assumes no liability for any direct, indirect, consequential or incidental damages arising out of use or inability to use both the hardware and software, even if Opticon has been informed about the possibility of such damages.

### Packaging

The packing materials are recyclable. We recommend that you save all packing material to use should you need to transport your scanner or send it for service. Damage caused by improper packaging during shipment is not covered by the warranty.

### Trademarks

Trademarks used are the property of their respective owners.

Opticon Inc. and Opticon Sensors Europe B.V. are wholly owned subsidiaries of OPTOELECTRONICS Co., Ltd., 12-17, Tsukagoshi 4-chome, Warabi-shi, Saitama, Japan 335-0002. TEL +81-(0) 48-446-1183; FAX +81-(0) 48-446-1184

---

## SUPPORT

### USA

Phone: 800-636-0090  
Email: [support@opticonusa.com](mailto:support@opticonusa.com)  
Web: [www.opticonusa.com](http://www.opticonusa.com)

### Europe

Email: [support@opticon.com](mailto:support@opticon.com)  
Web: [www.opticon.com](http://www.opticon.com)

---

## Revision Record

DOC\_ID: SI12025

Version: 1.2

Model Number: OPN2002 Series / OPN3002 Series

Version	Date	Description of Changes	Content
1.0	8/10/2012	—	New Document
1.1	10/10/2012	—	Formatting changed. BluetoothService.clearCallback () added
1.2	2/12/2013	—	OPN3002 support functions added

# Contents

Forward .....	3
Namespace Index .....	5
Hierarchy Index .....	6
Structural Index .....	8
File Index .....	10
Namespace .....	12
Class .....	14
File .....	101

## **Forward**

This document is primarily for aiding the development of Android applications that utilize the OPN2002i and OPN3002i data collectors. For details regarding device use or technical information necessary for Android development, please refer to reference documents that cover those subjects.

Explicit details regarding use of the API within this SDK and command references are not included with this document. Please refer to the "OPN2002i/OPN3002i SDK Users Guide" for more information, or the API calls, parameters, etc.

## API Documentation

File Name: bluetoothservice.jar

### Namespace Index

#### Package

The below list contains a summary of packages.

<a href="#">jp.co.opto.opnsdk</a> .....	11
<a href="#">jp.co.opto.opnsdk.callback</a> .....	11
<a href="#">jp.co.opto.opnsdk.observer</a> .....	11
<a href="#">jp.co.opto.opnsdk.setting</a> .....	11
<a href="#">jp.co.opto.opnsdk.setting.enums</a> .....	11

---

## Hierarchy Index

### Class Hierarchy

The list below is sorted but not in alphabetical order:

jp.co.opto.opnsdk.setting.enums.AckNakControl.....	13
jp.co.opto.opnsdk.setting.enums.AckNakWaitTime .....	13
jp.co.opto.opnsdk.setting.enums.AddonWaitTime .....	14
jp.co.opto.opnsdk.setting.enums.Authentication.....	15
jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime .....	16
jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime .....	18
jp.co.opto.opnsdk.BluetoothServiceState.....	27
jp.co.opto.opnsdk.setting.BluetoothSettings.....	28
jp.co.opto.opnsdk.setting.enums.BuzzerMode .....	32
jp.co.opto.opnsdk.setting.enums.BuzzerPeriod .....	33
jp.co.opto.opnsdk.setting.enums.BuzzerTone.....	34
jp.co.opto.opnsdk.setting.enums.BuzzerVolume .....	35
jp.co.opto.opnsdk.setting.enums.CapsLockMode .....	35
jp.co.opto.opnsdk.Command .....	36
jp.co.opto.opnsdk.setting.enums.CommandResponse.....	39
jp.co.opto.opnsdk.setting.enums.ConnectionMode.....	39
jp.co.opto.opnsdk.setting.DecoderSettings .....	40
jp.co.opto.opnsdk.setting.enums.EAN128_Mode .....	47
jp.co.opto.opnsdk.setting.enums.ErrorMessageMode.....	48
jp.co.opto.opnsdk.setting.enums.FunctionButtonInput .....	49
jp.co.opto.opnsdk.setting.enums.GoodReadLED_ONTime .....	52
jp.co.opto.opnsdk.setting.enums.IATA_CalculateCD.....	53
jp.co.opto.opnsdk.callback.IBluetoothCallback< T >.....	54
jp.co.opto.opnsdk.observer.IBluetoothObserver .....	55
jp.co.opto.opnsdk.setting.enums.InterCharacterDelay .....	56
jp.co.opto.opnsdk.setting.enums.KeyboardLanguage .....	58
jp.co.opto.opnsdk.setting.enums.MemoryOutputMethod .....	60
jp.co.opto.opnsdk.setting.enums.MSIPlessey_CalculateCD .....	61
jp.co.opto.opnsdk.setting.enums.MSIPlessey_TransmitCD .....	62
jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime .....	62
jp.co.opto.opnsdk.setting.enums.NumPadMode .....	64
jp.co.opto.opnsdk.setting.enums.NW7_CalculateCD .....	64
jp.co.opto.opnsdk.setting.enums.NW7_TransmitSTSP .....	65
jp.co.opto.opnsdk.setting.PrefixSettings .....	73
jp.co.opto.opnsdk.setting.enums.ReadableTime.....	77
jp.co.opto.opnsdk.setting.enums.ReadMode .....	78
jp.co.opto.opnsdk.setting.enums.RedundantReading .....	79
jp.co.opto.opnsdk.setting.Settings.....	81
jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime .....	83
jp.co.opto.opnsdk.setting.SuffixSettings.....	84
jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection .....	88
jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection .....	89
jp.co.opto.opnsdk.setting.enums.UPCA_TransmitCD.....	91
jp.co.opto.opnsdk.setting.enums.UPCE_TransmitCD .....	92

BroadcastReceiver	
jp.co.opto.opnsdk.BluetoothService .....	20
jp.co.opto.opnsdk.Opn2002BluetoothService .....	66
IBluetoothCallback	
jp.co.opto.opnsdk.callback.DoDiscoveryCallback .....	46
jp.co.opto.opnsdk.callback.Opn2002Callback< T > .....	71
Opn2002Callback	
jp.co.opto.opnsdk.callback.GetBatteryLevelCallback .....	50
jp.co.opto.opnsdk.callback.GetDateAndTimeCallback .....	51
jp.co.opto.opnsdk.callback.GetFirmwareVersionCallback .....	51
jp.co.opto.opnsdk.callback.GetSettingCallback .....	52
jp.co.opto.opnsdk.callback.VoidCallback .....	93
jp.co.opto.opnsdk.callback.EnableAckNakCallback .....	48
jp.co.opto.opnsdk.callback.SetDateAndTimeCallback .....	81

---



## Structural Index

### Structure

This section lists class, structures, unions, and interfaces:

<a href="#">jp.co.opto.opnsdk.setting.enums.AckNakControl</a>	13
<a href="#">jp.co.opto.opnsdk.setting.enums.AckNakWaitTime</a>	13
<a href="#">jp.co.opto.opnsdk.setting.enums.AddonWaitTime</a>	14
<a href="#">jp.co.opto.opnsdk.setting.enums.Authentication</a>	15
<a href="#">jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime</a>	16
<a href="#">jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime</a>	18
<a href="#">jp.co.opto.opnsdk.BluetoothService</a>	20
<a href="#">jp.co.opto.opnsdk.BluetoothServiceState</a>	27
<a href="#">jp.co.opto.opnsdk.setting.BluetoothSettings</a>	28
<a href="#">jp.co.opto.opnsdk.setting.enums.BuzzerMode</a>	32
<a href="#">jp.co.opto.opnsdk.setting.enums.BuzzerPeriod</a>	33
<a href="#">jp.co.opto.opnsdk.setting.enums.BuzzerTone</a>	34
<a href="#">jp.co.opto.opnsdk.setting.enums.BuzzerVolume</a>	35
<a href="#">jp.co.opto.opnsdk.setting.enums.CapsLockMode</a>	35
<a href="#">jp.co.opto.opnsdk.Command</a>	36
<a href="#">jp.co.opto.opnsdk.setting.enums.CommandResponse</a>	39
<a href="#">jp.co.opto.opnsdk.setting.enums.ConnectionMode</a>	39
<a href="#">jp.co.opto.opnsdk.setting.DecoderSettings</a>	40
<a href="#">jp.co.opto.opnsdk.callback.DoDiscoveryCallback</a>	46
<a href="#">jp.co.opto.opnsdk.setting.enums.EAN128_Mode</a>	47
<a href="#">jp.co.opto.opnsdk.callback.EnableAckNakCallback</a>	48
<a href="#">jp.co.opto.opnsdk.setting.enums.ErrorMessageMode</a>	48
<a href="#">jp.co.opto.opnsdk.setting.enums.FunctionButtonInput</a>	49
<a href="#">jp.co.opto.opnsdk.callback.GetBatteryLevelCallback</a>	50
<a href="#">jp.co.opto.opnsdk.callback.GetDateAndTimeCallback</a>	51
<a href="#">jp.co.opto.opnsdk.callback.GetFirmwareVersionCallback</a>	51
<a href="#">jp.co.opto.opnsdk.callback.GetSettingCallback</a>	52
<a href="#">jp.co.opto.opnsdk.setting.enums.GoodReadLED_ONTime</a>	52
<a href="#">jp.co.opto.opnsdk.setting.enums.IATA_CalculateCD</a>	53
<a href="#">jp.co.opto.opnsdk.callback.IBluetoothCallback&lt; T &gt;</a>	54
<a href="#">jp.co.opto.opnsdk.observer.IBluetoothObserver</a>	55
<a href="#">jp.co.opto.opnsdk.setting.enums.InterCharacterDelay</a>	56
<a href="#">jp.co.opto.opnsdk.setting.enums.KeyboardLanguage</a>	58
<a href="#">jp.co.opto.opnsdk.setting.enums.MemoryOutputMethod</a>	60
<a href="#">jp.co.opto.opnsdk.setting.enums.MSIPlessey_CalculateCD</a>	61
<a href="#">jp.co.opto.opnsdk.setting.enums.MSIPlessey_TransmitCD</a>	62
<a href="#">jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime</a>	62
<a href="#">jp.co.opto.opnsdk.setting.enums.NumPadMode</a>	64
<a href="#">jp.co.opto.opnsdk.setting.enums.NW7_CalculateCD</a>	64
<a href="#">jp.co.opto.opnsdk.setting.enums.NW7_TransmitSTSP</a>	65
<a href="#">jp.co.opto.opnsdk.Opn2002BluetoothService</a>	66
<a href="#">jp.co.opto.opnsdk.callback.Opn2002Callback&lt; T &gt;</a>	71
<a href="#">jp.co.opto.opnsdk.setting.PrefixSettings</a>	73
<a href="#">jp.co.opto.opnsdk.setting.enums.ReadableTime</a>	77

<a href="#"><u>jp.co.opto.opnsdk.setting.enums.ReadMode</u></a> .....	78
<a href="#"><u>jp.co.opto.opnsdk.setting.enums.RedundantReading</u></a> .....	79
<a href="#"><u>jp.co.opto.opnsdk.callback.SetDateAndTimeCallback</u></a> .....	81
<a href="#"><u>jp.co.opto.opnsdk.setting.Settings</u></a> .....	81
<a href="#"><u>jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime</u></a> .....	83
<a href="#"><u>jp.co.opto.opnsdk.setting.SuffixSettings</u></a> .....	84
<a href="#"><u>jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection</u></a> .....	88
<a href="#"><u>jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection</u></a> .....	89
<a href="#"><u>jp.co.opto.opnsdk.setting.enums.UPCA TransmitCD</u></a> .....	91
<a href="#"><u>jp.co.opto.opnsdk.setting.enums.UPCE TransmitCD</u></a> .....	92
<a href="#"><u>jp.co.opto.opnsdk.callback.VoidCallback</u></a> .....	93

---

## File Index

### File Summary

The below list contains a summary of files.

src/jp/co/opto/opnsdk/ <a href="#">BluetoothService.java</a>	94
src/jp/co/opto/opnsdk/ <a href="#">BluetoothServiceState.java</a>	94
src/jp/co/opto/opnsdk/ <a href="#">Command.java</a>	95
src/jp/co/opto/opnsdk/ <a href="#">Opn2002BluetoothService.java</a>	96
src/jp/co/opto/opnsdk/callback/ <a href="#">DoDiscoveryCallback.java</a>	94
src/jp/co/opto/opnsdk/callback/ <a href="#">EnableAckNakCallback.java</a>	94
src/jp/co/opto/opnsdk/callback/ <a href="#">GetBatteryLevelCallback.java</a>	94
src/jp/co/opto/opnsdk/callback/ <a href="#">GetDateAndTimeCallback.java</a>	94
src/jp/co/opto/opnsdk/callback/ <a href="#">GetFirmwareVersionCallback.java</a>	94
src/jp/co/opto/opnsdk/callback/ <a href="#">GetSettingCallback.java</a>	95
src/jp/co/opto/opnsdk/callback/ <a href="#">IBluetoothCallback.java</a>	95
src/jp/co/opto/opnsdk/callback/ <a href="#">Opn2002Callback.java</a>	95
src/jp/co/opto/opnsdk/callback/ <a href="#">SetDateAndTimeCallback.java</a>	95
src/jp/co/opto/opnsdk/callback/ <a href="#">VoidCallback.java</a>	95
src/jp/co/opto/opnsdk/observer/ <a href="#">IBluetoothObserver.java</a>	95
src/jp/co/opto/opnsdk/setting/ <a href="#">BluetoothSettings.java</a>	96
src/jp/co/opto/opnsdk/setting/ <a href="#">DecoderSettings.java</a>	96
src/jp/co/opto/opnsdk/setting/ <a href="#">PrefixSettings.java</a>	101
src/jp/co/opto/opnsdk/setting/ <a href="#">SettingArea.java</a>	101
src/jp/co/opto/opnsdk/setting/ <a href="#">Settings.java</a>	101
src/jp/co/opto/opnsdk/setting/ <a href="#">SuffixSettings.java</a>	101
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">AckNakControl.java</a>	96
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">AckNakWaitTime.java</a>	96
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">AddonWaitTime.java</a>	96
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">Authentication.java</a>	96
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">AutoDisconnectionTime.java</a>	96
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">AutoReconnectionTime.java</a>	97
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">BuzzerMode.java</a>	97
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">BuzzerPeriod.java</a>	97
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">BuzzerTone.java</a>	97
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">BuzzerVolume.java</a>	97
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">CapsLockMode.java</a>	97
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">CommandResponse.java</a>	97
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">ConnectionMode.java</a>	98
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">EAN128 Mode.java</a>	98
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">ErrorMessageMode.java</a>	98
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">FunctionButtonInput.java</a>	98
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">GoodReadLED_ONTime.java</a>	98
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">IATA CalculateCD.java</a>	98
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">InterCharacterDelay.java</a>	98
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">KeyboardLanguage.java</a>	98

src/jp/co/opto/opnsdk/setting/enums/ <a href="#">MemoryOutputMethod.java</a>	99
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">MSIPlessey CalculateCD.java</a>	99
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">MSIPlessey TransmitCD.java</a>	99
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">MultipleReadResetTime.java</a>	99
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">NumPadMode.java</a>	99
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">NW7 CalculateCD.java</a>	99
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">NW7 TransmitSTSP.java</a>	100
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">ReadableTime.java</a>	100
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">ReadMode.java</a>	100
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">RedundantReading.java</a>	100
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">SlaveConnectionWaitTime.java</a>	100
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">TriggerButtonPushTimeForConnection.java</a>	100
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">TriggerButtonPushTimeForDisconnection.java</a>	100
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">UPCA TransmitCD.java</a>	100
src/jp/co/opto/opnsdk/setting/enums/ <a href="#">UPCE TransmitCD.java</a>	100

---

## Namespace

---

### Package: jp.co.opto.opnsdk

#### Package

- package [callback](#)
- package [observer](#)
- package [setting](#)

#### Structure

- class [BluetoothService](#)
- enum [BluetoothServiceState](#)
- class [Command](#)
- class [Opn2002BluetoothService](#)

---

### Package: jp.co.opto.opnsdk.callback

#### Structure

- class [DoDiscoveryCallback](#)
- class [EnableAckNakCallback](#)
- class [GetBatteryLevelCallback](#)
- class [GetDateAndTimeCallback](#)
- class [GetFirmwareVersionCallback](#)
- class [GetSettingCallback](#)
- interface [IBluetoothCallback< T >](#)
- class [Opn2002Callback< T >](#)
- class [SetDateAndTimeCallback](#)
- class [VoidCallback](#)

---

### Package: jp.co.opto.opnsdk.observer

#### Structure

- interface [IBluetoothObserver](#)

---

### Package: jp.co.opto.opnsdk.setting

#### Package

- package [enums](#)

#### Structure

- class [BluetoothSettings](#)
- class [DecoderSettings](#)
- class [PrefixSettings](#)
- class [SettingArea](#)
- class [Settings](#)
- class [SuffixSettings](#)

---

### Package: jp.co.opto.opnsdk.setting.enums

#### Structure

- enum [AckNakControl](#)
- enum [AckNakWaitTime](#)
- enum [AddonWaitTime](#)

- enum [Authentication](#)
  - enum [AutoDisconnectionTime](#)
  - enum [AutoReconnectionTime](#)
  - enum [BuzzerMode](#)
  - enum [BuzzerPeriod](#)
  - enum [BuzzerTone](#)
  - enum [BuzzerVolume](#)
  - enum [CapsLockMode](#)
  - enum [CommandResponse](#)
  - enum [ConnectionMode](#)
  - enum [EAN128\\_Mode](#)
  - enum [ErrorMessageMode](#)
  - enum [FunctionButtonInput](#)
  - enum [GoodReadLED\\_ONTime](#)
  - enum [IATA\\_CalculateCD](#)
  - enum [InterCharacterDelay](#)
  - enum [KeyboardLanguage](#)
  - enum [MemoryOutputMethod](#)
  - enum [MSIPlessey\\_CalculateCD](#)
  - enum [MSIPlessey\\_TransmitCD](#)
  - enum [MultipleReadResetTime](#)
  - enum [NumPadMode](#)
  - enum [NW7\\_CalculateCD](#)
  - enum [NW7\\_TransmitSTSP](#)
  - enum [ReadableTime](#)
  - enum [ReadMode](#)
  - enum [RedundantReading](#)
  - enum [SlaveConnectionWaitTime](#)
  - enum [TriggerButtonPushTimeForConnection](#)
  - enum [TriggerButtonPushTimeForDisconnection](#)
  - enum [UPCA\\_TransmitCD](#)
  - enum [UPCE\\_TransmitCD](#)
-

## Class

### jp.co.opto.opnsdk.setting.enums.AckNakControl Enum Reference

#### Public Method

- String [toString](#) ()

#### Static Public Method

- static [AckNakControl searchByValue](#) (int value)

#### Public Variable

- [Disable](#) =(0, "Disable")
- [Enable](#) =(1, "Enable")
- [EnableNoResponse](#) =(2, "YES (No response)")

---

## Explanation

This class is responsible for the ACK/NAK controls.

---

## Method

static [AckNakControl](#) jp.co.opto.opnsdk.setting.enums.AckNakControl.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.AckNakControl.toString ()

---

## Variable

jp.co.opto.opnsdk.setting.enums.AckNakControl.Disable =(0, "NO")  
NO

jp.co.opto.opnsdk.setting.enums.AckNakControl.Enable =(1, "YES")  
YES

jp.co.opto.opnsdk.setting.enums.AckNakControl.EnableNoResponse =(2, "YES (NoResponse) ")  
YES (No Response)

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[AckNakControl.java](#)

---

### jp.co.opto.opnsdk.setting.enums.AckNakWaitTime Enum Reference

#### Public Method

- String [toString](#) ()

#### Static Public Method

- static [AckNakWaitTime searchByValue](#) (int value)

#### Public Variable

- [100msec](#) =(5, "100ms")
- [500msec](#) =(25, "500ms")
- [1000msec](#) =(50, "1000ms")

### Explanation

This class is responsible for the ACK/NAK wait time.

---

### Method

static [AckNakWaitTime](#) jp.co.opto.opnsdk.setting.enums.AckNakWaitTime.searchByValue (int value)[static]

String jp.co.opto.opnsdk.setting.enums.AckNakWaitTime.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.AckNakWaitTime.\_1000msec =(50, "1000ms")  
1s

jp.co.opto.opnsdk.setting.enums.AckNakWaitTime.\_100msec =(5, "100ms")  
100ms

jp.co.opto.opnsdk.setting.enums.AckNakWaitTime.\_500msec =(25, "500ms")  
500ms

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[AckNakWaitTime.java](#)

---

### jp.co.opto.opnsdk.setting.enums.AddonWaitTime Enum Reference

#### Public Method

- String [toString](#) ()

#### Static Public Method

- static [AddonWaitTime](#) [searchByValue](#) (int value)

#### Public Variable

- [Disable](#) =(0, "Nothing")
- [250msec](#) =(13, "250ms")
- [500msec](#) =(25, "500ms")
- [750msec](#) =(38, "750ms")

---

### Explanation

This class is responsible for the “add on timer”.

---



## Method

static [AddonWaitTime](#) jp.co.opto.opnsdk.setting.enums.AddonWaitTime.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.AddonWaitTime.toString ()

---

## Variable

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_20sec =(13, "250 sec")

250ms

jp.co.opto.opnsdk.setting.enums.AddonWaitTime.\_500msec =(25, "500ms")

500ms

jp.co.opto.opnsdk.setting.enums.AddonWaitTime.\_750msec =(38, "750ms")

750ms

jp.co.opto.opnsdk.setting.enums.AddonWaitTime.Disable =(0, "NO")

NO

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[AddonWaitTime.java](#)

---

## jp.co.opto.opnsdk.setting.enums.Authentication Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [Authentication](#) [searchByValue](#) (int *value*)

### Public Variable

- [Disable](#) =(0, "Disable verification")
- [EveryTime](#) =(1, "Enable verification (every time)")
- [FirstTime](#) =(7, "Enable verification (first time only)")

---

## Explanation

This class is responsible for authentication processing.

---

## Method

static [Authentication](#) jp.co.opto.opnsdk.setting.enums.Authentication.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.Authentication.toString ()

---

## Variable

**jp.co.opto.opnsdk.setting.enums.Authentication.Disable = (0, "NO Authentication")**

NO Authentication

**jp.co.opto.opnsdk.setting.enums.Authentication.EveryTime = (1, "YES Authentication (every time) ")**

YES Authentication (every time)

**jp.co.opto.opnsdk.setting.enums.Authentication.FirstTime = (7, "YES Authentication (first time only) ")**

YES Authentication (first time only)

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[Authentication.java](#)

---

## jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [AutoDisconnectionTime searchByValue](#) (int value)

### Public Variable

- [Disable](#) =(0, "Disable")
- [10min](#) =(30000, "10 min")
- [20min](#) =(60000, "20 min")
- [30min](#) =(90000, "30min")
- [40min](#) =(120000, "40 min")
- [50min](#) =(150000, "50 min")
- [60min](#) =(180000, "60 min")
- [1min](#) =(3000, "1 min")
- [2min](#) =(6000, "2 min")
- [3min](#) =(9000, "3 min")
- [4min](#) =(12000, "4 min")
- [5min](#) =(15000, "5 min")
- [6min](#) =(18000, "6 min")
- [7min](#) =(21000, "7 min")
- [8min](#) =(24000, "8 min")
- [9min](#) =(27000, "9 min")
- [10sec](#) =(500, "10 sec")
- [20sec](#) =(1000, "20 sec")
- [30sec](#) =(1500, "30 sec")
- [40sec](#) =(2000, "40 sec")
- [50sec](#) =(2500, "50 sec")

---

## Explanation

This class is responsible for the auto disconnect period.

---

## Method

static [AutoDisconnectionTime](#) jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.searchByValue  
(int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.toString ()

---

## Variable

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_10min =(30000, "10 min")

10 min

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_10sec =(500, "10 sec")

10 sec

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_1min =(3000, "1 min")

1 min

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_20min =(60000, "20 min")

20 min

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_20sec =(1000, "20 sec")

20 sec

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_2min =(6000, "2 min")

2 min

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_30min =(90000, "30 min")

30 min

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_30sec =(1500, "30 sec")

30 sec

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_3min =(9000, "3 min")

3 min

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_40min =(120000, "40 min")

40 min

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_40sec =(2000, "40 sec")

40 sec

jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_4min =(12000, "4 min")

4 min

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_50min** =(150000, "50 min")

50 min

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_50sec** =(2500, "50 sec")

50 sec

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_5min** =(15000, "5分")

5 min

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_60min** =(180000, "60 min")

60 min

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_6min** =(18000, "6 min")

6 min

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_7min** =(21000, "7 min")

7 min

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_8min** =(24000, "8 min")

8 min

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.\_9min** =(27000, "9 min")

9 min

**jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime.Disable** =(0, "Disable")

Disable

---

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/AutoDisconnectionTime.java](#)

---

## **jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime Enum Reference**

### **Public Method**

- String [toString](#) ()

### **Static Public Method**

- static [AutoReconnectionTime searchByValue](#) (int value)

### **Public Variable**

- [Disable](#) =(0, "Disable")
- [1min](#) =(3000, "1 min")
- [2min](#) =(6000, "2 min")
- [3min](#) =(9000, "3 min")
- [4min](#) =(12000, "4min")
- [5min](#) =(15000, "5 min")
- [6min](#) =(18000, "6 min")
- [7min](#) =(21000, "7 min")

- [8min](#) =(24000, "8 min")
- [9min](#) =(27000, "9 min")
- [10min](#) =(30000, "10 min")
- [11min](#) =(33000, "11 min")
- [12min](#) =(36000, "12 min")
- [13min](#) =(39000, "13 min")
- [14min](#) =(42000, "14 min")
- [15min](#) =(45000, "15 min")

---

### Explanation

This class is responsible for the effective auto reconnection period.

---

### Method

static [AutoReconnectionTime](#) `jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime.searchByValue`  
(int *value*)[static]

String `jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime.toString ()`

---

### Variable

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._10min` =(30000, "10 min")  
10 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._11min` =(33000, "11 min")  
11 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._12min` =(36000, "12 min")  
12 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._13min` =(39000, "13 min")  
13 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._14min` =(42000, "14 min")  
14 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._15min` =(45000, "15 min")  
15 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._1min` =(3000, "1 min")  
1 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._2min` =(6000, "2 min")  
2 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._3min =(9000, "3 min")`

3 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._4min =(12000, "4 min")`

4 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._5min =(15000, "5 min")`

5 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._6min =(18000, "6 min")`

6 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._7min =(21000, "7 min")`

7 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._8min =(24000, "8 min")`

8 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime._9min =(27000, "9 min")`

9 min

`jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime.Disable =(0, "Disable")`

Disable

---

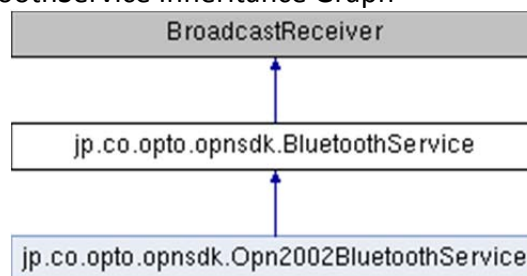
The documentation for this enum was generated from the following file:

- `src/jp/co/opto/opnsdk/setting/enums/AutoReconnectionTime.java`

---

## Class `jp.co.opto.opnsdk.BluetoothService`

`jp.co.opto.opnsdk.BluetoothService` Inheritance Graph



### Public Method

- [BluetoothService](#) (Context context)
- void [clearCallback](#) ()
- void [addObserver](#) ([IBluetoothObserver](#) observer)
- void [removeObserver](#) ([IBluetoothObserver](#) observer)
- BluetoothDevice [getConnetedDevice](#) ()
- boolean [hasBluetoothAdapter](#) ()

- boolean [showBluetoothSetting](#) (Context context)
- boolean [isEnabled](#) ()
- boolean [showBluetoothSetting](#) (Activity activity)
- boolean [showDiscoverable](#) (Context context)
- boolean [isDiscoverable](#) ()
- Set< BluetoothDevice > [getPairedDevices](#) ()
- void [doDiscovery](#) (Context context, [DoDiscoveryCallback](#) callback)
- void [cancelDiscovery](#) ()
- void [onReceive](#) (Context context, Intent intent)
- synchronized byte [getResponseValue](#) ()
- synchronized [BluetoothServiceState](#) [getState](#) ()
- synchronized void [start](#) ()
- synchronized void [connect](#) (String address, boolean secure)
- synchronized void [connect](#) (BluetoothDevice device, boolean secure)
- synchronized void [stop](#) ()
- void [write](#) (String out)
- boolean [isWriteFinished](#) ()

#### Static Public Method

- static Charset [getCharSet](#) ()
- static byte[] [getBytesFrom](#) (String str)
- static String [getStringFrom](#) (byte[] bytes)
- static void [setCharSet](#) (Charset charSet)

#### Static Public Variable

- static byte [ESC](#) = 0x1b
- static byte [CR](#) = 0x0d
- static byte [ACK](#) = 0x06
- static byte [NAK](#) = 0x15
- static byte [STX](#) = 0x02
- static byte [ETX](#) = 0x03

#### Protected Method

- void [addCallback](#) (IBluetoothCallback<?> callback)
- void [removeCallback](#) (IBluetoothCallback<?> callback)
- void [postObserver](#) (final int postType, final Object...obj)
- void [postObserver](#) (final [IBluetoothObserver](#) observer, final int postType, final Object[] obj)
- void [write](#) (byte[] out)
- void [write](#) (List< byte[] > commandList)

#### Protected Variable

- List< [IBluetoothObserver](#) > [observers](#) = new ArrayList<[IBluetoothObserver](#)>()
- List< IBluetoothCallback<?> > [callbackList](#) = new ArrayList<IBluetoothCallback<?>>()
- final int [POST\\_CONNECTED](#) = 1
- final int [POST\\_CONNECT\\_FAILED](#) = 2
- final int [POST\\_CONNECTION\\_LOST](#) = 3
- final int [POST\\_RECEIVE](#) = 4

#### Static Protected Variable

- static Charset [mCharSet](#) = Charset.forName("Shift-jis")

---

#### Explanation

This class enables Bluetooth SPP connection to remote devices.

#### Constructor and Destructor

#### jp.co.opto.opnsdk.BluetoothService.BluetoothService (Context context)

This is a provisional constructor.

**Argument:**

<i>context</i>	<i>context</i>
----------------	----------------

---

**Method**

**void jp.co.opto.opnsdk.BluetoothService.addCallback (IBluetoothCallback<?> *callback*)[protected]**

This adds the callback.

Pre-added callbacks will not be newly-added.

**Argument:**

<i>callback</i>	The callback to be added.
-----------------	---------------------------

**void jp.co.opto.opnsdk.BluetoothService.addObserver ([IBluetoothObserver](#) *observer*)**

This adds the observer.

All managed observers are notified of events created by transmitting to remote devices (Connection Complete, Connection Failure, Disconnect, and Receive Data).

Pre-added observers will not be newly-added.

**Argument:**

<i>observer</i>	The observer to be added.
-----------------	---------------------------

**void jp.co.opto.opnsdk.BluetoothService.cancelDiscovery ()**

Bluetooth device search cancels via the doDiscovery method.

**void jp.co.opto.opnsdk.BluetoothService.clearCallback ()**

All callbacks are forgotten.

**Argument:**

--	--

**synchronized void jp.co.opto.opnsdk.BluetoothService.connect (String *address*, boolean *secure*)**

Begin transmission via Client Mode to test SPP connection to the Bluetooth device (with BD address).

This method will switch the connection state to "BluetoothServiceState.connecting". Calling this method when in "BluetoothServiceState.connected" or BluetoothServiceState.connecting will release each of these connections.

**Argument:**

<i>address</i>	The remote device's BD address.
<i>secure</i>	When encrypted "true". If not, "false".

**Exception:**

<i>IllegalArgument Exception</i>	When the address is illegal.
--------------------------------------	------------------------------



**synchronized void jp.co.opto.opnsdk.BluetoothService.connect (BluetoothDevice *device*, boolean *secure*)**

Begin transmission via Client Mode to test SPP connection with the Bluetooth device. This method will switch the connection state to "BluetoothServiceState.connecting". Calling this method when in "BluetoothServiceState.connected" or BluetoothServiceState.connecting will release each of these connections.

**Argument:**

<i>device</i>	Remote device.
<i>secure</i>	When encrypted "true". If not, "false".

**void jp.co.opto.opnsdk.BluetoothService.doDiscovery (Context *context*, [DoDiscoveryCallback](#) *callback*)**

This searches for unpaired Bluetooth devices. Processing for this method is conducted asynchronously. The callback is implemented when processing finishes completion. If the search is already in progress, this process is canceled prior to implementation.

**Argument:**

<i>context</i>	<i>Context requesting the search (Activity, Service, etc.).</i>
<i>callback</i>	This is the callback which processes the search result.

**static byte [] jp.co.opto.opnsdk.BluetoothService.getBytesFrom (String *str*)[static]**

Received data is returned via the character code set within the class.

**Return Value:**

This returns the value converted by the character code specified via received data.

**static Charset jp.co.opto.opnsdk.BluetoothService.getCharSet ()[static]**

This returns the received data's character code.

**Return Value:**

The currently set character code (default value: Shift-JIS).

**BluetoothDevice jp.co.opto.opnsdk.BluetoothService.getConnetedDevice ()**

This returns the currently connected device.

**Return Value:**

The currently connected device (null when disconnected).

**Set<BluetoothDevice> jp.co.opto.opnsdk.BluetoothService.getPairedDevices ()**

This returns paired Bluetooth devices, and "null" is returned when the Android terminal does not support Bluetooth. An "empty set" object is returned when there are no paired devices.

**Return Value:**

This returns the paired device.

**synchronized byte jp.co.opto.opnsdk.BluetoothService.getResponseValue ()**

This captures data added to the end of strings received from the remote device.

**Return Value:**

1 byte of received data's end.

synchronized [BluetoothServiceState](#) **jp.co.opto.opnsdk.BluetoothService.getState ()**

This returns the connection state.

**Return Value:**

The connection state.

static String **jp.co.opto.opnsdk.BluetoothService.getStringFrom (byte[] bytes)[static]**

Specified byte strings are configured via character codes set in the class and returned.

**Argument:**

<i>bytes</i>	Converted subject data.
--------------	-------------------------

**Return Value:**

Converted value.

boolean **jp.co.opto.opnsdk.BluetoothService.hasBluetoothAdapter ()**

This returns whether or not the Android terminal supports Bluetooth.

**Return Value:**

If supported, "true". If unsupported, "false".

boolean **jp.co.opto.opnsdk.BluetoothService.isDiscoverable ()**

This returns whether or not the state is searchable via other Bluetooth devices.

It returns "false" when the Android terminal does not support Bluetooth.

**Return Value:**

If it is a searchable state, "true". Otherwise, "false".

boolean **jp.co.opto.opnsdk.BluetoothService.isEnabled ()**

This returns whether Bluetooth is enabled or not.

It returns "false" when the Android terminal does not support Bluetooth.

**Return Value:**

If enabled, "true". If disabled, "false".

boolean **jp.co.opto.opnsdk.BluetoothService.isWriteFinished ()**

void **jp.co.opto.opnsdk.BluetoothService.onReceive (Context context, Intent intent)**

void **jp.co.opto.opnsdk.BluetoothService.postObserver (final int postType, final Object... obj)[protected]**

This preforms notification processing for all observers managed by this object.

**Argument:**

<i>postType</i>	Notification Types (POST_CONNECTED, POST_CONNECT_FAILED、POST_CONNECTION_LOST、POST_RECEIVE).
<i>obj</i>	These are parameters also passed when notifying.

void **jp.co.opto.opnsdk.BluetoothService.postObserver (final [BluetoothObserver](#) observer, final int postType, final Object[] obj)[protected]**

This preforms notification processing in the observer (argument observer).

There are four different notification types:

- [BluetoothService#POST\\_CONNECTED](#)
- [BluetoothService#POST\\_CONNECT\\_FAILED](#)
- [BluetoothService#POST\\_CONNECTION\\_LOST](#)
- [BluetoothService#POST\\_RECEIVE](#)

**Argument:**

<i>observer</i>	The observer to be notified.
<i>postType</i>	The notification type.
<i>obj</i>	These are parameters also passed when notification occurs.

**void `jp.co.opto.opnsdk.BluetoothService.removeCallback (IBluetoothCallback<?> callback)`**[protected]

This deletes the callback.

**Argument:**

<i>callback</i>	The callback to be deleted.
-----------------	-----------------------------

**void `jp.co.opto.opnsdk.BluetoothService.removeObserver (IBluetoothObserver observer)`**

This deletes the observer.

**Argument:**

<i>observer</i>	The observer to be deleted.
-----------------	-----------------------------

**static void `jp.co.opto.opnsdk.BluetoothService.setCharSet (Charset charSet)`**[static]

This sets the received data's character code.

**Argument:**

<i>charSet</i>	The character code to be set.
----------------	-------------------------------

**boolean `jp.co.opto.opnsdk.BluetoothService.showBluetoothSetting (Context context)`**

This makes a request to the Android device to display the setting screen for enabling Bluetooth.

**Argument:**

<i>context</i>	The context which attempts to display the setting screen.
----------------	---

**Return Value:**

If the setting screen is displayed, true. If not, false.

**boolean `jp.co.opto.opnsdk.BluetoothService.showBluetoothSetting (Activity activity)`**

This makes a request to the Android device to display the setting screen for enabling Bluetooth.

**Argument:**

<i>activity</i>	The activity which attempts to display the setting screen.
-----------------	--

**Return Value:**

If the setting screen is displayed, true. If not, false.

**boolean `jp.co.opto.opnsdk.BluetoothService.showDiscoverable (Context context)`**

This makes a request to the Android device to display the setting screen used to activate search functionality in other Bluetooth devices. (Status will continue for a 300 second period.)

**Argument:**

<i>context</i>	The context which attempts to display the setting screen.
----------------	---

**Return Value:**

If the setting screen is displayed, true. If not, false.

**synchronized void jp.co.opto.opnsdk.BluetoothService.start ()**

Server Mode is utilized to start transmitting and waits for an SPP connection attempt from the device. This method will switch the connection state to "BluetoothServiceState.listen". Calling this method when in "BluetoothServiceState.connected" or "BluetoothServiceState.connecting" will release either of the connections. If a request from another device is made and connection confirmed, the wait state is released and switches to "BluetoothServiceState.connected". This supports both encryption and anti-encryption connection requests.

**synchronized void jp.co.opto.opnsdk.BluetoothService.stop ()**

All processing involved with SPP connection is stopped, and the state is set as "BluetoothServiceState.none". This method is usable regardless of the status of the BluetoothServiceState.

**void jp.co.opto.opnsdk.BluetoothService.write (String out)**

This sends character strings to a remote device. Before transmitting, the string is encoded and the write(byte[]) method is called. When changing the encoding type be sure to utilize the setCharSet method before calling this one.

**Argument:**

	The transmitted data string.
--	------------------------------

**void jp.co.opto.opnsdk.BluetoothService.write (byte[] out)[protected]**

This transmits byte arrays to a remote device.

This method is unable to detect whether or not the transmission is completed.

**Argument:**

	The byte array being transmitted.
--	-----------------------------------

**void jp.co.opto.opnsdk.BluetoothService.write (List< byte[]> commandList)[protected]**

This transmits multiple byte arrays to a remote device. After transmitting data to the remote device, the data end sent from the partner must be either ACK or NAK. If the data end is ACK the next byte array is transmitted; if NAK, the transmission is terminated. This method is unable to detect whether or not the transmission is completed.

**Argument:**

<i>commandList</i>	The list of transmitted byte arrays.
--------------------	--------------------------------------

---

**Variable**

**byte jp.co.opto.opnsdk.BluetoothService.ACK = 0x06[static]**

The ACK control character.

**List<IBluetoothCallback<?>> > jp.co.opto.opnsdk.BluetoothService.callbackList = new  
ArrayList<IBluetoothCallback<?>>()[protected]**

The list of synchronously processed callbacks implemented via this class.

**byte jp.co.opto.opnsdk.BluetoothService.CR = 0x0d[static]**

The CR control character.

**byte jp.co.opto.opnsdk.BluetoothService.ESC = 0x1b[static]**

The ESC control character.

**byte jp.co.opto.opnsdk.BluetoothService.ETX = 0x03[static]**

The ETX control character.

**Charset jp.co.opto.opnsdk.BluetoothService.mCharSet = Charset.forName("Shift-jis")[static], [protected]**

The encoding of character codes used in receiving and transmitting data.

**byte jp.co.opto.opnsdk.BluetoothService.NAK = 0x15[static]**

The NAK control character.

**List<[IBluetoothObserver](#)> jp.co.opto.opnsdk.BluetoothService.observers = new  
ArrayList<[IBluetoothObserver](#)>()[protected]**

This contains the list of objects and observers necessary to notify users of events created via this class. When an event is created the lists are notified to all registered objects.

**final int jp.co.opto.opnsdk.BluetoothService.POST\_CONNECT\_FAILED = 2[protected]**

The constant when the observer is notified of a "connection failure" state.

**final int jp.co.opto.opnsdk.BluetoothService.POST\_CONNECTED = 1[protected]**

The constant when the observer is notified of a "connection completed" state.

**final int jp.co.opto.opnsdk.BluetoothService.POST\_CONNECTION\_LOST = 3[protected]**

The constant when the observer is notified of a "disconnect" state.

**final int jp.co.opto.opnsdk.BluetoothService.POST\_RECEIVE = 4[protected]**

The constant when the observer is notified of a "data received" state.

**byte jp.co.opto.opnsdk.BluetoothService.STX = 0x02[static]**

The STX control character.

---

The explanation for this class was created from the file below:

- [src/jp/co/opto/opnsdk/BluetoothService.java](#)

---

**jp.co.opto.opnsdk.BluetoothServiceState Enum Reference**

**Public Variable**

- [none](#)
- [listen](#)
- [connecting](#)

## Explanation

This is the enumerated type for activating the BluetoothService “connection” state.

---

## Variable

### **jp.co.opto.opnsdk.BluetoothServiceState.connecting**

The Begin Connecting state

### **jp.co.opto.opnsdk.BluetoothServiceState.listen**

The Waiting for Connection state

### **jp.co.opto.opnsdk.BluetoothServiceState.none**

“None” state

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/[BluetoothServiceState.java](#)

---

## **Class jp.co.opto.opnsdk.setting.BluetoothSettings**

### **Public Method**

- [BluetoothSettings](#) (byte[] settingData)
- [ReadMode](#) [getReadMode](#) ()
- [ReadableTime](#) [getReadableTime](#) ()
- [RedundantReading](#) [getRedundantReading](#) ()
- [MultipleReadResetTime](#) [getMultipleReadResetTime](#) ()
- [AddonWaitTime](#) [getAddonWaitTime](#) ()
- [BuzzerVolume](#) [getBuzzerVolume](#) ()
- [GoodReadLED\\_ONTime](#) [getGoodReadLED\\_ONTime](#) ()
- boolean [isTriggerModeEnable](#) ()
- boolean [isTriggerRepeatEnable](#) ()
- boolean [isBuzzerEnable](#) ()
- [BuzzerTone](#) [getBuzzerTone](#) ()
- [BuzzerPeriod](#) [getBuzzerPeriod](#) ()
- [BuzzerMode](#) [getBuzzerMode](#) ()
- String [getDestinationAddress](#) ()
- [Authentication](#) [getAuthentication](#) ()
- boolean [isEncryptionEnable](#) ()
- [CommandResponse](#) [getCommandResponse](#) ()
- String [getPinCode](#) ()
- boolean [isTriggerConnectionEnable](#) ()
- boolean [isAddressBarcodeConnectionEnable](#) ()
- [AckNakControl](#) [getAckNakControl](#) ()
- [ConnectionMode](#) [getConnectionMode](#) ()
- [SlaveConnectionWaitTime](#) [getSlaveConnectionWaitTime](#) ()
- [AutoReconnectionTime](#) [getAutoReconnectionTime](#) ()
- [AutoDisconnectionTime](#) [getAutoDisconnectionTime](#) ()
- [TriggerButtonPushTimeForConnection](#) [getTriggerButtonPushTimeForConnection](#) ()
- [TriggerButtonPushTimeForDisconnection](#) [getTriggerButtonPushTimeForDisconnection](#) ()
- [AckNakWaitTime](#) [getAckNakWaitTime](#) ()
- boolean [isOutOfRangeMemoryEnable](#) ()

- boolean [isDataCollectModeEnable \(\)](#)
- boolean [isAutoConnectionEnable \(\)](#)
- boolean [isDisconnectionBuzzerFromDataCollector \(\)](#)
- boolean [isDisconnectionBuzzerFromDestination \(\)](#)
- [MemoryOutputMethod](#) [getMemoryOutputMethod \(\)](#)
- boolean [isUSBConnection \(\)](#)
- [FunctionButtonInput](#) [getFunctionButtonInput \(\)](#)
- String [getDeviceName \(\)](#)
- String [getDestinationAddressLastTime \(\)](#)
- [InterCharacterDelay](#) [getInterCharacterDelay \(\)](#)
- [KeyboardLanguage](#) [getKeyboardLanguage \(\)](#)
- boolean [isUseNumPad \(\)](#)
- [NumPadMode](#) [getNumPadMode \(\)](#)
- [CapsLockMode](#) [getCapsLockMode \(\)](#)
- boolean [isGoodReadVibratorEnable \(\)](#)

---

### Explanation

This is the class which activates settings related to Bluetooth transmission.

---

### Constructor and Destructor

**jp.co.opto.opnsdk.setting.BluetoothSettings.BluetoothSettings (byte[] *settingData*)**

This is the constructor.

**Argument:**

<i>settingData</i>	Setting data (format when transmitting).
--------------------	--

---

### Method

**[AckNakControl](#) jp.co.opto.opnsdk.setting.BluetoothSettings.getAckNakControl ()**

This returns the ACK/NAK control settings.

**[AckNakWaitTime](#) jp.co.opto.opnsdk.setting.BluetoothSettings.getAckNakWaitTime ()**

This returns the ACK/NAK Wait Time settings.

**[AddonWaitTime](#) jp.co.opto.opnsdk.setting.BluetoothSettings.getAddonWaitTime ()**

This returns the Add On Timer settings.

**[Authentication](#) jp.co.opto.opnsdk.setting.BluetoothSettings.getAuthentication ()**

This returns the connected partner address settings.

**[AutoDisconnectionTime](#) jp.co.opto.opnsdk.setting.BluetoothSettings.getAutoDisconnectionTime ()**

This returns the Auto Disconnection Time settings.

**[AutoReconnectionTime](#) jp.co.opto.opnsdk.setting.BluetoothSettings.getAutoReconnectionTime ()**

This returns the Auto Reconnection Time settings.

**BuzzerMode** `jp.co.opto.opnsdk.setting.BluetoothSettings.getBuzzerMode ()`

This returns the Buzzer Noise Timing settings.

**BuzzerPeriod** `jp.co.opto.opnsdk.setting.BluetoothSettings.getBuzzerPeriod ()`

This returns the Buzzer Noise Duration settings.

**BuzzerTone** `jp.co.opto.opnsdk.setting.BluetoothSettings.getBuzzerTone ()`

This returns the Buzzer Tone settings.

**BuzzerVolume** `jp.co.opto.opnsdk.setting.BluetoothSettings.getBuzzerVolume ()`

This returns the Buzzer Volume settings.

**CapsLockMode** `jp.co.opto.opnsdk.setting.BluetoothSettings.getCapsLockMode ()`

This returns the Caps Lock Mode settings.

**CommandResponse** `jp.co.opto.opnsdk.setting.BluetoothSettings.getCommandResponse ()`

This returns the Command Response settings.

**ConnectionMode** `jp.co.opto.opnsdk.setting.BluetoothSettings.getConnectionMode ()`

This returns the Connection Mode settings.

**String** `jp.co.opto.opnsdk.setting.BluetoothSettings.getDestinationAddress ()`

This returns the connected partner address settings.

**String** `jp.co.opto.opnsdk.setting.BluetoothSettings.getDestinationAddressLastTime ()`

This returns the previous slave connection partner address settings.

**String** `jp.co.opto.opnsdk.setting.BluetoothSettings.getDeviceName ()`

This returns the Bluetooth Device Name settings.

**FunctionButtonInput** `jp.co.opto.opnsdk.setting.BluetoothSettings.getFunctionButtonInput ()`

This returns the output settings when the function button is pressed.

**GoodReadLED\_ONTime** `jp.co.opto.opnsdk.setting.BluetoothSettings.getGoodReadLED_ONTime ()`

This returns the LED lighting duration settings.

**InterCharacterDelay** `jp.co.opto.opnsdk.setting.BluetoothSettings.getInterCharacterDelay ()`

This returns the Intercharacter Delay settings.

**KeyboardLanguage** `jp.co.opto.opnsdk.setting.BluetoothSettings.getKeyboardLanguage ()`

This returns the Keyboard Language settings.

**MemoryOutputMethod** `jp.co.opto.opnsdk.setting.BluetoothSettings.getMemoryOutputMethod ()`

This returns the Memory Data Output Process settings.

**MultipleReadResetTime** `jp.co.opto.opnsdk.setting.BluetoothSettings.getMultipleReadResetTime ()`

This returns the Multiple Read Reset Time settings.



**[NumPadMode](#)** `jp.co.opto.opnsdk.setting.BluetoothSettings.getNumPadMode ()`

This returns the NumPad Mode settings.

**`String jp.co.opto.opnsdk.setting.BluetoothSettings.getPinCode ()`**

This returns the PIN Code settings.

**[ReadableTime](#)** `jp.co.opto.opnsdk.setting.BluetoothSettings.getReadableTime ()`

This returns the Readable Time settings.

**[ReadMode](#)** `jp.co.opto.opnsdk.setting.BluetoothSettings.getReadMode ()`

This returns the Read Mode settings.

**[RedundantReading](#)** `jp.co.opto.opnsdk.setting.BluetoothSettings.getRedundantReading ()`

This returns the Check Quantity settings.

**[SlaveConnectionWaitTime](#)** `jp.co.opto.opnsdk.setting.BluetoothSettings.getSlaveConnectionWaitTime ()`

This returns the Slave Connection Wait Time settings.

**[TriggerButtonPushTimeForConnection](#)**

**`jp.co.opto.opnsdk.setting.BluetoothSettings.getTriggerButtonPushTimeForConnection ()`**

This returns the Trigger Button Push Time for Connection settings.

**[TriggerButtonPushTimeForDisconnection](#)**

**`jp.co.opto.opnsdk.setting.BluetoothSettings.getTriggerButtonPushTimeForDisconnection ()`**

This returns the Trigger Button Push Time for Disconnection settings.

**`boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isAddressBarcodeConnectionEnable ()`**

This returns connection process settings via address barcode scanning.

**`boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isAutoConnectionEnable ()`**

This returns automatic connection settings via address barcode scanning.

**`boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isBuzzerEnable ()`**

This returns the Buzzer Noise settings.

**`boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isDataCollectModeEnable ()`**

This returns the data collect settings.

**`boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isDisconnectionBuzzerFromDataCollector ()`**

This returns the data collector 'buzzer disconnect' settings.

**`boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isDisconnectionBuzzerFromDestination ()`**

This returns the connected partner 'buzzer disconnect' settings.

**`boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isEncryptionEnable ()`**

This returns the Enable Encryption settings.

**boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isOutOfRangeMemoryEnable ()**

This returns the Out of Range Memory settings.

**boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isTriggerConnectionEnable ()**

This returns the Trigger Connection settings.

**boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isTriggerModeEnable ()**

This returns the Trigger Mode settings.

**boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isTriggerRepeatEnable ()**

This returns the Trigger Repeat settings.

**boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isUSBConnection ()**

This returns the USB Connection COM Transmit settings.

**boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isUseNumPad ()**

This returns the Use NumPad settings.

**boolean jp.co.opto.opnsdk.setting.BluetoothSettings.isGoodReadVibratorEnable ()**

This returns the Good Read Enable Vibration settings.

The explanation for this class was created from the file below:

- [src/jp/co/opto/opnsdk/setting/BluetoothSettings.java](#)

## jp.co.opto.opnsdk.setting.enums.BuzzerMode Enum Reference

### Public Method

- String [toString \(\)](#)

### Static Public Method

- static [BuzzerMode searchByValue](#) (int value)

### Public Variable

- [BeforeTransmission](#) =(0, "Pre Transmit Buzzer")
- [AfterTransmission](#) =(1, "Post Transmit Buzzer")

### Explanation

This class is responsible for the buzzer vibration timing.

### Method

static [BuzzerMode](#) jp.co.opto.opnsdk.setting.enums.BuzzerMode.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.BuzzerMode.toString ()

### Variable

jp.co.opto.opnsdk.setting.enums.BuzzerMode.AfterTransmission =(1, "Post Transmit Buzzer")

Post Transmit Buzzer

**jp.co.opto.opnsdk.setting.enums.BuzzerMode.BeforeTrasmission** =(0, "Pre Transmit Buzzer")

Pre Transmit Buzzer

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/BuzzerMode.java](#)

---

#### **jp.co.opto.opnsdk.setting.enums.BuzzerPeriod Enum Reference**

##### **Public Method**

- String [toString](#) ()

##### **Static Public Method**

- static [BuzzerPeriod searchByValue](#) (int value)

##### **Public Variable**

- [100msec](#) =(5, "100ms")
- [200msec](#) =(10, "200ms")
- [400msec](#) =(20, "400ms")
- [50msec](#) =(2, "50ms")

---

#### **Explanation**

This class is responsible for the buzzer vibration duration.

---

#### **Method**

static [BuzzerPeriod](#) jp.co.opto.opnsdk.setting.enums.BuzzerPeriod.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.BuzzerPeriod.toString ()

---

#### **Variable**

jp.co.opto.opnsdk.setting.enums.BuzzerPeriod.\_100msec =(5, "100ms")  
100ms

jp.co.opto.opnsdk.setting.enums.BuzzerPeriod.\_200msec =(10, "200ms")  
200ms

jp.co.opto.opnsdk.setting.enums.BuzzerPeriod.\_400msec =(20, "400ms")  
400ms

jp.co.opto.opnsdk.setting.enums.BuzzerPeriod.\_50msec =(2, "50ms")  
50ms

---

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/BuzzerPeriod.java](#)
-

## jp.co.opto.opnsdk.setting.enums.BuzzerTone Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [BuzzerTone searchByValue](#) (int value)

### Public Variable

- [SingleTone](#) =(0, "single tone")
- [HighLow](#) =(1, "high low")
- [LowHigh](#) =(2, "low high")
- [4500Hz](#) =(3, "4.5KHz")
- [2200Hz 2000Hz](#) =(4, "2.2KHz-2KHz")

---

### Explanation

This class is responsible for the buzzer tone.

---

### Method

static [BuzzerTone](#) jp.co.opto.opnsdk.setting.enums.BuzzerTone.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.BuzzerTone.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.BuzzerTone.\_2200Hz\_2000Hz =(4, "2.2KHz-2KHz")  
2.2KHz-2KHz

jp.co.opto.opnsdk.setting.enums.BuzzerTone.\_4500Hz =(3, "4.5KHz")  
4.5KHz

jp.co.opto.opnsdk.setting.enums.BuzzerTone.HighLow =(1, "high low")  
High Low

jp.co.opto.opnsdk.setting.enums.BuzzerTone.LowHigh =(2, "low high")  
Low High

jp.co.opto.opnsdk.setting.enums.BuzzerTone.SingleTone =(0, "single tone")  
Single Tone

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[BuzzerTone.java](#)
-

## jp.co.opto.opnsdk.setting.enums.BuzzerVolume Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [BuzzerVolume searchByValue](#) (int value)

### Public Variable

- [MAX](#) =(127, "Maximum")
- [Large](#) =(32, "Large")
- [Medium](#) =(8, "Medium")
- [Small](#) =(1, "Small")

### Explanation

This class is responsible for the buzzer volume.

---

### Method

static [BuzzerVolume](#) jp.co.opto.opnsdk.setting.enums.BuzzerVolume.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.BuzzerVolume.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.BuzzerVolume.Large =(32, "Large")

Large

jp.co.opto.opnsdk.setting.enums.BuzzerVolume.MAX =(127, "MAX")

MAX

jp.co.opto.opnsdk.setting.enums.BuzzerVolume.Medium =(8, "Medium")

Medium

jp.co.opto.opnsdk.setting.enums.BuzzerVolume.Small =(1, "Small")

Small

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[BuzzerVolume.java](#)

---

## jp.co.opto.opnsdk.setting.enums.CapsLockMode Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [CapsLockMode searchByValue](#) (int value)

### Public Variable

- [Auto](#) =(0, "Auto Caps Lock Mode")
  - [OFF](#) =(1, "Caps Lock Mode: Off")
  - [ON](#) =(2, "Caps Lock Mode: On")
-

## Explanation

This class is responsible for the caps lock mode.

## Method

static [CapsLockMode](#) jp.co.opto.opnsdk.setting.enums.CapsLockMode.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.CapsLockMode.toString ()

---

## Variable

jp.co.opto.opnsdk.setting.enums.CapsLockMode.Auto =(0, "Auto Caps Lock Mode")

Auto Caps Lock Mode

jp.co.opto.opnsdk.setting.enums.CapsLockMode.OFF =(1, "Caps Lock Mode: Off")

Caps Lock Mode: Off

jp.co.opto.opnsdk.setting.enums.CapsLockMode.ON =(2, "Caps Lock Mode: On")

Caps Lock Mode: On

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[CapsLockMode.java](#)

---

## Class jp.co.opto.opnsdk.Command

### Structure

- enum **SettingChar**

### Static Public Variable

- static final String [INITIALIZE\\_SETTING\\_ALL](#) = "U1"
- static final String [INITIALIZE\\_SETTING\\_BLUETOOTH](#) = "]INIT"
- static final String [GET\\_FIRMWARE\\_VERSION](#) = "Z1"
- static final String [SAVE\\_SETTING](#) = "Z2"
- static final String [READ\\_BARCODE](#) = "Z"
- static final String [GET\\_STORED\\_BARCODE\\_ALL](#) = "[EBD"
- static final String [ERASE\\_STORED\\_BARCODE\\_ALL](#) = "+-MCLR-+"
- static final String [SET\\_DATE\\_AND\\_TIME\\_START](#) = "[EBE"
- static final String [SET\\_DATE\\_AND\\_TIME\\_END](#) = "[EBF"
- static final String [GET\\_DATE\\_AND\\_TIME](#) = "[EBG"
- static final String [GET\\_BATTERY\\_LEVEL](#) = "[BAL"
- static final String [GET\\_SETTING](#) = "[XZ5"
- static final String [SET\\_PIN\\_START](#) = "]PINS"
- static final String [SET\\_PIN\\_END](#) = "]PINE"
- static final String [SET\\_REMOTE\\_DEVICE\\_ADDRESS\\_START](#) = "]BDAS"
- static final String [SET\\_REMOTE\\_DEVICE\\_ADDRESS\\_END](#) = "]BDAE"
- static final String [SET\\_DEVICE\\_NAME\\_START](#) = "[E65"
- static final String [SET\\_DEVICE\\_NAME\\_END](#) = "[E66"
- static final String [SETTING\\_SIGNAL](#) = "ZZ"
- static final String [SETTING\\_PERMISSION](#) = "Z7"
- static final String [ENABLE\\_ACKNAK\\_CONTROL](#) = "P3"

- static final String [RESPONSE\\_COMMAND](#) = "WC"
- static final String [DISABLE\\_TRIGGER](#) = "S7"
- static final String [ENABEL\\_TRIGGER](#) = "S 8 "
- static final String [VIBRATOR\\_ON](#) = "V"
- static final String [BEEP\\_BUZZER](#) = "B"

---

### Explanation

This class defines basic commands among those transmittable to the data collector. For commands that are not defined by this class, please refer to the SDK Reference Manual or User's Manual.

---

### Variable

**final String jp.co.opto.opnsdk.Command.DISABLE\_TRIGGER = "S7"[static]**

Disable Trigger command

**final String jp.co.opto.opnsdk.Command.ENABEL\_TRIGGER = "S 8 "[static]**

Enable Trigger command

**final String jp.co.opto.opnsdk.Command.ENABLE\_ACKNAK\_CONTROL = "P3"[static]**

Enable ACK/NAK Controls command

**final String jp.co.opto.opnsdk.Command.ERASE\_STORED\_BARCODE\_ALL = "+-MCLR+"[static]**

Clear Memory Data command

**final String jp.co.opto.opnsdk.Command.GET\_BATTERY\_LEVEL = "[BAL"[static]**

Return Battery Voltage command

**final String jp.co.opto.opnsdk.Command.GET\_DATE\_AND\_TIME = "[EBG"[static]**

Return Data and Time command

**final String jp.co.opto.opnsdk.Command.GET\_FIRMWARE\_VERSION = "Z1"[static]**

Display Version command

**final String jp.co.opto.opnsdk.Command.GET\_SETTING = "[XZ5"[static]**

Return Current Settings command

**final String jp.co.opto.opnsdk.Command.GET\_STORED\_BARCODE\_ALL = "[EBD"[static]**

Output Memory Data command

**final String jp.co.opto.opnsdk.Command.INITIALIZE\_SETTING\_ALL = "U1"[static]**

Initialize Settings command

**final String jp.co.opto.opnsdk.Command.INITIALIZE\_SETTING\_BLUETOOTH = "]INIT"[static]**

Initialize Bluetooth Settings command

**final String jp.co.opto.opnsdk.Command.READ\_BARCODE = "Z"[static]**

Begin Reading Barcode command

**final String jp.co.opto.opnsdk.Command.RESPONSE\_COMMAND = "WC"[static]**

Command Response Settings command

**final String jp.co.opto.opnsdk.Command.SAVE\_SETTING = "Z2"[static]**

Save Settings command

**final String jp.co.opto.opnsdk.Command.SET\_DATE\_AND\_TIME\_END = "[EBF"[static]**

Finish RTC Settings command

**final String jp.co.opto.opnsdk.Command.SET\_DATE\_AND\_TIME\_START = "[EBE"[static]**

Begin RTC Settings command

**final String jp.co.opto.opnsdk.Command.SET\_DEVICE\_NAME\_END = "[E66"[static]**

Finish Device Name Settings command

**final String jp.co.opto.opnsdk.Command.SET\_DEVICE\_NAME\_START = "[E65"[static]**

Begin Device Name Settings command

**final String jp.co.opto.opnsdk.Command.SET\_PIN\_END = "]PINE"[static]**

Finish PIN Settings command

**final String jp.co.opto.opnsdk.Command.SET\_PIN\_START = "]PINS"[static]**

Begin PIN Settings command

**final String jp.co.opto.opnsdk.Command.SET\_REMOTE\_DEVICE\_ADDRESS\_END = "]BDAE"[static]**

Finish Partner Address Settings command

**final String jp.co.opto.opnsdk.Command.SET\_REMOTE\_DEVICE\_ADDRESS\_START = "]BDAS"[static]**

Begin Partner Address Settings command

**final String jp.co.opto.opnsdk.Command.SETTING\_PERMISSION = "Z7"[static]**

Begin/Finish Enable Reading Settings command

**final String jp.co.opto.opnsdk.Command.SETTING\_SIGNAL = "ZZ"[static]**

Begin/Finish Settings command

**final String jp.co.opto.opnsdk.Command.VIBRATOR\_ON = "V"[static]**

Vibration command

**final String jp.co.opto.opnsdk.Command.BEEP\_BUZZER = "B"[static]**

Beep Buzzer command

---

The explanation for this class was created from the file below:

- src/jp/co/opto/opnsdk/[Command.java](#)



## jp.co.opto.opnsdk.setting.enums.CommandResponse Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [CommandResponse searchByValue](#) (int value)

### Public Variable

- [AckNak](#) =(1, "YES (ACK/NAK) ")
- [UseProperty\\_AckNakControl](#) =(0, "Adhere to AckNakControl property settings")

---

### Explanation

This class is responsible for command responses.

---

### Method

static [CommandResponse](#) jp.co.opto.opnsdk.setting.enums.CommandResponse.searchByValue (int value)[static]

String jp.co.opto.opnsdk.setting.enums.CommandResponse.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.CommandResponse.AckNak =(1, "YES (ACK/NAK) ")  
YES (ACK/NAK)

jp.co.opto.opnsdk.setting.enums.CommandResponse.UseProperty\_AckNakControl =(0, "Adhere to AckNakControl property settings")  
Adhere to AckNakControl property settings

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[CommandResponse.java](#)

---

## jp.co.opto.opnsdk.setting.enums.ConnectionMode Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [ConnectionMode searchByValue](#) (int value)

### Public Variable

- [SPP\\_Master](#) =(0, "SPP Master")
- [SPP\\_Slave](#) =(1, "SPP Slave")
- [HID](#) =(2, "HID")

---

### Explanation

This class is responsible for the mode of connection.

---

## Method

static [ConnectionMode](#) `jp.co.opto.opnsdk.setting.enums.ConnectionMode.searchByValue (int value)[static]`

String `jp.co.opto.opnsdk.setting.enums.ConnectionMode.toString ()`

---

## Variable

`jp.co.opto.opnsdk.setting.enums.ConnectionMode.HID =(2, "HID")`  
HID

`jp.co.opto.opnsdk.setting.enums.ConnectionMode.SPP_Master =(0, "SPP Master")`  
SPP Master

`jp.co.opto.opnsdk.setting.enums.ConnectionMode.SPP_Slave =(1, "SPP Slave")`  
SPP Slave

---

The documentation for this enum was generated from the following file:

- `src/jp/co/opto/opnsdk/setting/enums/`[ConnectionMode.java](#)

---

## Class `jp.co.opto.opnsdk.setting.DecoderSettings`

### Public Method

- [DecoderSettings](#) (byte[] settingData)
- boolean [isEAN13\\_ReadEnable \(\)](#)
- boolean [isEAN13\\_Addon2ReadEnable \(\)](#)
- boolean [isEAN13\\_Addon5ReadEnable \(\)](#)
- boolean [isEAN13\\_TransmitCDEnable \(\)](#)
- boolean [isWPC\\_CaculateCDEnable \(\)](#)
- boolean [isEAN8\\_TransmitCDEnable \(\)](#)
- boolean [isUPCAE\\_ReadEnable \(\)](#)
- boolean [isUPCAE\\_Addon2ReadEnable \(\)](#)
- boolean [isUPCAE\\_Addon5ReadEnable \(\)](#)
- [UPCA\\_TransmitCD](#) [getUPCA\\_TransmitCD \(\)](#)
- [UPCE\\_TransmitCD](#) [getUPCE\\_TransmitCD \(\)](#)
- boolean [isCode39\\_ReadEnable \(\)](#)
- boolean [isCode39\\_CalculateCDEnable \(\)](#)
- boolean [isCode39\\_TransmitCDEnable \(\)](#)
- boolean [isCode39\\_TransmitSTSPEnable \(\)](#)
- boolean [isNW7\\_ReadEnable \(\)](#)
- boolean [isNW7\\_TransmitCDEnable \(\)](#)
- [NW7\\_CalculateCD](#) [getNW7\\_CalculateCD \(\)](#)
- [NW7\\_TransmitSTSP](#) [getNW7\\_TransmitSTSP \(\)](#)
- boolean [isInterleaved2of5\\_ReadEnable \(\)](#)
- boolean [isInterleaved2of5\\_TransmitCDEnable \(\)](#)
- boolean [isInterleaved2of5\\_CalculateCDEnable \(\)](#)
- boolean [isIndustrial2of5\\_ReadEnable \(\)](#)
- boolean [isIndustrial2of5\\_TransmitCDEnable \(\)](#)

- boolean [isIndustrial2of5\\_CalculateCDEnable \(\)](#)
- boolean [isCode93\\_ReadEnable \(\)](#)
- boolean [isCode93\\_CalculateCDEnable \(\)](#)
- boolean [isCode128\\_ReadEnable \(\)](#)
- boolean [isCode128\\_CalculateCDEnable \(\)](#)
- boolean [isSCode\\_ReadEnable \(\)](#)
- [EAN128\\_Mode](#) [getEAN128\\_Mode \(\)](#)
- boolean [isIATA\\_ReadEnable \(\)](#)
- boolean [isIATA\\_TransmitCDEnable \(\)](#)
- [IATA\\_CalculateCD](#) [getIATA\\_CalculateCD \(\)](#)
- boolean [isTelepen\\_ReadEnable \(\)](#)
- boolean [isMatrix2of5\\_ReadEnable \(\)](#)
- boolean [isTriOptic\\_ReadEnable \(\)](#)
- boolean [isMSIPlessey\\_ReadEnable \(\)](#)
- [MSIPlessey\\_TransmitCD](#) [getMSIPlessey\\_TransmitCD \(\)](#)
- [MSIPlessey\\_CalculateCD](#) [getMSIPlessey\\_CalculateCD \(\)](#)
- boolean [isUKPlessey\\_ReadEnable \(\)](#)
- boolean [isGS1DataBar\\_ReadEnable \(\)](#)
- boolean [isGS1DataBarLimited\\_ReadEnable \(\)](#)
- boolean [isGS1Expanded\\_ReadEnable \(\)](#)
- boolean [isCode3of5\\_ReadEnable \(\)](#)
- boolean [isMicroPDF417\\_ReadEnable \(\)](#)
- boolean [isPDF417\\_ReadEnable \(\)](#)
- boolean [isCode11\\_ReadEnable \(\)](#)
- boolean [isGS1DataBar\\_TransmitCDEnable \(\)](#)
- [ErrorMessageMode](#) [getErrorMessageMode \(\)](#)
- byte[] [getErrorMessageNoLabel \(\)](#)
- byte[] [getErrorMessageDecodeFailed \(\)](#)
- boolean [isIntelligentMail\\_ReadEnable \(\)](#)
- boolean [isPostnet\\_ReadEnable \(\)](#)
- boolean [isJapanesePostal\\_ReadEnable \(\)](#)
- boolean [isCodablockF\\_ReadEnable \(\)](#)
- boolean [isDataMatrixECC200\\_ReadEnable \(\)](#)
- boolean [isDataMatrixECC000\\_140\\_ReadEnable \(\)](#)
- boolean [isAztecCode\\_ReadEnable \(\)](#)
- boolean [isAztecRunes\\_ReadEnable \(\)](#)
- boolean [isChineseSensibleCode\\_ReadEnable \(\)](#)
- boolean [isQRcode\\_ReadEnable \(\)](#)
- boolean [isMicroQRcode\\_ReadEnable \(\)](#)
- boolean [isMaxiCode\\_ReadEnable \(\)](#)
- boolean [isCompositeOnGS1Databar\\_ReadEnable \(\)](#)
- boolean [isCompositeOnUPSEAN\\_ReadEnable \(\)](#)

---

#### Explanation

This class displays settings in correlation with the decoder.

---

## Constructor and Destructor

**jp.co.opto.opnsdk.setting.DecoderSettings.DecoderSettings (byte[] *settingData*)**

This is the constructor.

**Argument:**

<i>settingData</i>	Setting data (format when transmitting).
--------------------	--

---

## Method

**[EAN128\\_Mode](#) jp.co.opto.opnsdk.setting.DecoderSettings.getEAN128\_Mode ()**

This returns the ENA-128 Reading Mode settings.

**byte [] jp.co.opto.opnsdk.setting.DecoderSettings.getErrorMessageDecodeFailed ()**

This returns the error message that occurs when decoding fails.

**[ErrorMessageMode](#) jp.co.opto.opnsdk.setting.DecoderSettings.getErrorMessageMode ()**

This returns the error message that occurs when scanning fails.

**byte [] jp.co.opto.opnsdk.setting.DecoderSettings.getErrorMessageNoLabel ()**

This returns the error message that occurs when the label is not found.

**[IATA\\_CalculateCD](#) jp.co.opto.opnsdk.setting.DecoderSettings.getIATA\_CalculateCD ()**

This returns the IATA CD calculation settings.

**[MSIPlessey\\_CalculateCD](#) jp.co.opto.opnsdk.setting.DecoderSettings.getMSIPlessey\_CalculateCD ()**

This returns the MSI/Plessey CD calculation settings.

**[MSIPlessey\\_TransmitCD](#) jp.co.opto.opnsdk.setting.DecoderSettings.getMSIPlessey\_TransmitCD ()**

This returns the MSI/Plessey CD transmit settings.

**[NW7\\_CalculateCD](#) jp.co.opto.opnsdk.setting.DecoderSettings.getNW7\_CalculateCD ()**

This returns the NW-7 CD calculation settings.

**[NW7\\_TransmitSTSP](#) jp.co.opto.opnsdk.setting.DecoderSettings.getNW7\_TransmitSTSP ()**

This returns the NW-7 Start/Stop settings.

**[UPCA\\_TransmitCD](#) jp.co.opto.opnsdk.setting.DecoderSettings.getUPCA\_TransmitCD ()**

This returns the UPC-A CD transmit settings.

**[UPCE\\_TransmitCD](#) jp.co.opto.opnsdk.setting.DecoderSettings.getUPCE\_TransmitCD ()**

This returns the UPC-E CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode11\_ReadEnable ()**

This returns the enable Code11 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode128\_CalculateCDEnable ()**

This returns the Code 93 CD calculation settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode128\_ReadEnable ()**

This returns the enable Code 128 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode39\_CalculateCDEnable ()**

This returns the Code 93 CD calculation settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode39\_ReadEnable ()**

This returns the enable Code 39 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode39\_TransmitCDEnable ()**

This returns the Code 39 CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode39\_TransmitSTSPEnable ()**

This returns the Code 39 Start/Stop settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode3of5\_ReadEnable ()**

This returns the enable Code 3 of 5 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode93\_CalculateCDEnable ()**

This returns the Code 93 CD calculation settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCode93\_ReadEnable ()**

This returns the enable Code 93 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isEAN13\_Addon2ReadEnable ()**

This returns the enable EAN 13/8 (Add On 2) scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isEAN13\_Addon5ReadEnable ()**

This returns the enable EAN 13/8 (Add On 5) scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isEAN13\_ReadEnable ()**

This returns the enable EAN 13/8 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isEAN13\_TransmitCDEnable ()**

This returns the EAN 13 CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isEAN8\_TransmitCDEnable ()**

This returns the EAN 8 CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isGS1DataBar\_ReadEnable ()**

This returns the enable GS1 DataBar scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isGS1DataBar\_TransmitCDEnable ()**

This returns the GS1 DataBar CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isGS1DataBarLimited\_ReadEnable ()**

This returns the enable GS1 DataBar Limited scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isGS1Expanded\_ReadEnable ()**

This returns the enable GS1 DataBar Expanded scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isIATA\_ReadEnable ()**

This returns the enable IATA scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isIATA\_TransmitCDEnable ()**

This returns the IATA CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isIndustrial2of5\_CalculateCDEnable ()**

This returns the Industrial 2 of 5 CD calculation settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isIndustrial2of5\_ReadEnable ()**

This returns the enable Industrial 2 of 5 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isIndustrial2of5\_TransmitCDEnable ()**

This returns the Industrial 2 of 5 CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isInterleaved2of5\_CalculateCDEnable ()**

This returns the Interleaved 2 of 5 CD calculation settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isInterleaved2of5\_ReadEnable ()**

This returns the enable Interleaved 2 of 5 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isInterleaved2of5\_TransmitCDEnable ()**

This returns the Interleaved 2 of 5 CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isMatrix2of5\_ReadEnable ()**

This returns the enable Matrix 2 of 5 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isMicroPDF417\_ReadEnable ()**

This returns the enable Micro PDF 417 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isMSIPlessey\_ReadEnable ()**

This returns the enable MSI/Plessey scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isNW7\_ReadEnable ()**

This returns the enable NW-7 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isNW7\_TransmitCDEnable ()**

This returns the NW-7 CD transmit settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isPDF417\_ReadEnable ()**

This returns the enable PDF 417 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isSCode\_ReadEnable ()**

This returns the enable SCode scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isTelepen\_ReadEnable ()**

This returns the enable Telepen scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isTriOptic\_ReadEnable ()**

This returns the enable Tri-Optic scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isUKPlessey\_ReadEnable ()**

This returns the enable MSI/Plessey scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isUPCAE\_Addon2ReadEnable ()**

This returns the enable UPC-A/E (Add On 2) scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isUPCAE\_Addon5ReadEnable ()**

This returns the enable UPC-A/E (Add On 5) scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isUPCAE\_ReadEnable ()**

This returns the enable UPC-A/E scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isWPC\_CaculateCDEnable ()**

This returns the WPC (UPC, EAN, JAN) CD calculation settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isIntelligentMail\_ReadEnable ()**

This returns the enable Intelligent Mail scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isPostnet\_ReadEnable ()**

This returns the enable Postnet scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isJapanesePostal\_ReadEnable ()**

This returns the enable Japanese Postal scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCodablockF\_ReadEnable ()**

This returns the enable CodablockF scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isDataMatrixECC200\_ReadEnable ()**

This returns the enable Data Matrix ECC200 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isDataMatrixECC000\_140\_ReadEnable ()**

This returns the enable Data Matrix ECC000-140 scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isAztecCode\_ReadEnable ()**

This returns the enable Aztec Code scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isAztecRunes\_ReadEnable ()**

This returns the enable Aztec Runes scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isChineseSensibleCode\_ReadEnable ()**

This returns the enable Chinese Sensible Code scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isQRcode\_ReadEnable ()**

This returns the enable QR code scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isMicroQRcode\_ReadEnable ()**

This returns the enable Micro QR code scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isMaxiCode\_ReadEnable ()**

This returns the enable Maxi Code scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCompositeOnGS1Databar\_ReadEnable ()**

This returns the enable Composite on GS1 Databar scanning settings.

**boolean jp.co.opto.opnsdk.setting.DecoderSettings.isCompositeOnUPCEAN\_ReadEnable ()**

This returns the enable Composite on UPC/EAN scanning settings.

---

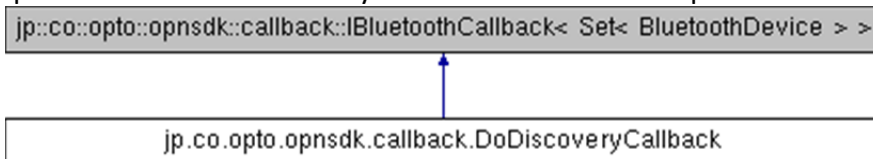
The explanation for this class was created from the file below:

- src/jp/co/opto/opnsdk/setting/[DecoderSettings.java](#)

---

#### Class jp.co.opto.opnsdk.callback.DoDiscoveryCallback

jp.co.opto.opnsdk.callback.DoDiscoveryCallback Inheritance Graph



#### Public Method

- void [execute](#) (Set< BluetoothDevice > devices)
- boolean [canExecute](#) (Set< BluetoothDevice > devices)

#### Protected Method

- abstract void [callbackExecute](#) (Set< BluetoothDevice > devices)
-



### Explanation

This is the abstract class of the callback necessary for performing Bluetooth device searches.

---

### Method

**abstract void jp.co.opto.opnsdk.callback.DoDiscoveryCallback.callbackExecute (Set< BluetoothDevice > *devices*)[protected], [pure virtual]**

This executes the callback. Implementation will be handled via the sub-class.

**Argument:**

<i>devices</i>	The device being searched.
----------------	----------------------------

**boolean jp.co.opto.opnsdk.callback.DoDiscoveryCallback.canExecute (Set< BluetoothDevice > *devices*)**

**void jp.co.opto.opnsdk.callback.DoDiscoveryCallback.execute (Set< BluetoothDevice > *devices*)**

---

The explanation for this class was created from the file below:

- src/jp/co/opto/opnsdk/callback/[DoDiscoveryCallback.java](#)

---

### jp.co.opto.opnsdk.setting.enums.EAN128\_Mode Enum Reference

#### Public Method

- String [toString](#) ()

#### Static Public Method

- static [EAN128\\_Mode searchByValue](#) (int *value*)

#### Public Variable

- [Read\\_IfPossible](#) =(2, "Enable scanning if possible")
- [Read\\_Only](#) =(1, "Enable EAN-128 scanning only")
- [NotRead](#) =(0, "Do not read, output as Code128")

#### Explanation

This class is responsible for scanning EAN-128 symbologies.

---

### Method

**static [EAN128\\_Mode](#) jp.co.opto.opnsdk.setting.enums.EAN128\_Mode.searchByValue (int *value*)[static]**

**String jp.co.opto.opnsdk.setting.enums.EAN128\_Mode.toString ()**

---

### Variable

**jp.co.opto.opnsdk.setting.enums.EAN128\_Mode.NotRead =(0, "Do not read, output as Code128")**

Do not read, output as Code 128

**jp.co.opto.opnsdk.setting.enums.EAN128\_Mode.Read\_IfPossible =(2, "Enable reading if Possible")**

Enable scanning if possible

`jp.co.opto.opnsdk.setting.enums.EAN128_Mode.Read_Only` =(1, "Enable reading EAN-128 Only")

Enable EAN-128 scanning only

---

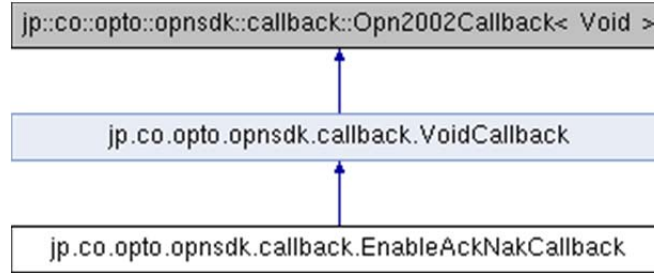
The documentation for this enum was generated from the following file:

- `src/jp/co/opto/opnsdk/setting/enums/EAN128_Mode.java`

---

**Class `jp.co.opto.opnsdk.callback.EnableAckNakCallback`**

`jp.co.opto.opnsdk.callback.EnableAckNakCallback` Inheritance Graph



**Public Method**

- [EnableAckNakCallback](#) ()
- boolean [canExecute](#) (String str)

**Additional Inherited Members**

---

**Explanation**

This is the abstract class of the callback necessary for enabling AckNak.

---

**Constructor and Destructor**

`jp.co.opto.opnsdk.callback.EnableAckNakCallback.EnableAckNakCallback` ()

---

**Method**

boolean `jp.co.opto.opnsdk.callback.EnableAckNakCallback.canExecute` (String str)

---

The explanation for this class was created from the file below:

- `src/jp/co/opto/opnsdk/callback/EnableAckNakCallback.java`

---

**`jp.co.opto.opnsdk.setting.enums.ErrorMessageMode` Enum Reference**

**Public Method**

- String [toString](#) ()

**Static Public Method**

- static [ErrorMessageMode searchByValue](#) (int value)

**Public Variable**

- [Disable](#) =(0, "Do not transmit")
  - [Enable ReferSeparately](#) =(9, "Transmit")
-

### Explanation

This error message displays when a scanning failure occurs.

---

### Method

static [ErrorMessageMode](#) jp.co.opto.opnsdk.setting.enums.ErrorMessageMode.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.ErrorMessageMode.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.ErrorMessageMode.Disable =(0, "Do not transmit")  
Do not transmit

jp.co.opto.opnsdk.setting.enums.ErrorMessageMode.Enable\_ReferSeparately =(9, "Transmit")  
Transmit

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[ErrorMessageMode.java](#)

---

### jp.co.opto.opnsdk.setting.enums.FunctionButtonInput Enum Reference

#### Public Method

- String [toString](#) ()

#### Static Public Method

- static [FunctionButtonInput](#) [searchByValue](#) (int value)

#### Public Variable

- [HT](#) =(0x09, "HT")
- [LF](#) =(0x0A, "LF")
- [CR](#) =(0x0D, "CR")
- [CAN](#) =(0x18, "CAN")
- [ESC](#) =(0x1B, "ESC")

---

### Explanation

This is outputted when the function button is pressed.

---

### Method

static [FunctionButtonInput](#) jp.co.opto.opnsdk.setting.enums.FunctionButtonInput.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.FunctionButtonInput.toString ()

---

## Variable

`jp.co.opto.opnsdk.setting.enums.FunctionButtonInput.CAN =(0x18, "CAN")`

CAN

`jp.co.opto.opnsdk.setting.enums.FunctionButtonInput.CR =(0x0D, "CR")`

CR

`jp.co.opto.opnsdk.setting.enums.FunctionButtonInput.ESC =(0x1B, "ESC")`

ESC

`jp.co.opto.opnsdk.setting.enums.FunctionButtonInput.HT =(0x09, "HT")`

HT

`jp.co.opto.opnsdk.setting.enums.FunctionButtonInput.LF =(0x0A, "LF")`

LF

---

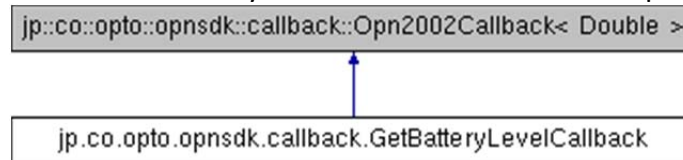
The documentation for this enum was generated from the following file:

- `src/jp/co/opto/opnsdk/setting/enums/FunctionButtonInput.java`

---

## Class `jp.co.opto.opnsdk.callback.GetBatteryLevelCallback`

`jp.co.opto.opnsdk.callback.GetBatteryLevelCallback` Inheritance Graph



## Public Method

- boolean [canExecute](#) (String str)

## Protected Method

- Double [createParameter](#) (String str)

---

## Explanation

This is the abstract class of the callback necessary for capturing the parity value.

---

## Method

**boolean** `jp.co.opto.opnsdk.callback.GetBatteryLevelCallback.canExecute` (String str)

**Double** `jp.co.opto.opnsdk.callback.GetBatteryLevelCallback.createParameter` (String str)[protected]

---

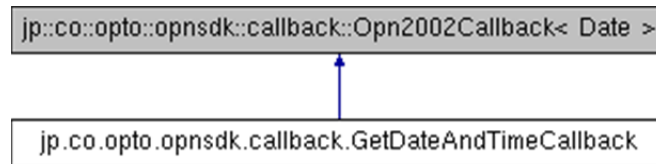
The explanation for this class was created from the file below:

- [src/jp/co/opto/opnsdk/callback/GetBatteryLevelCallback.java](#)

---

#### Class `jp.co.opto.opnsdk.callback.GetDateAndTimeCallback`

`jp.co.opto.opnsdk.callback.GetDateAndTimeCallback` Inheritance Graph



##### Public Method

- boolean [canExecute](#) (String str)

##### Protected Method

- Date [createParameter](#) (String str)

---

##### Explanation

This is the abstract class of the callback necessary for capturing the date and time.

---

##### Method

boolean `jp.co.opto.opnsdk.callback.GetDateAndTimeCallback.canExecute` (String str)

Date `jp.co.opto.opnsdk.callback.GetDateAndTimeCallback.createParameter` (String str)[protected]

---

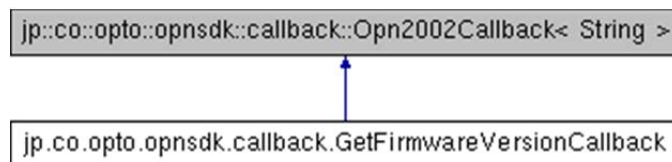
The explanation for this class was created from the file below:

- [src/jp/co/opto/opnsdk/callback/GetDateAndTimeCallback.java](#)

---

#### Class `jp.co.opto.opnsdk.callback.GetFirmwareVersionCallback`

`jp.co.opto.opnsdk.callback.GetFirmwareVersionCallback` Inheritance Graph



##### Public Method

- boolean [canExecute](#) (String str)

##### Protected Method

- String [createParameter](#) (String str)

---

##### Explanation

This is abstract class of the callback necessary for returning version information.

---

## Method

boolean `jp.co.opto.opnsdk.callback.GetFirmwareVersionCallback.canExecute (String str)`

String `jp.co.opto.opnsdk.callback.GetFirmwareVersionCallback.createParameter (String str)[protected]`

---

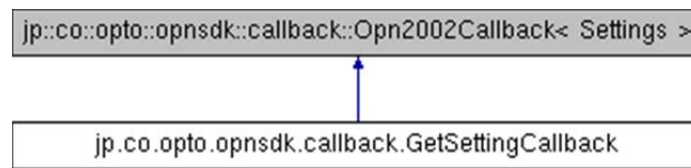
The explanation for this class was created from the file below:

- `src/jp/co/opto/opnsdk/callback/GetFirmwareVersionCallback.java`

---

## Class `jp.co.opto.opnsdk.callback.GetSettingCallback`

`jp.co.opto.opnsdk.callback.GetSettingCallback` Inheritance Graph



### Public Method

- boolean `canExecute (String str)`
- void `execute (String str)`

### Protected Method

- `Settings createParameter (String str)`

---

## Explanation

This is the abstract class of the callback necessary to return all settings.

## Method

boolean `jp.co.opto.opnsdk.callback.GetSettingCallback.canExecute (String str)`

[Settings](#) `jp.co.opto.opnsdk.callback.GetSettingCallback.createParameter (String str)[protected]`

void `jp.co.opto.opnsdk.callback.GetSettingCallback.execute (String str)`

---

The explanation for this class was created from the file below:

- `src/jp/co/opto/opnsdk/callback/GetSettingCallback.java`

---

## `jp.co.opto.opnsdk.setting.enums.GoodReadLED_ONTime` Enum Reference

### Public Method

- String `toString ()`

### Static Public Method

- static `GoodReadLED_ONTime searchByValue (int value)`

### Public Variable

- `Disable` =(0, "Disable")
- `200msec` =(10, "200ms")
- `400msec` =(20, "400ms")
- `800msec` =(40, "800ms")

### Explanation

This class is responsible for the LED lighting time duration.

---

### Method

static [GoodReadLED\\_ONTime](#) jp.co.opto.opnsdk.setting.enums.GoodReadLED\_ONTime.searchByValue (int value)[static]

String jp.co.opto.opnsdk.setting.enums.GoodReadLED\_ONTime.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.GoodReadLED\_ONTime.\_200msec =(10, "200ms")  
200ms

jp.co.opto.opnsdk.setting.enums.GoodReadLED\_ONTime.\_400msec =(20, "400ms")  
400ms

jp.co.opto.opnsdk.setting.enums.GoodReadLED\_ONTime.\_800msec =(40, "800ms")  
800ms

jp.co.opto.opnsdk.setting.enums.GoodReadLED\_ONTime.Disable =(0, "Disable")  
Disable

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[GoodReadLED\\_ONTime.java](#)

---

### jp.co.opto.opnsdk.setting.enums.IATA\_CalculateCD Enum Reference

#### Public Method

- String [toString](#) ()

#### Static Public Method

- static [IATA\\_CalculateCD\\_searchByValue](#) (int value)

#### Public Variable

- [CalculateCD\\_CPN\\_Form\\_Serial](#) =(1, "Calculate CD (CPN+FORM SERIAL)")
- [CalculateCD\\_Form\\_Serial](#) =(2, "Calculate CD (FORM SERIAL)")
- [CalculateCD\\_All\\_Data](#) =(3, "Calculate CD (ALL DATA)")
- [NotCalculateCD](#) =(0, "Do not calculate CD")

---

### Explanation

This class is responsible for calculating the IATA CD.

---

## Method

static [IATA\\_CalculateCD](#) jp.co.opto.opnsdk.setting.enums.IATA\_CalculateCD.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.IATA\_CalculateCD.toString ()

---

## Variable

jp.co.opto.opnsdk.setting.enums.IATA\_CalculateCD.CalculateCD\_All\_Data =(3, "Calculate CD (ALL DATA)")  
Calculate CD (ALL DATA)

jp.co.opto.opnsdk.setting.enums.IATA\_CalculateCD.CalculateCD\_CPN\_Form\_Serial =(1, "Calculate CD (CPN+FORM SERIAL)")  
Calculate CD (CPN+FORM SERIAL)

jp.co.opto.opnsdk.setting.enums.IATA\_CalculateCD.CalculateCD\_Form\_Serial =(2, "Calculate CD (FORM SERIAL)")  
Calculate CD (FORM SERIAL)

jp.co.opto.opnsdk.setting.enums.IATA\_CalculateCD.NotCalculateCD =(0, "Do not calculate CD")  
Do not calculate CD

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[IATA\\_CalculateCD.java](#)

---

Interface jp.co.opto.opnsdk.callback.IBluetoothCallback< T >

## Public Method

- void [execute](#) (T parameter)
- boolean [canExecute](#) (T parameter)

---

## Explanation

This is the interface for the callback which handles asynchronous processing.

## Argument:

<T>	<i>The parameter type when executing the callback.</i>
-----	--

---

## Method

boolean jp.co.opto.opnsdk.callback.IBluetoothCallback< T >.canExecute (T *parameter*)

This determines the executability of the callback.

## Argument:

<i>parameter</i>	Parameters.
------------------	-------------



**Return Value:**

If executable, "true". If not, "false".

**void `jp.co.opto.opnsdk.callback.IBluetoothCallback< T >.execute (T parameter)`**

This method executes the callback once asynchronous processing is completed.

**Argument:**

<i>parameter</i>	Parameters.
------------------	-------------

---

The explanation for this interface was created from the file below:

- `src/jp/co/opto/opnsdk/callback/BluetoothCallback.java`

---

**Interface `jp.co.opto.opnsdk.observer.IBluetoothObserver`**

**Public Method**

- void [connected](#) (BluetoothDevice device)
- void [connectFailed](#) ()
- void [connectionLost](#) ()
- void [receive](#) (String str)

---

**Explanation**

This is the interface for notifying when events are created when transmitting to the remote device.

---

**Method**

**void `jp.co.opto.opnsdk.observer.IBluetoothObserver.connected (BluetoothDevice device)`**

This is executed once connected to a remote device.

**Argument:**

<i>device</i>	The connected remote device.
---------------	------------------------------

**void `jp.co.opto.opnsdk.observer.IBluetoothObserver.connectFailed ()`**

A connection test to the remote device is conducted, and upon failure this method is executed.

**void `jp.co.opto.opnsdk.observer.IBluetoothObserver.connectionLost ()`**

This is executed once connection is released from the remote device.

**void `jp.co.opto.opnsdk.observer.IBluetoothObserver.receive (String str)`**

This is executed when data is received from the remote device.

**Argument:**

<i>str</i>	The received data.
------------	--------------------

---

The explanation for this interface was created from the file below:

- [src/jp/co/opto/opnsdk/observer/IBluetoothObserver.java](#)

---

## jp.co.opto.opnsdk.setting.enums.InterCharacterDelay Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [InterCharacterDelay searchByValue](#) (int value)

### Public Variable

- [0msec](#) =(0, "0ms")
- [10msec](#) =(10, "10ms")
- [20msec](#) =(20, "20ms")
- [30msec](#) =(30, "30ms")
- [40msec](#) =(40, "40ms")
- [50msec](#) =(50, "50ms")
- [60msec](#) =(60, "60ms")
- [70msec](#) =(70, "70ms")
- [80msec](#) =(80, "80ms")
- [90msec](#) =(90, "90ms")
- [100msec](#) =(100, "100ms")
- [150msec](#) =(150, "150ms")
- [200msec](#) =(200, "200ms")
- [250msec](#) =(250, "250ms")
- [300msec](#) =(300, "300ms")
- [350msec](#) =(350, "350ms")
- [400msec](#) =(400, "400ms")
- [450msec](#) =(450, "450ms")
- [500msec](#) =(500, "500ms")
- [550msec](#) =(550, "550ms")
- [600msec](#) =(600, "600ms")

---

### Explanation

This class is responsible for the intercharacter delay.

---

### Method

static [InterCharacterDelay](#) [jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.searchByValue](#) (int value)[static]

String [jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.toString](#) ()

---

### Variable

[jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\\_0msec](#) =(0, "0ms")

0ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_100msec =(100, "100ms")**  
100ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_10msec =(10, "10ms")**  
10ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_150msec =(150, "150ms")**  
150ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_200msec =(200, "200ms")**  
200ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_20msec =(20, "20ms")**  
20ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_250msec =(250, "250ms")**  
250ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_300msec =(300, "300ms")**  
300ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_30msec =(30, "30ms")**  
30ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_350msec =(350, "350ms")**  
350ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_400msec =(400, "400ms")**  
400ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_40msec =(40, "40ms")**  
40ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_450msec =(450, "450ms")**  
450ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_500msec =(500, "500ms")**  
500ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_50msec =(50, "50ms")**  
50ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_550msec =(550, "550ms")**  
550ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_600msec =(600, "600ms")**

600ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_60msec =(60, "60ms")**

60ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_70msec =(70, "70ms")**

70ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_80msec =(80, "80ms")**

80ms

**jp.co.opto.opnsdk.setting.enums.InterCharacterDelay.\_90msec =(90, "90ms")**

90ms

---

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/InterCharacterDelay.java](#)

---

## jp.co.opto.opnsdk.setting.enums.KeyboardLanguage Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [KeyboardLanguage searchByValue](#) (int value)

### Public Variable

- [USA](#) =(0x01, "USA")
- [UK](#) =(0x02, "UK")
- [French](#) =(0x03, "French")
- [German](#) =(0x04, "German")
- [Italian](#) =(0x05, "Italian")
- [Belgian](#) =(0x09, "Belgian")
- [Spanish](#) =(0x0A, "Spanish")
- [Portuguese](#) =(0x0B, "Portuguese")
- [Dutch](#) =(0x0C, "Dutch")
- [Swedish](#) =(0x0D, "Swedish")
- [Finnish](#) =(0x0E, "Finnish")
- [SwissGerman](#) =(0x0F, "SwissGerman")
- [SwissFrench](#) =(0x10, "SwissFrench")
- [Japanese](#) =(0x11, "Japanese")
- [Danish](#) =(0x12, "Danish")
- [Norwegian](#) =(0x13, "Norwegian")
- [Czech](#) =(0x15, "Czech")

---

### Explanation

This class is responsible for the different keyboard languages.

---

## Method

static [KeyboardLanguage](#) jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.toString ()

---

## Variable

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Belgian =(0x09, "Belgian")  
Belgian

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Czech =(0x15, "Czech")  
Czech

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Danish =(0x12, "Danish")  
Danish

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Dutch =(0x0C, "Dutch")  
Dutch

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Finnish =(0x0E, "Finnish")  
Finnish

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.French =(0x03, "French")  
French

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.German =(0x04, "German")  
German

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Italian =(0x05, "Italian")  
Italian

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Japanese =(0x11, "Japanese")  
Japanese

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Norwegian =(0x13, "Norwegian")  
Norwegian

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Portuguese =(0x0B, "Portuguese")  
Portuguese

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Spanish =(0x0A, "Spanish")  
Spanish

jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.Swedish =(0x0D, "Swedish")  
Swedish

**jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.SwissFrench** =(0x10, "SwissFrench")

SwissFrench

**jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.SwissGerman** =(0x0F, "SwissGerman")

SwissGerman

**jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.UK** =(0x02, "UK")

UK

**jp.co.opto.opnsdk.setting.enums.KeyboardLanguage.USA** =(0x01, "USA")

USA

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[KeyboardLanguage.java](#)

---

## **jp.co.opto.opnsdk.setting.enums.MemoryOutputMethod Enum Reference**

### **Public Method**

- String [toString](#) ()

### **Static Public Method**

- static [MemoryOutputMethod searchByValue](#) (int value)

### **Public Variable**

- [OnConnected](#) =(0, "Immediately outputs when connecting")
- [FunctionButtonOrCommand](#) =(1, "Outputted via function key press or commands")

---

### **Explanation**

This is the outputting method for memory data.

---

### **Method**

static [MemoryOutputMethod](#) jp.co.opto.opnsdk.setting.enums.MemoryOutputMethod.searchByValue  
(int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.MemoryOutputMethod.toString ()

---

### **Variable**

**jp.co.opto.opnsdk.setting.enums.MemoryOutputMethod.FunctionButtonOrCommand** =(1, "Outputted via  
function key press or commands")

Outputted via function key press or commands

**jp.co.opto.opnsdk.setting.enums.MemoryOutputMethod.OnConnected** =(0, "Immediately outputs when  
connecting")

Immediately outputs when connecting

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/MemoryOutputMethod.java](#)

---

## jp.co.opto.opnsdk.setting.enums.MSIPlessey\_CalculateCD Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [MSIPlessey\\_CalculateCD\\_searchByValue](#) (int value)

### Public Variable

- [CalculateCD\\_CD1\\_only\\_Mod10](#) =(1, "Calculate CD CD1 only (Mod10)")
- [CalculateCD\\_CDs\\_Mod10\\_Mod10](#) =(2, "Calculate CD (Mod10/Mod10)")
- [CalculateCD\\_CDs\\_Mod10\\_Mod11](#) =(3, "Calculate CD (Mod10/Mod11)")
- [CalculateCD\\_CDs\\_Mod11\\_Mod10](#) =(4, "Calculate CD (Mod11/Mod10)")
- [NotCalculateCD](#) =(0, "Do not calculate CD")

---

### Explanation

This class is responsible for calculating the MSI/Plessey CD.

---

### Method

static [MSIPlessey\\_CalculateCD](#) jp.co.opto.opnsdk.setting.enums.MSIPlessey\_CalculateCD.searchByValue (int value)[static]

String jp.co.opto.opnsdk.setting.enums.MSIPlessey\_CalculateCD.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.MSIPlessey\_CalculateCD.CalculateCD\_CD1\_only\_Mod10 =(1, "Calculate CD CD1 only (Mod10)")

Calculate CD CD1 only (Mod10)

jp.co.opto.opnsdk.setting.enums.MSIPlessey\_CalculateCD.CalculateCD\_CDs\_Mod10\_Mod10 =(2, "Calculate CD (Mod10/Mod10)")

Calculate CD (Mod10/Mod10)

jp.co.opto.opnsdk.setting.enums.MSIPlessey\_CalculateCD.CalculateCD\_CDs\_Mod10\_Mod11 =(3, "Calculate CD (Mod10/Mod11)")

Calculate CD (Mod10/Mod11)

jp.co.opto.opnsdk.setting.enums.MSIPlessey\_CalculateCD.CalculateCD\_CDs\_Mod11\_Mod10 =(4, "Calculate CD (Mod11/Mod10)")

Calculate CD (Mod11/Mod10)

jp.co.opto.opnsdk.setting.enums.MSIPlessey\_CalculateCD.NotCalculateCD =(0, "Do not calculate CD")

Do not calculate CD

---

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/MSIPlessey\\_CalculateCD.java](#)

---

#### jp.co.opto.opnsdk.setting.enums.MSIPlessey\_TransmitCD Enum Reference

##### Public Method

- String [toString](#) ()

##### Static Public Method

- static [MSIPlessey\\_TransmitCD\\_searchByValue](#) (int value)

##### Public Variable

- [TransmitCD\\_CD1](#) =(1, "Transmit CD CD1")
- [TransmitCD\\_CD1\\_CD2](#) =(2, "Transmit CD CD1 and CD2")
- [NotTransmitCD](#) =(0, "Do not transmit CD")

##### Explanation

This class is responsible for transmitting the MSI/Plessey CD.

---

##### Method

static [MSIPlessey\\_TransmitCD](#) jp.co.opto.opnsdk.setting.enums.MSIPlessey\_TransmitCD.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.MSIPlessey\_TransmitCD.toString ()

---

##### Variable

jp.co.opto.opnsdk.setting.enums.MSIPlessey\_TransmitCD.NotTransmitCD =(0, "Do not transmit CD")  
Do not transmit CD

jp.co.opto.opnsdk.setting.enums.MSIPlessey\_TransmitCD.TransmitCD\_CD1 =(1, "Transmit CD CD1")  
Transmit CD CD1

jp.co.opto.opnsdk.setting.enums.MSIPlessey\_TransmitCD.TransmitCD\_CD1\_CD2 =(2, "Transmit CD CD1 and CD2")  
Transmit CD CD1 and CD2

---

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/MSIPlessey\\_TransmitCD.java](#)

---

#### jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime Enum Reference

##### Public Method

- String [toString](#) ()

##### Static Public Method

- static [MultipleReadResetTime\\_searchByValue](#) (int value)

##### Public Variable

- [Infinity](#) =(0, "Infinity")
- [50msec](#) =(3, "50ms")
- [100msec](#) =(5, "100ms")



- [200msec](#) =(10, "200ms")
- [300msec](#) =(15, "300ms")
- [400msec](#) =(20, "400ms")
- [500msec](#) =(25, "500ms")
- [600msec](#) =(30, "600ms")

---

#### Explanation

This class is responsible for the multiple read reset timer.

---

#### Method

static [MultipleReadResetTime](#) jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.searchByValue  
(int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.toString ()

---

#### Variable

jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.\_100msec =(5, "100ms")  
100ms

jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.\_200msec =(10, "200ms")  
200ms

jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.\_300msec =(15, "300ms")  
300ms

jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.\_400msec =(20, "400ms")  
400ms

jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.\_500msec =(25, "500ms")  
500ms

jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.\_50msec =(3, "50ms")  
50ms

jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.\_600msec =(30, "600ms")  
600ms

jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime.Infinity =(0, "Infinity")  
Infinity

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[MultipleReadResetTime.java](#)

## jp.co.opto.opnsdk.setting.enums.NumPadMode Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [NumPadMode searchByValue](#) (int value)

### Public Variable

- [Auto](#) =(0x00, "Behaves according to the AUTO host's NumLock key state)
- [OFF](#) =(0x01, "Forcefully changes into a state where the NumLock key is not pressed")

### Explanation

This class is responsible for the numpad mode.

---

### Method

static [NumPadMode](#) jp.co.opto.opnsdk.setting.enums.NumPadMode.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.NumPadMode.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.NumPadMode.Auto =(0x00, "Behaves according to the AUTO host's NumLock key state")

Behaves according to the AUTO host's NumLock key state

jp.co.opto.opnsdk.setting.enums.NumPadMode.OFF =(0x01, "Forcefully changes into a state where the NumLock key is not pressed")

Forcefully changes into a state where the NumLock key is not pressed

---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[NumPadMode.java](#)

---

## jp.co.opto.opnsdk.setting.enums.NW7\_CalculateCD Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [NW7\\_CalculateCD searchByValue](#) (int value)

### Public Variable

- [CalculateCD\\_Mod10\\_WeightHalf](#) =(1, "Calculate CD Mod10/W1,2 spec1")
- [CalculateCD\\_Mod16](#) =(2, "Calculate CD Mod16")
- [CalculateCD\\_7check](#) =(3, "Calculate CD. 7 checks")
- [CalculateCD\\_Mod11](#) =(4, "Calculate CD Mod11")
- [NotCalculateCD](#) =(0, "Do not calculate CD")

---

### Explanation

This class is responsible for calculating the NW-7 CD.

## Method

static [NW7\\_CalculateCD](#) jp.co.opto.opnsdk.setting.enums.NW7\_CalculateCD.searchByValue (int value)[static]

String jp.co.opto.opnsdk.setting.enums.NW7\_CalculateCD.toString ()

## Variable

jp.co.opto.opnsdk.setting.enums.NW7\_CalculateCD.CalculateCD\_7check =(3, "Calculate CD. 7 checks")  
Calculate CD. 7 checks

jp.co.opto.opnsdk.setting.enums.NW7\_CalculateCD.CalculateCD\_Mod10\_WeightHalf =(1, "Calculate CD Mod10/W1,2 spec1")  
Calculate CD Mod10/W1, 2 spec1

jp.co.opto.opnsdk.setting.enums.NW7\_CalculateCD.CalculateCD\_Mod11 =(4, "Calculate CD Mod11")  
Calculate CD Mod11

jp.co.opto.opnsdk.setting.enums.NW7\_CalculateCD.CalculateCD\_Mod16 =(2, "Calculate CD Mod16")  
Calculate CD Mod16

jp.co.opto.opnsdk.setting.enums.NW7\_CalculateCD.NotCalculateCD =(0, "Do not calculate CD")  
Do not calculate CD

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[NW7\\_CalculateCD.java](#)

## jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP Enum Reference

### Public Method

- String [toString](#) ()

### Static Public Method

- static [NW7\\_TransmitSTSP\\_searchByValue](#) (int value)

### Public Variable

- [Transmit\\_ABCD\\_TN\\_ast\\_E](#) =(1, "Transmit ABCD/TN\*E")
- [Transmit\\_abcd\\_tn\\_ast\\_e](#) =(2, "Transmit abcd/tn\*e")
- [Transmit\\_ABCD](#) =(3, "Transmit ABCD")
- [Transmit\\_abcd](#) =(4, "Transmit abcd")
- [Transmit\\_DC1DC2DC3DC4](#) =(5, "Transmit DC1DC2DC3DC4")
- [NotTransmit](#) =(0, "Do not transmit")

## Explanation

This class is responsible for transmitting the NW-7 start/stop.

## Method

static [NW7\\_TransmitSTSP](#) jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP.searchByValue (int value)[static]

String jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP.toString ()

---

## Variable

jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP.NotTransmit =(0, "Do not trasnmit")

Do not transmit

jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP.Transmit\_ABCD =(3, "Transmit ABCD")

Transmit ABCD

jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP.Transmit\_abcd =(4, "Transmit abcd")

Transmit abcd

jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP.Transmit\_ABCD\_TN\_ast\_E =(1, "Transmit ABCD/TN\*E")

Transmit ABCD/TN\*E

jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP.Transmit\_abcd\_tn\_ast\_e =(2, "Transmit abcd/tn\*e")

Transmit abcd/tn\*e

jp.co.opto.opnsdk.setting.enums.NW7\_TransmitSTSP.Transmit\_DC1DC2DC3DC4 =(5, "Transmit DC1DC2DC3DC4")

Transmit DC1DC2DC3DC4

---

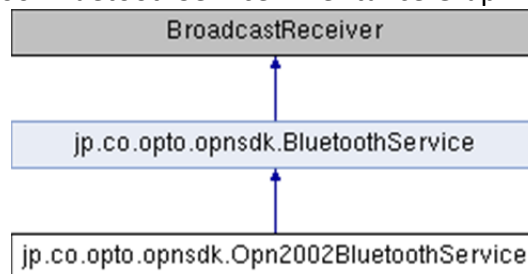
The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/NW7\\_TransmitSTSP.java](#)

---

## Class jp.co.opto.opnsdk.Opn2002BluetoothService

jp.co.opto.opnsdk.Opn2002BluetoothService Inheritance Graph



### Public Method

- void [initializeSettingAll](#) ([VoidCallback](#) callback)
- void [initializeSettingBluetooth](#) ([VoidCallback](#) callback)
- void [getFirmwareVersion](#) ([GetFirmwareVersionCallback](#) callback)
- void [readBarcode](#) ([VoidCallback](#) callback)
- void [vibratorOn](#)([VoidCallback](#) callback)
- void [beepBuzzer](#) ([VoidCallback](#) callback)
- void [getDateAndTime](#) ([GetDateAndTimeCallback](#) callback)
- void [getBatteryLevel](#) ([GetBatteryLevelCallback](#) callback)
- void [saveSetting](#) ([VoidCallback](#) callback)
- void [getStoredBarcodeAll](#) ([VoidCallback](#) callback)
- void [eraseStoredBarcodeAll](#) ([VoidCallback](#) callback)
- void [setPIN](#) (String str, [VoidCallback](#) callback)
- void [setBDAddress](#) (String str, [VoidCallback](#) callback)
- void [setDeviceName](#) (String str, [VoidCallback](#) callback)
- void [setDateAndTime](#) (String dateStr, [SetDateAndTimeCallback](#) callback)
- void [getSetting](#) ([GetSettingCallback](#) callback)
- void [enableAckNak](#) ()
- void [enableAckNak](#) ([EnableAckNakCallback](#) callback)
- void [write](#) (String out)
- void [write](#) (String out, [Opn2002Callback<?>](#) callback)

### Static Public Method

- static [Opn2002BluetoothService](#) [getInstance](#) ()
- static [Opn2002BluetoothService](#) [getInstance](#) (Context context)
- static byte[] [toPrimitive](#) (Byte[] array)

### Static Public Variable

- static final byte[] [EMPTY\\_BYTE\\_ARRAY](#) = new byte[0]

### Protected Method

- void [postObserver](#) (final int postType, final Object...obj)

### Additional Inherited Members

---

### Explanation

This is the OPN2002i Bluetooth Service class designed to execute SPP connection with the OPN2002i/OPN3002i, and is provided as a basic command method for these devices. Because these are executed via [GoF] Singleton Pattern, only one instance is created. Please be sure to use [Opn2002BluetoothService#getInstance\(\)](#) when returning instances. Basic transmit processing is handled the same as "[BluetoothService](#)".

---

### Method

**void [jp.co.opto.opnsdk.Opn2002BluetoothService.enableAckNak](#) ()**

Sets the ACK/NAK control settings to "enable". Processing for this method is conducted asynchronously.

**void [jp.co.opto.opnsdk.Opn2002BluetoothService.enableAckNak](#) ([EnableAckNakCallback](#) callback)**

Sets the ACK/NAK control settings to "enable". Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void [jp.co.opto.opnsdk.Opn2002BluetoothService.eraseStoredBarcodeAll](#) ([VoidCallback](#) *callback*)**

This clears the data collector's memory. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void [jp.co.opto.opnsdk.Opn2002BluetoothService.getBatteryLevel](#) ([GetBatteryLevelCallback](#) *callback*)**

This returns the data collector's battery voltage. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void [jp.co.opto.opnsdk.Opn2002BluetoothService.getDateAndTime](#) ([GetDateAndTimeCallback](#) *callback*)**

This returns the current date and time settings for the device's real-time clock. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void [jp.co.opto.opnsdk.Opn2002BluetoothService.getFirmwareVersion](#) ([GetFirmwareVersionCallback](#) *callback*)**

This returns the firmware version of the device. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**static [Opn2002BluetoothService](#) [jp.co.opto.opnsdk.Opn2002BluetoothService.getInstance](#) ()[static]**

This returns the single existing instance. If [getInstance\(Context context\)](#) is not being executed, "Null" is returned.

**Return Value:**

Instance

**static [Opn2002BluetoothService](#) [jp.co.opto.opnsdk.Opn2002BluetoothService.getInstance](#) (Context *context*)[static]**

This returns the single existing instance.

**Argument:**

<i>context</i>	context
----------------	---------

**Return Value:**

Instance

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.getSetting` ([GetSettingCallback](#) *callback*)**

This returns all data collector settings. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.getStoredBarcodeAll` ([VoidCallback](#) *callback*)**

This outputs the device's memory. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.initializeSettingAll` ([VoidCallback](#) *callback*)**

This initializes the data collector settings. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.initializeSettingBluetooth` ([VoidCallback](#) *callback*)**

This initializes the data collector's Bluetooth settings. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.postObserver` (final int *postType*, final Object... *obj*)[protected]**

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.readBarcode` ([VoidCallback](#) *callback*)**

This begins the barcode scanning functions of the device. Processing for this method is conducted asynchronously. The callback is executed when processing is completed. The laser sets to the "ON" state for a determined period of time.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.vibratorOn` ([VoidCallback](#) *callback*)**

This activates the vibration function for the device. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.beepBuzzer` ([VoidCallback](#) *callback*)**

This activates the data collector buzzer. Processing for this method is conducted asynchronously. The callback is executed when processing is completed. The buzzer begins to beep.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.saveSetting` ([VoidCallback](#) *callback*)**

The settings saved in the data collector's volatile memory are saved as non-volatile memory. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>callback</i>	The callback executed once processing is completed.
-----------------	---

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.setBDAddress` (String *str*, [VoidCallback](#) *callback*)**

This sets the BD address for the connecting device to the data collector. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>str</i>	This is the partner device BD address to be set.
<i>callback</i>	The callback executed once processing is completed.

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.setDateAndTime` (String *dateStr*, [SetDateAndTimeCallback](#) *callback*)**

This sets the current date and time settings for the device's real-time clock. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>dateStr</i>	The newly set data and time character string (yyyyMMddHHmmss).
<i>callback</i>	The callback executed once processing is completed.

**Exception:**

<i>IllegalArgumentException</i>	This is when the <i>dateStr</i> is "Null".
---------------------------------	--

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.setDeviceName` (String *str*, [VoidCallback](#) *callback*)**

This sets the data collector's device name. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>str</i>	The new device name to be set.
<i>callback</i>	The callback executed once processing is completed.

**void `jp.co.opto.opnsdk.Opn2002BluetoothService.setPIN` (String *str*, [VoidCallback](#) *callback*)**

This sets the PIN code for the data collector. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>str</i>	The new PIN code to be set.
<i>callback</i>	The callback executed once processing is completed.



**static byte [] jp.co.opto.opnsdk.Opn2002BluetoothService.toPrimitive (Byte[] array)[static]**

This changes the byte array to a more “*primitive*” version.

**Argument:**

<i>array</i>	
--------------	--

**Return Value:**

**void jp.co.opto.opnsdk.Opn2002BluetoothService.write (String out)**

This transmits commands to the data collector.

**Argument:**

<i>out</i>	The commands to be transmitted.
------------	---------------------------------

**Exception:**

<i>IllegalArgumentException</i> <i>Exception</i>	When "out" is "NULLNull".
---	---------------------------

**reference:**

[BluetoothService::write\(String\)](#)

**void jp.co.opto.opnsdk.Opn2002BluetoothService.write (String out, Opn2002Callback<?> callback)**

This transmits commands to the data collector. Processing for this method is conducted asynchronously. The callback is executed when processing is completed.

**Argument:**

<i>out</i>	The commands to be transmitted.
<i>callback</i>	The callback executed once processing is completed.

---

## Variable

**final byte [] jp.co.opto.opnsdk.Opn2002BluetoothService.EMPTY\_BYTE\_ARRAY = new byte[0][static]**

---

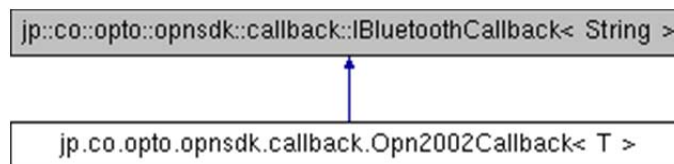
The explanation for this class was created from the file below:

- src/jp/co/opto/opnsdk/[Opn2002BluetoothService.java](#)

---

**Class jp.co.opto.opnsdk.callback.Opn2002Callback< T >**

jp.co.opto.opnsdk.callback.Opn2002Callback< T > Inheritance Graph



## Public Method

- boolean [getIsError](#) ()
- void [execute](#) (String str)
- boolean [canExecute](#) (String str)

## Protected Method

- void [setIsError](#) (boolean isError)

- abstract T [createParameter](#) (String str)
- abstract void [callbackExecute](#) (T parameter)
- HashMap<String, String> [getSettings](#) (String str)

#### Protected Variable

- boolean [mIsError](#) = false

#### Explanation

This asynchronous class executes the callback.

#### Argument:

<T>	callbackExecute's argument type.
-----	----------------------------------

---

#### Method

**abstract void jp.co.opto.opnsdk.callback.Opn2002Callback< T >.callbackExecute (T *parameter*)[protected], [pure virtual]**

This executes the callback. Implementation is handled via the sub-class.

**boolean jp.co.opto.opnsdk.callback.Opn2002Callback< T >.canExecute (String str)**

**abstract T jp.co.opto.opnsdk.callback.Opn2002Callback< T >.createParameter (String str)[protected], [pure virtual]**

This creates the parameters passed when the callback is executed. Implementation will be handled via the sub-class.

#### Argument:

<i>str</i>	The received character string.
------------	--------------------------------

#### Return Value:

Parameters.

**void jp.co.opto.opnsdk.callback.Opn2002Callback< T >.execute (String str)**

**boolean jp.co.opto.opnsdk.callback.Opn2002Callback< T >.getIsError ()**

The error state is returned in conjunction with the asynchronous processing result.

#### Return Value:

If there is an error, "true". Otherwise, "false".

**HashMap<String, String> jp.co.opto.opnsdk.callback.Opn2002Callback< T >.getSettings (String str)[protected]**

This returns the setting character string transmitted from the data collector and returns it as "HashMap". If the string is written along the lines of "KEY1=VAL1[CR]KEY2=VAL2[CR] · · · KEYn=VALn", each will be split into either "[CR]" or "=", and then the HashMap object is created.

#### Argument:

<i>str</i>	The received data (setting character string).
------------	---

**Return Value:**

Setting.

**void jp.co.opto.opnsdk.callback.Opn2002Callback< T >.setIsError (boolean *isError*)[protected]**

The error state is returned in conjunction with the asynchronous processing result.

**Argument:**

<i>isError</i>	If there is an error, "true". Otherwise, "false".
----------------	---

**Variable**

**boolean jp.co.opto.opnsdk.callback.Opn2002Callback< T >.mIsError = false[protected]**

The explanation for this class was created from the file below:

- src/jp/co/opto/opnsdk/callback/[Opn2002Callback.java](#)

---

**Class jp.co.opto.opnsdk.setting.PrefixSettings**

**Public Method**

- [PrefixSettings](#) (byte[] settingData)
- int[] [getmUPCA](#) ()
- int[] [getmUPCA\\_Addon](#) ()
- int[] [getmUPCE](#) ()
- int[] [getmUPCE\\_Addon](#) ()
- int[] [getmEAN13](#) ()
- int[] [getmEAN13\\_Addon](#) ()
- int[] [getmEAN8](#) ()
- int[] [getmEAN8\\_Addon](#) ()
- int[] [getCode39](#) ()
- int[] [getmTriOptic](#) ()
- int[] [getmNW7](#) ()
- int[] [getmD2of5](#) ()
- int[] [getmI2of5](#) ()
- int[] [getmSCode](#) ()
- int[] [getmMatrix2of5](#) ()
- int[] [getCode93](#) ()
- int[] [getCode128](#) ()
- int[] [getmMSIPlessey](#) ()
- int[] [getmIATA](#) ()
- int[] [getmUKPlessey](#) ()
- int[] [getmTelepen](#) ()
- int[] [getmRSS](#) ()
- int[] [getmPDF417](#) ()
- int[] [getmMicroPDF417](#) ()
- int[] [getCode11](#) ()
- int[] [getCode3of5](#) ()
- int[] [getmEAN128](#) ()
- int[] [getIntelligentMail](#) ()
- int[] [getPostnet](#) ()

- `int[] getJapanesePostal ()`
- `int[] getCodablockF ()`
- `int[] getDataMatrixECC200 ECC000 140 ()`
- `int[] getAztecCode AztecRunes ()`
- `int[] getChineseSensibleCode ()`
- `int[] getQRcode MicroQR ()`
- `int[] getMaxiCode ()`
- `int[] getCompositeOnGS1Databar UPCEAN ()`
- `int[] getmCommon ()`
- `int[] getmAllCode ()`

---

### Explanation

This class activates settings related to the prefix.

---

### Constructor and Destructor

**`jp.co.opto.opnsdk.setting.PrefixSettings.PrefixSettings (byte[] settingData)`**

This is the constructor.

**Argument:**

<i>settingData</i>	Setting data (format when transmitting).
--------------------	--

---

### Method

**`int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmAllCode ()`**

This returns the prefix for all codes.

**`int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmCode11 ()`**

This returns the Code 11 prefix.

**`int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmCode128 ()`**

This returns the Code 128 prefix.

**`int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmCode39 ()`**

This returns the Code 39 prefix.

**`int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmCode3of5 ()`**

This returns the Code 3 of 5 prefix.

**`int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmCode93 ()`**

This returns the Code 93 prefix.

**`int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmCommon ()`**

This returns the common suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmD2of5 ()**

This returns the D 2 of 5 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmEAN128 ()**

This returns the EAN-128 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmEAN13 ()**

This returns the EAN-13 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmEAN13\_Addon ()**

This returns the EAN 13 Add On prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmEAN8 ()**

This returns the EAN-8 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmEAN8\_Addon ()**

This returns the EAN 8 Add On prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmI2of5 ()**

This returns the I 2 of 5 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmIATA ()**

This returns the IATA prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmMatrix2of5 ()**

This returns the Matrix 2 of 5 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmMicroPDF417 ()**

This returns the Micro PDF 417 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmMSIPlessey ()**

This returns the MSI/Plessey prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmNW7 ()**

This returns the NW7 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmPDF417 ()**

This returns the PDF 417 prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmRSS ()**

This returns the mRSS prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getmSCode ()**

This returns the SCode prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMTelepen ()**

This returns the Telepen prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMTriOptic ()**

This returns the TriOptic prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMUKPlessey ()**

This returns the UK/Plessey prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMUPCA ()**

This returns the UPC-A prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMUPCA\_Addon ()**

This returns the UPC-A Add On prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMUPCE ()**

This returns the UPC-E prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMUPCE\_Addon ()**

This returns the UPC-E Add On prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getIntelligentMail ()**

This returns the IntelligentMail prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getPostnet ()**

This returns the Postnet prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getJapanesePostal ()**

This returns the Japanese Postal prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getCodablockF ()**

This returns the CodablockF prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getDataMatrixECC200\_ECC000\_140 ()**

This returns the prefix for all codes.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getAztecCode\_AztecRunes ()**

This returns the Aztec Code/Aztec Runes prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getChineseSensibleCode ()**

This returns the Chinese Sensible Code prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getQRcode\_MicroQR ()**

This returns the QR/Micro QR prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMaxiCode ()**

This returns the Maxi Code prefix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getCompositeOnGS1Databar\_UPCEAN ()**

This returns the Composite (on GS1 Databar, UPC/EAN) prefix

---

The explanation for this class was created from the file below:

- src/jp/co/opto/opnsdk/setting/[PrefixSettings.java](#)

---

#### **jp.co.opto.opnsdk.setting.enums.ReadableTime Enum Reference**

##### **Public Method**

- String [toString](#) ()

##### **Static Public Method**

- static [ReadableTime searchByValue](#) (int value)

##### **Public Variable**

- [Infinity](#) =(0, "Infinity")
- [0sec](#) =(-1, "0 sec")
- [1sec](#) =(50, "1 sec")
- [2sec](#) =(100, "2 sec")
- [3sec](#) =(150, "3 sec")
- [4sec](#) =(200, "4 sec")
- [5sec](#) =(250, "5 sec")
- [6sec](#) =(300, "6 sec")
- [7sec](#) =(350, "7 sec")
- [8sec](#) =(400, "8 sec")
- [9sec](#) =(450, "9 sec")

---

#### **Explanation**

This class is responsible for the scanning time duration.

---

#### **Method**

static [ReadableTime](#) jp.co.opto.opnsdk.setting.enums.ReadableTime.searchByValue (int *value*)[static]

String jp.co.opto.opnsdk.setting.enums.ReadableTime.toString ()

---

#### **Variable**

jp.co.opto.opnsdk.setting.enums.ReadableTime.\_0sec =(-1, "0 sec")

0 sec

jp.co.opto.opnsdk.setting.enums.ReadableTime.\_1sec =(50, "1 sec")

1 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime._2sec =(100, "2 sec")`

2 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime._3sec =(150, "3 sec")`

3 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime._4sec =(200, "4 sec")`

4 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime._5sec =(250, "5 sec")`

5 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime._6sec =(300, "6 sec")`

6 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime._7sec =(350, "7 sec")`

7 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime._8sec =(400, "8 sec")`

8 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime._9sec =(450, "9 sec")`

9 sec

`jp.co.opto.opnsdk.setting.enums.ReadableTime.Infinity =(0, "Infinity")`

Infinity

---

The documentation for this enum was generated from the following file:

- `src/jp/co/opto/opnsdk/setting/enums/ReadableTime.java`

---

## **jp.co.opto.opnsdk.setting.enums.ReadMode Enum Reference**

### **Public Method**

- String [toString\(\)](#)

### **Static Public Method**

- static [ReadMode searchByValue](#) (int value)

### **Public Variable**

- [SingleRead](#) =(1, "Single read")
- [MultipleRead](#) =(2, "Multiple read")
- [ContinuousRead](#) =(3, "Continuous Read")

---

### **Explanation**

This class is responsible for the available scanning modes within this SDK.

---



## Method

static [ReadMode](#) `jp.co.opto.opnsdk.setting.enums.ReadMode.searchByValue (int value)[static]`

`String jp.co.opto.opnsdk.setting.enums.ReadMode.toString ()`

---

## Variable

`jp.co.opto.opnsdk.setting.enums.ReadMode.ContinuousRead =(3, "連続読み")`

Continuous Read

`jp.co.opto.opnsdk.setting.enums.ReadMode.MultipleRead =(2, "Multiple read")`

Multiple Read

`jp.co.opto.opnsdk.setting.enums.ReadMode.SingleRead =(1, "Single read")`

Single Read

---

The documentation for this enum was generated from the following file:

- `src/jp/co/opto/opnsdk/setting/enums/ReadMode.java`

---

## `jp.co.opto.opnsdk.setting.enums.RedundantReading` Enum Reference

### Public Method

- `String toString ()`

### Static Public Method

- static `RedundantReading searchByValue \(int value\)`

### Public Variable

- `NoRedundancy =(0, "1 scan 0 checks")`
  - `2TimesRedundancy =(1, "2 scans 1 checks")`
  - `3TimesRedundancy =(2, "3 scans 2 checks")`
  - `4TimesRedundancy =(3, "4 scans 3 checks")`
  - `5TimesRedundancy =(4, "5 scans 4 checks")`
  - `6TimesRedundancy =(5, "6 scans 5 checks")`
  - `7TimesRedundancy =(6, "7 scans 6 checks")`
  - `8TimesRedundancy =(7, "8 scans 7 checks")`
  - `9TimesRedundancy =(8, "9 scans 8 checks")`
  - `10TimesRedundancy =(9, "10 scans 9 checks")`
  - `11TimesRedundancy =(10, "11 scans 10 checks")`
  - `12TimesRedundancy =(11, "12 scans 11 checks")`
  - `13TimesRedundancy =(12, "13 scans 12 checks")`
  - `14TimesRedundancy =(13, "14 scans 13 checks")`
  - `15TimesRedundancy =(14, "15 scans 14 checks")`
  - `16TimesRedundancy =(15, "16 scans 15 checks")`
-

## Explanation

This class is responsible for the number of available checks.

## Method

static [RedundantReading](#) jp.co.opto.opnsdk.setting.enums.RedundantReading.searchByValue (int value)[static]

String jp.co.opto.opnsdk.setting.enums.RedundantReading.toString ()

---

## Variable

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_10TimesRedundancy =(9, "10 scans 9 C=checks")  
10 scans 9 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_11TimesRedundancy =(10, "11 scans 10 checks ")  
11 scans 10 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_12TimesRedundancy =(11, "12 scans 11 checks")  
12 scans 11 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_13TimesRedundancy =(12, "13 scans 12 checks")  
13 scans 12 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_14TimesRedundancy =(13, "14 scans 13 checks")  
14 scans 13 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_15TimesRedundancy =(14, "15 scans 14 checks")  
15 scans 14 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_16TimesRedundancy =(15, "16 scans 15 checks")  
16 scans 15 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_2TimesRedundancy =(1, "2 scans 1 check")  
2 scans 1 check

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_3TimesRedundancy =(2, "3 scans 2 checks")  
3 scans 2 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_4TimesRedundancy =(3, "4 scans 3 checks")  
4 scans 3 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_5TimesRedundancy =(4, "5 scans 4 checks")  
5 scans 4 checks

jp.co.opto.opnsdk.setting.enums.RedundantReading.\_6TimesRedundancy =(5, "6 scans 5 checks")  
6 scans 5 checks

**jp.co.opto.opnsdk.setting.enums.RedundantReading.\_7TimesRedundancy**=(6, "7 scans 6 checks")  
7 scans 6 checks

**jp.co.opto.opnsdk.setting.enums.RedundantReading.\_8TimesRedundancy**=(7, "8 scans 7 checks")  
8 scans 7 checks

**jp.co.opto.opnsdk.setting.enums.RedundantReading.\_9TimesRedundancy**=(8, "9 scans 8 checks")  
9 scans 8 checks

**jp.co.opto.opnsdk.setting.enums.RedundantReading.NoRedundancy**=(0, "1 scan 0 checks")  
1 scans 0 checks

---

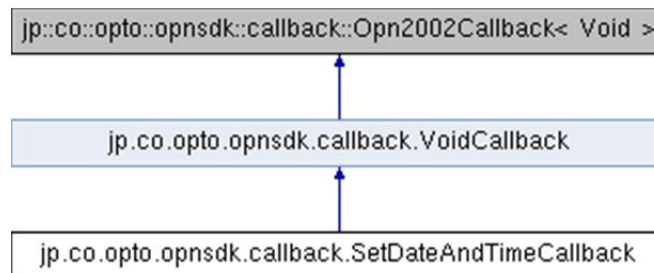
The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/RedundantReading.java](#)

---

**Class jp.co.opto.opnsdk.callback.SetDateAndTimeCallback**

jp.co.opto.opnsdk.callback.SetDateAndTimeCallback Inheritance Graph



#### Public Method

- boolean [canExecute](#) (String str)

#### Additional Inherited Members

---

#### Explanation

This abstract class is responsible for setting the date and time callback.

---

#### Method

boolean **jp.co.opto.opnsdk.callback.SetDateAndTimeCallback.canExecute** (String str)

---

The explanation for this class was created from the file below:

- [src/jp/co/opto/opnsdk/callback/SetDateAndTimeCallback.java](#)

---

**Class jp.co.opto.opnsdk.setting.Settings**

#### Public Method

- [Settings](#) (byte[] settingData)
- [DecoderSettings](#) getDecoderSettings ()
- [BluetoothSettings](#) getBluetoothSettings ()
- [PrefixSettings](#) getPrefixSettings ()
- [SuffixSettings](#) getSuffixSettings ()
- String [toString](#) ()

---

### Explanation

This class displays the device's setting value. The format will differ depending on whether the data being set is "Internal Storage" or "Transmitting".

#### \* Internal Storage

Separated by two-byte units (this unit is called a "block").

#### \* Transmitting

This is separated by four-byte units. When the data collector is transmitting, internal storage data is changed from two to four bytes.

#### \* Example:

Internal Storage: 0x07 0x4A

→ character 0 character 7 character 4 character A

→ ASCII Code 48 ASCII Code 55 ASCII Code 52 ASCII Code 65

Transmitting: 0x30 0x37 0x34 0x41

---

### Constructor and Destructor

#### **jp.co.opto.opnsdk.setting.Settings.Settings (byte[] *settingData*)**

This is the constructor.

Please see the "[Settings](#)" Explanation for details regarding the settingData format.

#### **Argument:**

<i>settingData</i>	Setting data (format when transmitting).
--------------------	--

---

### Method

#### **[BluetoothSettings](#) jp.co.opto.opnsdk.setting.Settings.getBluetoothSettings ()**

This returns settings related to Bluetooth transmitting.

#### **[DecoderSettings](#) jp.co.opto.opnsdk.setting.Settings.getDecoderSettings ()**

This returns settings related to the decoder.

#### **[PrefixSettings](#) jp.co.opto.opnsdk.setting.Settings.getPrefixSettings ()**

This returns settings related to the prefix.

[SuffixSettings](#) `jp.co.opto.opnsdk.setting.Settings.getSuffixSettings ()`

This returns settings related to the suffix.

`String jp.co.opto.opnsdk.setting.Settings.toString ()`

Contents of the setting are returned as sixteen-decimal strings. These contents are expressed in the internal storage formatting. For further details, please refer to the explanation for "[Settings](#)".

**Reference:**

`java.lang.Object: toString()`

---

The explanation for this class was created from the file below:

- `src/jp/co/opto/opnsdk/setting/Settings.java`

---

**`jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime` Enum Reference**

**Public Method**

- `String toString ()`

**Static Public Method**

- `static SlaveConnectionWaitTime searchByValue (int value)`

**Public Variable**

- `30sec =(1500, "30 sec")`
- `1min =(3000, "1 min")`
- `2min =(6000, "2 min")`
- `3min =(9000, "3 min")`
- `4min =(12000, "4min")`

---

**Explanation**

This class is responsible for the slave connection wait time.

---

**Method**

static [SlaveConnectionWaitTime](#)

`jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime.searchByValue (int value)[static]`

`String jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime.toString ()`

---

**Variable**

`jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime._1min =(3000, "1 min")`

1 min

`jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime._2min =(6000, "2 min")`

2 min

`jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime._30sec =(1500, "30 sec")`

30 sec

`jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime._3min =(9000, "3 min")`

3 min

`jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime._4min =(12000, "4 min")`

4 min

---

The documentation for this enum was generated from the following file:

- `src/jp/co/opto/opnsdk/setting/enums/SlaveConnectionWaitTime.java`

---

## Class `jp.co.opto.opnsdk.setting.SuffixSettings`

### Public Method

- [SuffixSettings](#) (byte[] settingData)
- [getmUPCA](#) ()
- [getmUPCA\\_Addon](#) ()
- [getmUPCE](#) ()
- [getmUPCE\\_Addon](#) ()
- [getmEAN13](#) ()
- [getmEAN13\\_Addon](#) ()
- [getmEAN8](#) ()
- [getmEAN8\\_Addon](#) ()
- [getmCode39](#) ()
- [getmTriOptic](#) ()
- [getmNW7](#) ()
- [getmD2of5](#) ()
- [getmI2of5](#) ()
- [getmSCode](#) ()
- [getmMatrix2of5](#) ()
- [getmCode93](#) ()
- [getmCode128](#) ()
- [getmMSIPlessey](#) ()
- [getmIATA](#) ()
- [getmUKPlessey](#) ()
- [getmTelepen](#) ()
- [getmRSS](#) ()
- [getmPDF417](#) ()
- [getmMicroPDF417](#) ()
- [getmCode11](#) ()
- [getmCode3of5](#) ()
- [getmEAN128](#) ()
- [getIntelligentMail](#) ()
- [getPostnet](#) ()
- [getJapanesePostal](#) ()
- [getCodablockF](#) ()

- [int\[\] `getDataMatrixECC200 ECC000 140` \(\)](#)
- [int\[\] `getAztecCode\_AztecRunes` \(\)](#)
- [int\[\] `getChineseSensibleCode` \(\)](#)
- [int\[\] `getQRcode\_MicroQR` \(\)](#)
- [int\[\] `getMaxiCode` \(\)](#)
- [int\[\] `getCompositeOnGS1Databar\_UPCEAN` \(\)](#)
- [int\[\] `getmCommon` \(\)](#)
- [int\[\] `getmAllCode` \(\)](#)

---

### Explanation

This class is responsible for activating settings related to the suffix.

### Constructor and Destructor

**jp.co.opto.opnsdk.setting.SuffixSettings.SuffixSettings (byte[] *settingData*)**

This is the constructor.

#### Argument:

<i>settingData</i>	Setting data (format when transmitting).
--------------------	--

---

### Method

**int [] `jp.co.opto.opnsdk.setting.SuffixSettings.getmAllCode` ()**

This returns the suffix for all codes.

**int [] `jp.co.opto.opnsdk.setting.SuffixSettings.getmCode11` ()**

This returns the Code 11 suffix.

**int [] `jp.co.opto.opnsdk.setting.SuffixSettings.getmCode128` ()**

This returns the Code 128 suffix.

**int [] `jp.co.opto.opnsdk.setting.SuffixSettings.getmCode39` ()**

This returns the Code 39 suffix.

**int [] `jp.co.opto.opnsdk.setting.SuffixSettings.getmCode3of5` ()**

This returns the Code 3 of 5 suffix.

**int [] `jp.co.opto.opnsdk.setting.SuffixSettings.getmCode93` ()**

This returns the Code 93 suffix.

**int [] `jp.co.opto.opnsdk.setting.SuffixSettings.getmCommon` ()**

This returns the common suffix.

**int [] `jp.co.opto.opnsdk.setting.SuffixSettings.getmD2of5` ()**

This returns the D 2 of 5 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmEAN128 ()**

This returns the EAN-128 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmEAN13 ()**

This returns the EAN-13 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmEAN13\_Addon ()**

This returns the EAN 13 Add On suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmEAN8 ()**

This returns the EAN-8 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmEAN8\_Addon ()**

This returns the EAN 8 Add On suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmI2of5 ()**

This returns the I 2 of 5 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmIATA ()**

This returns the IATA suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmMatrix2of5 ()**

This returns the Matrix 2 of 5 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmMicroPDF417 ()**

This returns the Micro PDF 417 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmMSIPlessey ()**

This returns the MSI/Plessey suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmNW7 ()**

This returns the NW7 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmPDF417 ()**

This returns the PDF 417 suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmRSS ()**

This returns the mRSS suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmSCode ()**

This returns the SCode suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmTelepen ()**

This returns the Telepen suffix.



**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmTriOptic ()**

This returns the TriOptic suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmUKPlessey ()**

This returns the UK/Plessey suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmUPCA ()**

This returns the UPC-A suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmUPCA\_Addon ()**

This returns the UPC-A Add On suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmUPCE ()**

This returns the UPC-E suffix.

**int [] jp.co.opto.opnsdk.setting.SuffixSettings.getmUPCE\_Addon ()**

This returns the UPC-E Add On suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getIntelligentMail ()**

This returns the IntelligentMail suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getPostnet ()**

This returns the Postnet suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getJapanesePostal ()**

This returns the Japanese Postal suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getCodablockF ()**

This returns the CodablockF suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getDataMatrixECC200\_ECC000\_140 ()**

This returns the Data Matrix (ECC200, ECC000-140) suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getAztecCode\_AztecRunes ()**

This returns the Aztec Code/Aztec Runes suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getChineseSensibleCode ()**

This returns the Chinese Sensible Code suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getQRcode\_MicroQR ()**

This returns the QR/Micro QR suffix.

**int [] jp.co.opto.opnsdk.setting.PrefixSettings.getMaxiCode ()**

This returns the Maxi Code suffix.

`int [] jp.co.opto.opnsdk.setting.PrefixSettings.getCompositeOnGS1Databar_UPCEAN ()`

This returns the Composite (on GS1 Databar, UPC/EAN) suffix.

---

The explanation for this class was created from the file below:

- `src/jp/co/opto/opnsdk/setting/SuffixSettings.java`

---

## `jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection` Enum Reference

### Public Method

- String `toString ()`

### Static Public Method

- static
- `TriggerButtonPushTimeForConnection searchByValue (int value)`

### Public Variable

- `Disable` =(0, "Disable trigger connect")
- `1sec` =(50, "1 sec")
- `2sec` =(100, "2 sec")
- `3sec` =(150, "3 sec")
- `4sec` =(200, "4 sec")
- `5sec` =(250, "5 sec")
- `6sec` =(300, "6 sec")
- `7sec` =(350, "7 sec")
- `8sec` =(400, "8 sec")
- `9sec` =(450, "9 sec")

---

### Explanation

This class is responsible for the trigger button push time duration when connecting.

---

### Method

static `TriggerButtonPushTimeForConnection`

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.searchByValue (int value)[static]`

String `jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.toString ()`

---

### Variable

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection._1sec` =(50, "1 sec")

1 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection._2sec` =(100, "2 sec")

2 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection._3sec` =(150, "3 sec")

3 sec

**jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.\_4sec** =(200, "4 sec")

4 sec

**jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.\_5sec** =(250, "5 sec")

5 sec

**jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.\_6sec** =(300, "6 sec")

6 sec

**jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.\_7sec** =(350, "7 sec")

7 sec

**jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.\_8sec** =(400, "8 sec")

8 sec

**jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.\_9sec** =(450, "9 sec")

9 sec

**jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection.Disable** =(0, "Disable trigger connect")

Disable trigger connect

---

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/TriggerButtonPushTimeForConnection.java](#)

---

## **jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection Enum Reference**

### **Public Method**

- String [toString](#) ()

### **Static Public Method**

- static
- [TriggerButtonPushTimeForDisconnection searchByValue](#) (int value)

### **Public Variable**

- [Disable](#) =(0, "Disable trigger disconnect")
  - [1sec](#) =(50, "1 sec")
  - [2sec](#) =(100, "2 sec")
  - [3sec](#) =(150, "3 sec")
  - [4sec](#) =(200, "4 sec")
  - [5sec](#) =(250, "5 sec")
  - [6sec](#) =(300, "6 sec")
  - [7sec](#) =(350, "7 sec")
  - [8sec](#) =(400, "8 sec")
  - [9sec](#) =(450, "9 sec")
-

## Explanation

This class is responsible for the trigger button push time duration when disconnecting.

---

## Method

static [TriggerButtonPushTimeForDisconnection](#)

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection.searchByValue (int value)[static]`

`String jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection.toString ()`

---

## Variable

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._1sec =(50, "1 sec")`  
1 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._2sec =(100, "2 sec")`  
2 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._3sec =(150, "3 sec")`  
3 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._4sec =(200, "4 sec")`  
4 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._5sec =(250, "5 sec")`  
5 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._6sec =(300, "6 sec")`  
6 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._7sec =(350, "7 sec")`  
7 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._8sec =(400, "8 sec")`  
8 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection._9sec =(450, "9 sec")`  
9 sec

`jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection.Disable =(0, "Disable trigger disconnect")`

This disables trigger connection.

---

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/TriggerButtonPushTimeForDisconnection.java](#)

#### jp.co.opto.opnsdk.setting.enums.UPCA\_TransmitCD Enum Reference

##### Public Method

- String [toString](#) ()

##### Static Public Method

- static [UPCA\\_TransmitCD searchByValue](#) (int b2, int b4)

##### Public Variable

- [Digit13\\_Reading0\\_TransmitCD](#) =(1, 1, "13-digit (read 0 and transmit CD)")
- [Digit12\\_NotReading0\\_TransmitCD](#) =(1, 0, "12-digit (do not read)")
- [Digit12\\_Reading0\\_NotTransmitCD](#) =(0, 1, "12-digit (do not transmit CD)")
- [Digit11\\_NotReading0\\_NotTransmitCD](#) =(0, 0, "11-digit (do not read 0 or transmit CD)")

---

#### Explanation

This class is responsible for the UPC-A format (transmit CD).

---

#### Method

static [UPCA\\_TransmitCD](#) jp.co.opto.opnsdk.setting.enums.UPCA\_TransmitCD.searchByValue (int *b2*, int *b4*)[static]

String jp.co.opto.opnsdk.setting.enums.UPCA\_TransmitCD.toString ()

---

#### Variable

jp.co.opto.opnsdk.setting.enums.UPCA\_TransmitCD.Digit11\_NotReading0\_NotTransmitCD =(0, 0, "11-digit (do not read 0 or transmit CD) ")

11-digit (do not read 0 or transmit CD)

jp.co.opto.opnsdk.setting.enums.UPCA\_TransmitCD.Digit12\_NotReading0\_TransmitCD =(1, 0, "12-digit (do not read 0) ")

12-digit (do not read 0)

jp.co.opto.opnsdk.setting.enums.UPCA\_TransmitCD.Digit12\_Reading0\_NotTransmitCD =(0, 1, "12-digit (do not transmit CD) ")

12-digit (do not transmit CD)

jp.co.opto.opnsdk.setting.enums.UPCA\_TransmitCD.Digit13\_Reading0\_TransmitCD =(1, 1, "13-digit (read 0 and transmit CD) ")

13-digit (read 0 and transmit CD)

---

The documentation for this enum was generated from the following file:

- [src/jp/co/opto/opnsdk/setting/enums/UPCA\\_TransmitCD.java](#)
-

## jp.co.opto.opnsdk.setting.enums.UPCE\_TransmitCD Enum Reference

### Public Method

- String [toString\(\)](#)

### Static Public Method

- static [UPCE\\_TransmitCD searchByValue](#) (int b2, int b4)

### Public Variable

- [Digit8\\_Reading0\\_TransmitCD](#) =(1, 1, "8-digit (read 0 and transmit CD) ")
- [Digit7\\_NotReading0\\_TransmitCD](#) =(1, 0, "7-digit (do not read 0) ")
- [Digit7\\_Reading0\\_NotTransmitCD](#) =(0, 1, "7-digit (do not transmit CD) ")
- [Digit6\\_NotReading0\\_NotTransmitCD](#) =(0, 0, "6-digit (do not read 0 or transmit CD) ")

---

### Explanation

This class is responsible for the UPC-E format (transmit CD).

---

### Method

static [UPCE\\_TransmitCD](#) jp.co.opto.opnsdk.setting.enums.UPCE\_TransmitCD.searchByValue (int *b2*, int *b4*)[static]

String jp.co.opto.opnsdk.setting.enums.UPCE\_TransmitCD.toString ()

---

### Variable

jp.co.opto.opnsdk.setting.enums.UPCE\_TransmitCD.Digit6\_NotReading0\_NotTransmitCD =(0, 0, "6-digit (do not read 0 or transmit CD) ")

6-digit (do not read 0 or transmit CD)

jp.co.opto.opnsdk.setting.enums.UPCE\_TransmitCD.Digit7\_NotReading0\_TransmitCD =(1, 0, "7-digit (do not read 0) ")

7-digit (do not read 0)

jp.co.opto.opnsdk.setting.enums.UPCE\_TransmitCD.Digit7\_Reading0\_NotTransmitCD =(0, 1, "7-digit (do not transmit CD) ")

7-digit (do not transmit CD)

jp.co.opto.opnsdk.setting.enums.UPCE\_TransmitCD.Digit8\_Reading0\_TransmitCD =(1, 1, "8-digit (read 0 and transmit CD) ")

8-digit (read 0 and transmit CD)

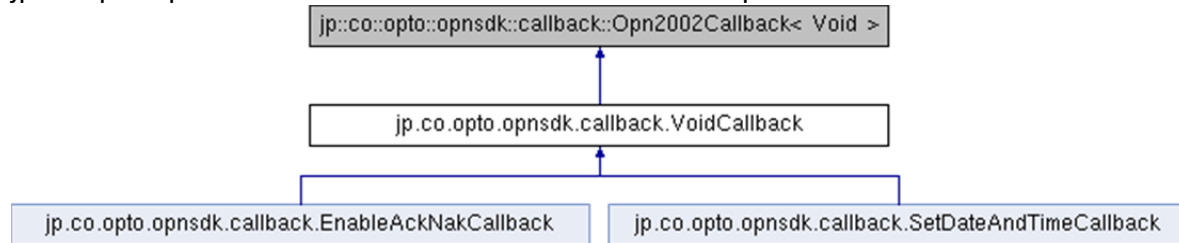
---

The documentation for this enum was generated from the following file:

- src/jp/co/opto/opnsdk/setting/enums/[UPCE\\_TransmitCD.java](#)
-

## Class `jp.co.opto.opnsdk.callback.VoidCallback`

### `jp.co.opto.opnsdk.callback.VoidCallback` Inheritance Graph



#### Public Method

- boolean [canExecute](#) (String *str*)

#### Protected Method

- Void [createParameter](#) (String *str*)

---

#### Explanation

This abstract class is responsible for the callback of commands which do not return results (ACK/NAK are returned).

---

#### Method

boolean `jp.co.opto.opnsdk.callback.VoidCallback.canExecute (String str)`

Void `jp.co.opto.opnsdk.callback.VoidCallback.createParameter (String str)[protected]`

---

The explanation for this class was created from the file below:

- `src/jp/co/opto/opnsdk/callback/`[VoidCallback.java](#)

## File

**src/jp/co/opto/opnsdk/BluetoothService.java**

### Structure

- class [jp.co.opto.opnsdk.BluetoothService](#)

### Package

- package [jp.co.opto.opnsdk](#)
- 

**src/jp/co/opto/opnsdk/BluetoothServiceState.java**

### Structure

- enum [jp.co.opto.opnsdk.BluetoothServiceState](#)

### Package

- package [jp.co.opto.opnsdk](#)
- 

**src/jp/co/opto/opnsdk/callback/DoDiscoveryCallback.java**

### Structure

- class [jp.co.opto.opnsdk.callback.DoDiscoveryCallback](#)

### Package

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/callback/EnableAckNakCallback.java**

### Structure

- class [jp.co.opto.opnsdk.callback.EnableAckNakCallback](#)

### Package

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/callback/GetBatteryLevelCallback.java**

### Structure

- class [jp.co.opto.opnsdk.callback.GetBatteryLevelCallback](#)

### Package

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/callback/GetDateAndTimeCallback.java**

### Structure

- class [jp.co.opto.opnsdk.callback.GetDateAndTimeCallback](#)

### Package

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/callback/GetFirmwareVersionCallback.java**

### Structure

- class [jp.co.opto.opnsdk.callback.GetFirmwareVersionCallback](#)

### Package

- package [jp.co.opto.opnsdk.callback](#)
-



**src/jp/co/opto/opnsdk/callback/GetSettingCallback.java**

**Structure**

- class [jp.co.opto.opnsdk.callback.GetSettingCallback](#)

**Package**

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/callback/IBluetoothCallback.java**

**Structure**

- interface [jp.co.opto.opnsdk.callback.IBluetoothCallback< T >](#)

**Package**

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/callback/Opn2002Callback.java**

**Structure**

- class [jp.co.opto.opnsdk.callback.Opn2002Callback< T >](#)

**Package**

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/callback/SetDateAndTimeCallback.java**

**Structure**

- class [jp.co.opto.opnsdk.callback.SetDateAndTimeCallback](#)

**Package**

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/callback/VoidCallback.java**

**Structure**

- class [jp.co.opto.opnsdk.callback.VoidCallback](#)

**Package**

- package [jp.co.opto.opnsdk.callback](#)
- 

**src/jp/co/opto/opnsdk/Command.java**

**Structure**

- class [jp.co.opto.opnsdk.Command](#)
- enum [jp.co.opto.opnsdk.Command.SettingChar](#)

**Package**

- package [jp.co.opto.opnsdk](#)
- 

**src/jp/co/opto/opnsdk/observer/IBluetoothObserver.java**

**Structure**

- interface [jp.co.opto.opnsdk.observer.IBluetoothObserver](#)

**Package**

- package [jp.co.opto.opnsdk.observer](#)
-

**src/jp/co/opto/opnsdk/Opn2002BluetoothService.java**

**Structure**

- class [jp.co.opto.opnsdk.Opn2002BluetoothService](#)

**Package**

- package [jp.co.opto.opnsdk](#)
- 

**src/jp/co/opto/opnsdk/setting/BluetoothSettings.java**

**Structure**

- class [jp.co.opto.opnsdk.setting.BluetoothSettings](#)

**Package**

- package [jp.co.opto.opnsdk.setting](#)
- 

**src/jp/co/opto/opnsdk/setting/DecoderSettings.java**

**Structure**

- class [jp.co.opto.opnsdk.setting.DecoderSettings](#)

**Package**

- package [jp.co.opto.opnsdk.setting](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/AckNakControl.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.AckNakControl](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/AckNakWaitTime.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.AckNakWaitTime](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/AddonWaitTime.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.AddonWaitTime](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/Authentication.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.Authentication](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/AutoDisconnectionTime.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.AutoDisconnectionTime](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/AutoReconnectionTime.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.AutoReconnectionTime](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/BuzzerMode.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.BuzzerMode](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/BuzzerPeriod.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.BuzzerPeriod](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/BuzzerTone.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.BuzzerTone](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/BuzzerVolume.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.BuzzerVolume](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/CapsLockMode.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.CapsLockMode](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/CommandResponse.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.CommandResponse](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
-

**src/jp/co/opto/opnsdk/setting/enums/ConnectionMode.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.ConnectionMode](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/EAN128\_Mode.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.EAN128\\_Mode](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/ErrorMessageMode.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.ErrorMessageMode](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/FunctionButtonInput.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.FunctionButtonInput](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/GoodReadLED\_ONTime.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.GoodReadLED\\_ONTime](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/IATA\_CalculateCD.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.IATA\\_CalculateCD](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/InterCharacterDelay.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.InterCharacterDelay](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/KeyboardLanguage.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.KeyboardLanguage](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/MemoryOutputMethod.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.MemoryOutputMethod](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/MSIPlessey\_CalculateCD.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.MSIPlessey\\_CalculateCD](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/MSIPlessey\_TransmitCD.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.MSIPlessey\\_TransmitCD](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/MultipleReadResetTime.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.MultipleReadResetTime](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/NumPadMode.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.NumPadMode](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

---

**src/jp/co/opto/opnsdk/setting/enums/NW7\_CalculateCD.java**

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.NW7\\_CalculateCD](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enum](#)

**src/jp/co/opto/opnsdk/setting/enums/NW7\_TransmitSTSP.java****Structure**

- enum [jp.co.opto.opnsdk.setting.enums.NW7\\_TransmitSTSP](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)src/jp/co/opto/opnsdk/setting/enums/ReadableTime.java

**Structure**

- enum [jp.co.opto.opnsdk.setting.enums.ReadableTime](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/ReadMode.java****Structure**

- enum [jp.co.opto.opnsdk.setting.enums.ReadMode](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/RedundantReading.java****Structure**

- enum [jp.co.opto.opnsdk.setting.enums.RedundantReading](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/SlaveConnectionWaitTime.java****Structure**

- enum [jp.co.opto.opnsdk.setting.enums.SlaveConnectionWaitTime](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/TriggerButtonPushTimeForConnection.java****Structure**

- enum [jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForConnection](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

**src/jp/co/opto/opnsdk/setting/enums/TriggerButtonPushTimeForDisconnection.java****Structure**

- enum [jp.co.opto.opnsdk.setting.enums.TriggerButtonPushTimeForDisconnection](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/UPCA\_TransmitCD.java****Structure**

- enum [jp.co.opto.opnsdk.setting.enums.UPCA\\_TransmitCD](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)
- 

**src/jp/co/opto/opnsdk/setting/enums/UPCE\_TransmitCD.java****Structure**

- enum [jp.co.opto.opnsdk.setting.enums.UPCE\\_TransmitCD](#)

**Package**

- package [jp.co.opto.opnsdk.setting.enums](#)

**src/jp/co/opto/opnsdk/setting/PrefixSettings.java**

**Structure**

- class [jp.co.opto.opnsdk.setting.PrefixSettings](#)

**Package**

- package [jp.co.opto.opnsdk.setting](#)
- 

**src/jp/co/opto/opnsdk/setting/SettingArea.java**

**Structure**

- class [jp.co.opto.opnsdk.setting.SettingArea](#)

**Package**

- package [jp.co.opto.opnsdk.setting](#)
- 

**src/jp/co/opto/opnsdk/setting/Settings.java**

**Structure**

- class [jp.co.opto.opnsdk.setting.Settings](#)

**Package**

- package [jp.co.opto.opnsdk.setting](#)
- 

**src/jp/co/opto/opnsdk/setting/SuffixSettings.java**

**Structure**

- class [jp.co.opto.opnsdk.setting.SuffixSettings](#)

**Package**

- package [jp.co.opto.opnsdk.setting](#)

★ Disclaimer

- This document has been carefully edited in order to minimize the number of mistakes and clerical errors. Opticon bears no responsibility regarding any direct loss or damage to the customer based on the off-chance of potential mistakes within this document.
- This document is subject to change without notice due to revisions made in the official specification manual.
- The following terminology utilized in this document are the properties of the below companies.

\* Bluetooth is the trademark of Bluetooth SIG, USA.

\* All other company and products named within this document are properties of their owners.

Model Name: OPN2002 Series  
OPN3002 Series  
Revision Number: 1.2  
DOC\_ID: SI12025