

SWAMP サービス SDK 仕様書

SWAMP 1.3.0 Release Edition

Revision 1.3.0

2017/03/10

Copyright© 3bitter 株式会社

改訂履歴

Revision	更新日	更新内容サマリー	更新者
1.0.0	2016/3/5	初版作成（Normal Edition から版分け）	上田
1.2.0	2016/10/14	SWAMP Edition として各 Edition を統合	上田
1.2.1	2017/1/11	標準ユースケース向けにデフォルト設定値を調整	上田
1.3.0	2017/3/10	dynamic ビーコン使用機能について追記	上田

1	はじめに.....	4
2	用語定義.....	5
3	SDK 構成ファイルと依存モジュール.....	7
3-1	SDK 構成ファイル.....	7
3-2	SDK が依存するフレームワーク.....	8
4	【設定ファイル仕様】.....	9
4-1	TbBTDeveloperPreference.plist.....	9
5	【クラス仕様】.....	11
5-1	TbBTManager クラス.....	11
5-2	TbBTDefaults クラス.....	24
5-3	TbBTPreliminary クラス.....	27
5-4	TbBTDeveloperPreference クラス.....	28
5-5	TbBTServiceBeaconData クラス.....	29
5-6	TbBTRegionNotificationSettingOptions クラス.....	30
5-7	TbBTBeaconizer クラス.....	31
5-8	TbBTOnetimeValidationClient クラス.....	33
6	端末言語設定対応.....	35
6-1	リソースファイル.....	35

1 はじめに

本 SDK (Software Development Kit) は、アプリケーションに対し、3bitter 株式会社が提供する専用ビーコン端末の制御使用、ビーコン端末で決定されるサービス用ビーコン領域或いは 3bitter 株式会社の運営管理下において利用可能な領域の使用と、一部使用状況の記録を可能とします。

1. バージョン

最新の SDK バージョンは 1.3.0 です

2. 必要要件

本 SDK は、iOS7.0 以上の動作環境を対象としています

3. 制約事項

4. 入手方法

直接お渡ししております

5. 問い合わせ

con@3bitter.com 宛にお問い合わせください。

2 用語定義

仕様説明上の用語を以下のように定義しています。一般的な技術用語については、別途定義を確認ください。

1. サービス・サーバ
3bitter 株式会社が運営する、または連動させるサービス資源を保有するサーバ（群）の総称。
2. [デバイス] サービスビーコン／サービス専用ビーコン／サービス固有ビーコン
3bitter 株式会社が提供する、独自のビーコン端末。
3. [デバイス] 指定ビーコン／指定済みビーコン（Designated Beacon）
上記 2 で定義されるビーコン端末のうち、ユーザ・インタフェースを使ったユーザ対話を経て、ユーザに「反応させる対象／反応をさせない対象」と指定された、特定のビーコン端末。
4. [論理データ] 予約ビーコン領域（Reserved Region）
上記 2 で定義されるビーコン端末（固定値ビーコン）用にサービス上割り当てられている領域定義。あらかじめ用意された領域 ID を持っています。指定済みビーコン領域も含みます
5. [論理データ] 指定ビーコン領域（Designated Beacon Region）
上記 3 で定義されるビーコン端末用に SDK で割り当てられている領域定義。
6. [論理データ] 管理ビーコン領域（Managed Region）
2 以外のビーコンにサービス上で割り当てられていて、アプリで使えるように管理されている領域情報。（領域情報は不特定多数）
7. サービス・キャンペーン

3bitter 株式会社の運用サービス下で管理される、領域連動型サービス用のキャンペーン情報。管理内容及び通知内容を含む。

8. 管理システム

本 SDK を使用したサービスが成り立つように、あんなことやこんなことを設定管理する／してもらうためのシステム。広義としては、サービス・サーバを含む。

9. セグメント

サービス運用上、ビーコン端末群を区分けするための予約情報パート。

10. キーコード

2 で定義されるビーコン端末を、ユーザが所有権の明確化、他のビーコン領域での反応の制限等を目的として、ビーコン端末の使用指定をするためにサービス側で割り当てている端末管理上の文字列。

11. スイッチャービーコン

他の 3bitter 用ビーコンの検知を開始したり、停止したりする制御用に SDK を介して指定定義するビーコン。

[New ver.1.3.0~]

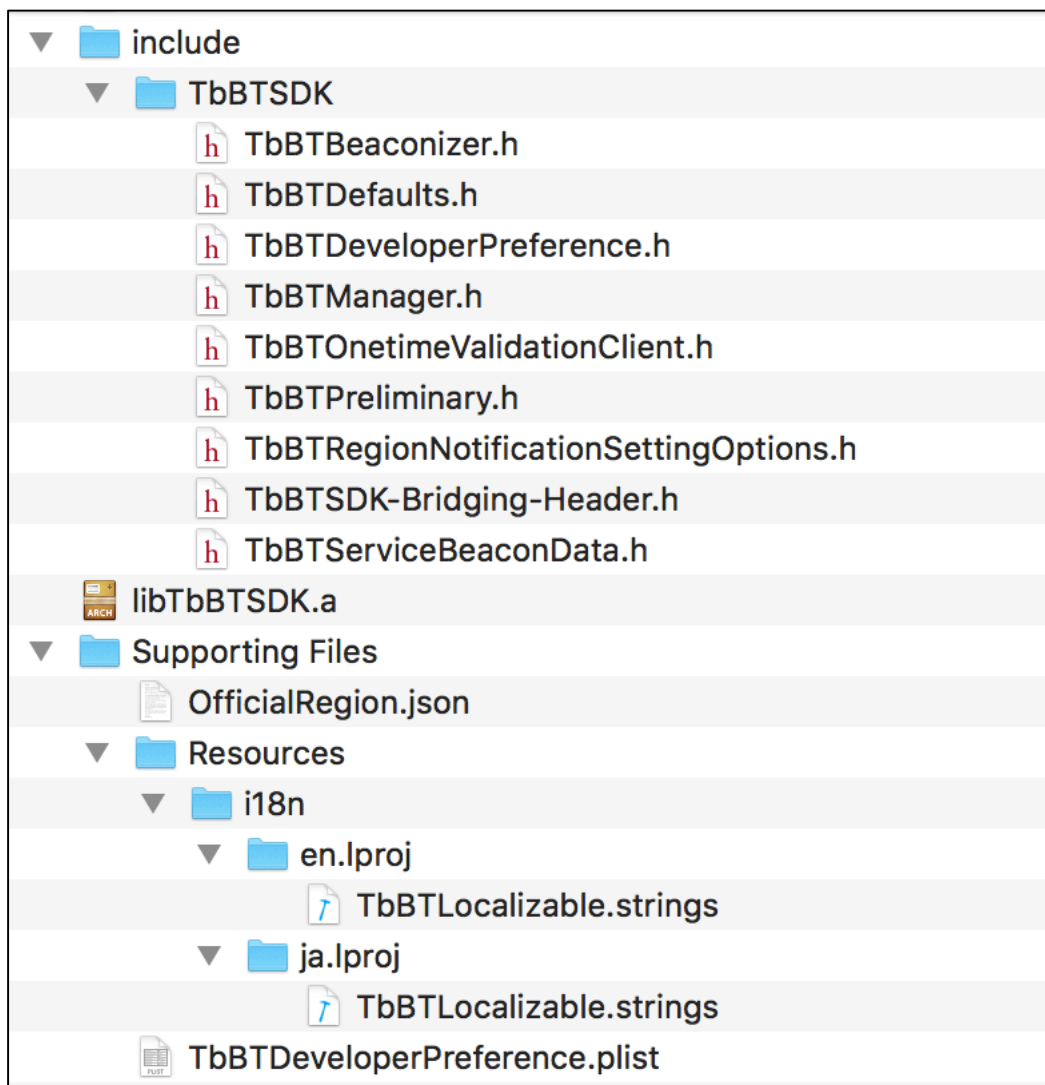
12. 可変値ビーコン（ダイナミックビーコン）

従来のビーコン（本書内では「固定値ビーコン」と表記）と違い、発信内容の一部値を可変としたビーコン。

※ ビーコン（とビーコン電波が形成する領域）にアイデンティティをもたせ、その他センシティブな情報に結びつけて使用する場合などに、同一ビーコンをコピーあるいは偽装したなりすましを予防・早期事後対応（なりすましの無効化）する目的の製品

3 SDK 構成ファイルと依存モジュール

3-1 SDK 構成ファイル



本体静的ライブラリ (1)

- libTbBTSDK.a

ヘッダファイル (9)

- TbBTPreliminary.h ... サービスを開始するにあたり、事前準備をするためのクラスのヘッダファイル
- TbBTServiceDefaults.hサービスのデフォルト設定クラスのヘッダファイル
- TbBTManager.h ... ビーコン領域に反応して処理を担うメインのクラスの

ヘッダファイル

- **TbBTDeveloperPreference.h** ... SDKを導入するアプリ毎の設定を保持するクラスのヘッダファイル
- **TbBTServiceBeaconData.h** ... 本サービスのビーコンを制御するための情報クラスのヘッダファイル
- **TbBTRegionNotificationSettingOptions.h** ... ビーコン領域の反応特性を指定するための設定オプションクラスのヘッダファイル
- **TbBTBeaconizer.h** ... スマートフォン端末（iOS 端末）を利用して、アプリ上からビーコン発信を実行するためのクラスのヘッダファイル

[New ver.1.3.0～]

- **TbBTONetimeValidationClient.h** ... 可変値ビーコンの現在の妥当性をチェックするためのクラスのヘッダファイル

[開発言語ブリッジ用ヘッダファイル]

- **TbBTSDK-Bridging-Header.h** ... Swift 言語で開発されるアプリから SDK のクラスを呼び出すための橋渡し用宣言ファイル

[サポートファイル]

TbBTDeveloperPreference.plist アプリケーション毎の設定値を設定するためのファイル

Resources – i18n ... 多言語表示用のメッセージ定義ファイル群

Resources – OfficialRegion.json ... SDK ビルド時点での公式ビーコン領域用情報(※ SDK は使用時の最新ビーコン領域情報をサーバから取得します)

3-2 SDK が依存するフレームワーク

本 SDK は以下のフレームワークに依存しています。

1. CoreLocation.framework (required)
2. CoreBluetooth.framework (required)
3. AdSupport.framework (optional ※ ビルドする版により必要)
4. SystemConfiguration.framework (required)

4 【設定ファイル仕様】

4 - 1 TbBTDeveloperPreference.plist

Key	Type	Value
▼ Root	Dictionary	(6 items)
TestMode	Boolean	YES
DeveloperAppCode	String	D24c345Lc7XRfGf7
SDKEdition	String	SWAMP
SDKVersion	String	1.3.0
MaxDesignatableBeaconRegions	Number	19
RegionUpdateIntervalHours	Number	12

[役割]

アプリケーション毎の、サービス設定を担います。

TestMode (Boolean)

テスト使用モード (YES) か本運用モード (NO) を指定します。デフォルトで「YES」になっています。公開前に「NO」に変更してください。

DeveloperAppCode (String)

サービス運営から取得したアプリ固有のアクセス・トークンを設定します。

SDKEdition (String)

SDK のタイプを表します。正常な SDK か否かの判定に使用しますので、変更しないでください。本エディションでは、「SWAMP」となります

SDKVersion (String)

SDK のバージョンを表します。正常な SDK か否かの判定に使用しますので、変更しないでください。

MaxDesignatableBeaconRegions (Number)

SDK を介して登録する指定ビーコン領域の最大数です。指定ビーコン以外に基本領域がモニタリングされる必要があるので、デフォルトで 19 です。

(注) アプリに対しては OS により 20 領域のモニタリングに制限されるので、SDK 外で別の領域をモニタリングする場合はこの制限値によらず、反応する指定ビーコンの数が少なくなる可能性があります

RegionUpdateIntervalHours (Number)

サーバからアプリで使用可能な公式ビーコン以外の領域情報を取得する場合に、一度更新した後に何時間の更新スキップをするかを指定します。登録している領域情報を頻繁に更新する場合には、更新情報を早期に反映させるため短くします。一度登録された領域情報を更新しない場合は長くすることにより余分なサーバ間通信を発生させずに済みますが、登録領域が増えた場合にアプリ側に反映するタイミングが遅くなることがあります。デフォルトでは 12 時間です。

5 【クラス仕様】

5 - 1 TbBTManager クラス

[役割]

サービス・サーバとのやり取りを統括します。

アプリケーションから指示を受け、TbBTManagerDelegate プロトコルを実装した呼び出し元に処理結果後の対応を委譲します。

Singleton クラスのため、アプリケーションで 1 つのインスタンスになります。

[プロパティ]

id<TbBTManagerDelegate> delegate TbBTManagerDelegate プロトコルを実装するインスタンス 必須

NSString auid アプリ側でユーザの特定に使用する文字列 任意

[メソッド]

(生成と初期化)

+ (TbBTManager *) initSharedManagerUnderAgreement:(BOOL)allowed

サービス使用への許可を受けとり、インスタンスを返します。

allowed = NO の場合、戻り値は nil になります。

前提条件：SDK 同梱の、[TbBTDeveloperPreference.plist]が必要です。

+ (TbBTManager *)sharedManager

アプリケーション内で共有される TbBTManager クラスのインスタンスを返します。**※ initSharedManagerUnderAgreement:YES が実行されていない場合、戻り値は nil になります。**

(SDK の使用に関わるハードウェア及びユーザ設定状態のチェック)

+ (BOOL)isBeaconEventConditionMet

ユーザによる全体またはアプリに対しての位置情報サービスの設定状態が、ビーコン領域に関わる処理が「常に」期待通りに実行できる状態か否かを返すユ

ユーティリティメソッドです。

ビーコン領域関連のメソッドが期待通りの処理ができる状態であれば YES を、ビーコン領域に関係するメソッドが期待通りに実行できない状態である場合は NO を返します。

※ 位置情報サービス、Bluetooth 等の端末対応やユーザ設定の状況で判定されます

※ ユーザによる Bluetooth のオン／オフ状況は確認しません

+ (BOOL)isBeaconEventConditionMetForForegroundOnly

ユーザによる全体またはアプリに対しての設定状態が、ビーコン領域に関わる処理が「アプリがアクティブな状態に限っては」実行を妨げないかどうかをチェックするユーティリティメソッドです。

期待通りの処理ができる状態であれば YES を、ビーコン領域に関係するメソッドが期待通りに実行できない状態である場合は NO を返します。

+ (BOOL)isSupportedBeaconEventByDevice

ビーコン領域に関わる処理が端末によってサポートされているかどうかのみをチェックするユーティリティメソッドです。ユーザ設定の状況については確認しません。

期待通りの処理ができる状態であれば YES を、ビーコン領域に関係するメソッドが期待通りに実行できない状態である場合は NO を返します。

(ビーコンの検出に関わるユーザ Bluetooth 設定状態のチェック)

- (void)checkCurrentBluetoothAvailability

ユーザによる現在の OS レベルでの Bluetooth 設定がオンになっていて、Bluetooth が受信できる状態かチェックします。

非同期メソッドのため、本メソッドを使用する場合は TbBTManager のデリゲート指定をしたクラスに、didDetermineBluetoothAvailability: デリゲートメソッドを実装してください。

(アプリ側の設定値)

- **(TbBTDeveloperPreference *)**preference

アプリ固有のサービス使用時の設定を取得します (※ TbBTDeveloperPreference の仕様を参照)。

(サービス専用ビーコン領域)

- **(NSArray *)**initialRegions

専用ビーコン領域としてサービス用に用意されている領域のうち有効なビーコン領域リストを CoreLocation.Framework で用意されている CLBeaconRegion 型で返します。

- **(NSArray *)**tbDynamicBeaconRegions

専用ビーコン領域としてサービス用に用意されている領域のうち可変値ビーコン (ダイナミックビーコン) 用の領域リストを CoreLocation.Framework で用意されている CLBeaconRegion 型で返します。

- **(BOOL)**isReservedRegion:(CLBeaconRegion *)region

指定済みビーコンの領域を含め、SWAMP 専用固定値ビーコン領域として予約されている領域か否かをチェックします。

与えられたビーコン領域が専用固定値ビーコンの領域であれば、YES を、専用ビーコンでない領域であれば、NO を返します

- **(BOOL)**isInitialRegion:(CLBeaconRegion *)region

指定済みビーコン以外の、SWAMP 専用固定値ビーコン領域 (複数のビーコン端末に反応する汎用領域) か否かをチェックします。

指定済みビーコン以外の SWAMP 専用固定値ビーコンに反応する領域なら YES、それ以外なら NO を返します

- **(BOOL)**isTbDynamicBeaconRegion:(CLBeaconRegion *)region

渡された領域 region が、3bitter 提供の可変値ビーコンの領域か否かを返します。YES であれば、3bitter 製可変値ビーコンの領域です。

(サービス用ビーコン領域のコントロール委任)

**-(BOOL)startMonitoringTbBTInitialRegions:(CLLocationManager
*)locationManager**

不特定状態の、サービス用に定義されたビーコン領域 (InitialRegions) のモニタリングを開始します。この状態のままビーコンの指定がないと、全てのサービスビーコンに反応します。問題なく処理された場合 YES を、処理の途中で問題が発生した場合、NO を返します。

locationManager が nil の場合、NO を返します

※ 本メソッドを使用した場合、以下の領域プロパティ値はデフォルトのままです

CLRegion.notifyOnEntry = YES

CLRegion.notifyOnExit = YES

CLBeaconRegion.notifyEntryStateOnDisplay = NO

**-(BOOL)startMonitoringTbBTInitialRegions:(CLLocationManager
*)locationManager**

withSupportTypeOptions:(TbBTRegionNotificationSettingOptions *)options

不特定状態の、サービス用に定義されたビーコン領域のモニタリングを options (※TbBTRegionNotificationSettingOptions クラス参照) で指定された設定で開始します。この状態のままビーコンの指定がないと、全てのサービスビーコンに反応します。問題なく処理された場合 YES を、処理の途中で問題が発生した場合、NO を返します。

※ 本メソッドを使用した場合、サービス用ビーコン領域の、以下の領域プロパティ値を変更します

CLRegion.notifyOnEntry

CLRegion.notifyOnExit

CLBeaconRegion.notifyEntryStateOnDisplay

[New ver.1.3.0～]

**-(BOOL)startMonitoringTbBTDynamicRegions:(CLLocationManager
*)locationManager**

3bitter 製可変値ビーコンの作る領域 (DynamicRegions) のモニタリングを開始します。問題なく処理された場合 YES を、処理の途中で問題が発生した場合、

NO を返します。

locationManager が nil の場合、NO を返します

※ 本メソッドを使用した場合、以下の領域プロパティ値はデフォルトのままです

```
CLRegion.notifyOnEntry = YES
```

```
CLRegion.notifyOnExit = YES
```

```
CLBeaconRegion.notifyEntryStateOnDisplay = NO
```

[New ver.1.3.0~]

-(BOOL)startMonitoringTbBTDynamicRegions:(CLLocationManager *)locationManager

withSupportTypeOptions:(TbBTRegionNotificationSettingOptions *)options

3bitter 製可変値ビーコンの作る領域 (DynamicRegions) のモニタリングを options (※TbBTRegionNotificationSettingOptions クラス参照) で指定された設定で開始します。問題なく処理された場合 YES を、処理の途中で問題が発生した場合、NO を返します。

※ 本メソッドを使用した場合、ビーコン領域の、以下の領域プロパティ値を変更します

```
CLRegion.notifyOnEntry
```

```
CLRegion.notifyOnExit
```

```
CLBeaconRegion.notifyEntryStateOnDisplay
```

-(BOOL)startMonitoringTbBTNonSwitcherBeaconRegions:(CLLocationManager *)locationManager

スイッチャービーコン (用語定義参照) 以外の 3bitter 製ビーコン領域のモニタリングを開始します。スイッチャービーコン (領域) のイベントからの使用を想定しています。

-(BOOL)startMonitoringTbBTNonSwitcherBeaconRegions:(CLLocationManager *)locationManager

withSupportTypeOptions:(TbBTRegionNotificationSettingOptions *)options

スイッチャービーコン (用語定義参照) 以外の 3bitter 製ビーコン領域のモニタリングを開始します。

options パラメータを指定して、反応イベントの選択ができます。

-(NSUInteger)stopMonitoringTbBTInitialRegions:(CLLocationManager *)locationManager

不特定状態の、サービス用に定義されたビーコン領域のモニタリングの停止指示を（locationManager に）出した領域の数を返します。

locationManager が nil の場合、0 を返します

※ 実際に停止したか否かは、locationManager のデリゲート側で判別になります

[New ver.1.3.0～]

-(NSUInteger)stopMonitoringTbBTDynamicRegions:(CLLocationManager *)locationManager

3bitter 製可変値ビーコン領域のモニタリングの停止指示を（locationManager に）出した領域の数を返します。

locationManager が nil の場合、0 を返します

※ 実際に停止したか否かは、locationManager のデリゲート側で判別になります

-(NSUInteger)stopMonitoringTbBTAllRegions:(CLLocationManager *)locationManager

サービス用に定義されたビーコン領域のモニタリングを停止します。停止指示を（locationManager に）出した領域の数を返します。

locationManager が nil の場合、0 を返します

-(NSUInteger)stopMonitoringTbBTNonSwitcherBeaconRegions:(CLLocationManager *)locationManager

スイッチャービーコン（用語定義参照）を登録しているときに、スイッチャービーコン以外のビーコンのモニタリングを停止させます。スイッチャービーコンのビーコン（領域）イベントからコールする想定の方法です。

停止指示を（locationManager に）出した領域の数を返します。

locationManager が nil の場合、0 を返します

-(BOOL)startRangingTbBTStaticBeacons:(CLLocationManager *)locationManager

サービス用ビーコン（固定値タイプ）の計測を開始します。

[New ver.1.3.0～]

-(BOOL)startRangingTbBTDynamicBeacons:(CLLocationManager *)locationManager

3bitter 専用動的ビーコン（可変値タイプ）の計測を開始します。

-(BOOL)stopRangingTbBTStaticBeacons:(CLLocationManager *)locationManager

3bitter 専用ビーコン（固定値タイプ）の計測を停止します。

[New ver.1.3.0～]

-(BOOL)stopRangingTbBTDynamicBeacons:(CLLocationManager *)locationManager

3bitter 専用動的ビーコン（可変値タイプ）の計測を停止します。

-(NSArray *)keyCodesForBeacons:(NSArray *)beacons ofRegion:(CLBeaconRegion *)region

サービス用固定値ビーコンの制御用情報 TbBTServiceBeaconData 型の配列を返します。**region** がサービス用の固定値ビーコン領域でない場合 （3bitter 製可変値ビーコン領域含む） や **beacons** の数が 0 の場合、nil が返ります

※ 本メソッドでは、指定済みビーコンの情報も応答配列に含みます

-(NSArray *)keyCodesForBeaconsExcludeDesignated:(NSArray *)beacons ofRegion:(CLBeaconRegion *)region

渡された **beacons** のうち、指定済み以外のサービス用固定値ビーコンの制御用情報 TbBTServiceBeaconData 型の配列を返します。**region** がサービス用の固定値ビーコン領域でない場合 （3bitter 製可変値ビーコン領域含む） や **beacons** の数が 0 の場合、nil が返ります

-(NSArray *)beaconsTrack:(NSArray *)beacons ofRegion:(CLBeaconRegion

***)region**

渡された region が 3bitter 専用固定値ビーコン領域である場合に、規定条件に従い検出結果調整処理を行い、規定条件に達した場合サーバに非同期で記録送信を行います。送信の再送制御は SDK 内で実行（※1）します。

同時に累積された TbBTServiceBeaconData 型の配列を返します。

規定条件に達していない場合、nil が返ります。

規定条件に達しても適切なビーコン情報が得られない場合（※2）や処理タイムアウトした場合は、空の配列が返ります。

タイムアウトは 15 秒です。

region がサービス用の固定値ビーコン領域でない場合（3bitter 製可変値ビーコン領域含む）や beacons の数が 0 の場合、nil が返ります

※1 再送制御（アプリがアクティブな場合のみ） 上限回数は 3 回です。

※2 beacons が空リストで渡され続けた場合等で起こり得ます。

-(NSInteger)specifyNewUsableServiceRegionsWithCodes:(NSArray

***)beaconDatas forRegion:(CLBeaconRegion *)region**

locationManager:(CLLocationManager *)locationManager

サービス固有ビーコン領域について、与えられた情報（TbBTServiceBeaconData 型）にマッチする領域のみアプリが使用できるように制限します。与えられたビーコン特有領域のみのモニタリングを開始し、それまでモニタリングされていた region のモニタリング（不特定領域に反応）を停止します。

処理結果に従い以下の値を返します。

[応答コード値]

値 説明		beaconDatas 登録
1	与えられた指定ビーコン情報が正常に登録されました	全て登録
0	指定ビーコン情報の登録数が上限に達しました（上限数以下のビーコンは登録されます）	一部登録
-1	必須パラメーター locationManager の値が nil です	されない
-2	必須パラメーター region の値が nil です	されない

-3	必須パラメーター <code>beaconDatas</code> の要素数が 0 です	されない
-4	与えられた <code>region</code> がサービス用の適切なビーコン領域ではありません	されない
-5	与えられた <code>beaconDatas</code> の中に不当なキーコード値のビーコン情報があります	されない
-6	登録可能なビーコン数を超えています	されない
-7	SDK 側で情報の永続化処理に失敗しました	されない

※ 本メソッドを使用した場合、以下の領域プロパティ値はデフォルトのままです

`CLRegion.notifyOnEntry = YES`

`CLRegion.notifyOnExit = YES`

`CLBeaconRegion.notifyEntryStateOnDisplay = NO`

- (NSInteger)specifyNewUsableServiceRegionsWithCodes:(NSArray
*)beaconDatas forRegion:(CLBeaconRegion *)region
locationManager:(CLLocationManager *)locationManager
withOptions:(TbBTRegionNotificationSettingOptions *)options

サービス固有ビーコン領域について、与えられた情報 (TbBTServiceBeaconData 型) にマッチする領域のみアプリが使用できるように制限します。与えられたビーコン特有領域のみのモニタリングを開始し、それまでモニタリングされていた `region` のモニタリング（不特定領域に反応）を停止します。

処理結果に従い返る値は以下の通りです。

値	説明	beaconDatas 登録
1	与えられた指定ビーコン情報が正常に登録されました	全て登録
0	指定ビーコン情報の登録数が上限に達しました (上限数以下のビーコンは登録されます)	一部登録
-1	必須パラメーター <code>locationManager</code> の値が <code>nil</code> です	されない
-2	必須パラメーター <code>region</code> の値が <code>nil</code> です	されない
-3	必須パラメーター <code>beaconDatas</code> の要素数が 0 です	されない
-4	与えられた <code>region</code> がサービス用の適切なビーコン領域では	されない

	ありません	
-5	与えられた <code>beaconDatas</code> の中に不当なキーコード値のビーコン情報があります	されない
-6	登録可能なビーコン数を超えています	されない
-7	SDK 側で情報の永続化処理に失敗しました	されない
-8	必須パラメーター <code>options</code> の値が <code>nil</code> です	されない

※ 本メソッドを使用した場合、サービス用ビーコン領域の、以下の領域プロパティ値を変更します

`CLRegion.notifyOnEntry`

`CLRegion.notifyOnExit`

`CLBeaconRegion.notifyEntryStateOnDisplay`

- (NSArray *)currentUsableServiceBeaconDatas

現状で反応するように指定されているサービス固有ビーコン情報 (`TbBTServiceBeaconData` のインスタンス) のリストを返します。該当ビーコン見つからなければ `nil` が返ります

- (NSArray *)regionsOfDesignatedBeacons

(ユーザに) 指定されているビーコンの、iOS 仕様での領域情報 (`CLBeaconRegion` のインスタンス) を配列で返します。存在しなければ `nil` が返ります

- (BOOL)updateDesignatedBeaconData:(TbBTServiceBeaconData *)beaconData

SDK を介して登録されているビーコンの情報を上書き更新します。

※ 現行仕様では、`switcher` フラグのみを変更します

更新登録に成功したら YES を、失敗したら NO を返します

- (BOOL)updateDesignatedBeaconDatas:(NSArray *)beaconDatas

SDK を介して登録されている複数のビーコンの情報を一括で上書き更新します。

引数として、`TbBTServiceBeaconData` のインスタンスの配列を受け取ります。

※ 現行仕様では、**switcher** フラグのみを変更します
更新登録に成功したら YES を、失敗したら NO を返します

- (BOOL)releaseUsableServiceBeacon:(TbBTServiceBeaconData *)beaconData locationManager:(CLLocationManager *)locationManager

指示された反応ビーコンのモニタリングを停止し、反応指定を解除します。処理に成功すると YES、成功しなかった場合 NO を返します。

※ 以降、対象のビーコンを反応させるには
startMonitoringTbBTInitialRegions メソッドを使用したモニタリング制御で、該当サービス領域のモニタリングを開始する必要があります。

- (BOOL)hasDesignatedBeacon

排他使用のため（ユーザに）指定されたビーコンが存在するか否かを判定します。指定ビーコンが登録済みの場合 YES を、登録が無い場合に NO を返します

- (BOOL)isDesignatedBeacon:(CLBeacon *)beacon

渡されたビーコンが指定済みビーコンであるか否かを判定します。指定済みビーコンであるある場合に YES を、指定済みビーコンと異なる場合に NO を返します

（サービス側管理のアプリ毎に使用可能なビーコン領域の管理）

-(void)updateManagedRegionsWithType:(TbBTManagedRegionType)regionType

サービス・サーバ側で管理されている、アプリで使用可能な最新の領域情報を取得し、新規、廃棄の仕分けをします。このメソッドを実行しただけでは、実際にモニタリングされている領域の更新はされません。

※ **regionType** に指定可能な値については [TbBTDefaults] クラスの、「TbBTManagedRegionType」を参照してください。

-(void)updateOwnedRegionsOfBeaconType:(TbBTBeaconType)beaconType

アプリ開発側で SWAMP システムに登録した最新の自社専用ビーコン領域情報をビーコンタイプを指定して更新取得します。

- (NSArray *)currentManagedRegions

サービス・サーバ側管理から取得され SDK に保持されている領域情報の一覧（配列）を返します

※実際にモニタリングされている領域とは異なります

- (NSArray *)newManagedRegions

サービス・サーバ側管理から領域情報が取得された後で、取得前と比較して（管理側で）追加された領域情報の一覧（配列）を返します

- (NSArray *)abandonedManagedRegions

サービス・サーバ側管理から領域情報が取得された後で、取得前と比較して廃棄された領域情報の一覧（配列）を返します

- (BOOL)isManagedRegion:(CLBeaconRegion *)region

3bitter 提供の専用ビーコンの領域以外で、サービス用にアプリで利用できる領域か否かをチェックします。

与えられたビーコン領域がアプリで使えるようにサービス登録されたものであれば YES を、本サービスに関係のない領域であれば、NO を返します

-(BOOL)fireTestLoggingRequestForBeaconRegionEvent:(CLBeaconRegion *)region event:(TbBTEventType)eventType

TbBTDeveloperPreference.plist 内「TestMode」設定が「YES」の場合に、本メソッドを使用して、ビーコン領域でのイベントが発生した時の情報をサーバ側に記録し、処理確認をすることができます。

※ eventType に指定可能な値については[TbBTServiceDefaults]クラスの、「TbBTEventType」を参照してください。

[デリゲートプロトコルメソッド (TbBTManagerDelegate)]

(Bluetooth 使用可否確認結果の通知)

- (void)didDetermineBluetoothAvailability:(BOOL)available

現在の Bluetooth 使用可否チェック結果を返します。

available = YES の場合使用可能（ユーザによる OS レベルでの設定 ON）

available = NO の場合、使用不可（ユーザによる設定 OFF 及び、他機能による制限がある場合）

（領域情報処理結果の通知）

-(void)didPrepareRefreshRegionsWithResult(TbBTPrepareResultType)resultType

管理システムからの最新領域情報の取得と仕分けが問題なく完了したことを通知します。結果区分により、newManagedRegions 及び abandonedManagedRegions の内容が変わります。

※ [TbBTServiceDefaults]クラスの、「TbBTPrepareResultType」を参照してください。

-(void)didFailToPrepareLatestRegionsWithError:(NSError *)error

管理システムからの最新領域情報の取得及び仕分けに失敗したことを通知します。

[エラー区分]

- ・ 応答結果の解析に失敗
- ・ アプリに対し提供可能な管理されている領域が存在しない（エラードメイン：TbBTFetchError、エラーコード:1400）
- ・ 前回の更新から設定のインターバル時間が経過していないため更新処理をスキップした（エラードメイン：TbBTFetchError、エラーコード:1500）
- ・
- ・ 通信エラー（エラードメイン：TbBTHTTPError、エラーコード：HTTP ステータスコード）

（その他低レベルエラーが発生した場合は規定区分での NSError オブジェクトが返されます）

-(void)didSendTestLogRequestSuccessfully

管理システムへテストログリクエストが送信され、受け付けられた（HTTP 200 OK）ことを通知します。

※非同期送信のため、管理システム側でのログデータの処理結果の影響は受けません。

- (void)didFailToSendTestLogRequestWithError:(NSError *)error

管理システムへのテストログリクエスト送信が、error により失敗したことを通知します。

5 - 2 TbBTDefaults クラス

[役割]

サービス固有の設定を保持します

[プロパティ]

NSArray usingServiceRegionInfos (読み取り専用)

実際に使用されている専用固定値ビーコンの領域情報 (NSDictionary 型) のリスト

NSArray reservedServiceUUIDs (読み取り専用)

サービス固有に予約されている専用の領域 UUID のリスト。

NSArray usingDynamicRegionInfos (読み取り専用)

使用されている専用可変値ビーコンの領域情報 (NSDictionary 型) のリスト

[タイプ定義]

TbBTHttpStatus: HTTP 通信のステータス区分

※ RFC2616 に定義の主な HTTP Status Code に同じ

TbBTEventType: ビーコン領域関連のイベント区分

- **TbBTEventTypeDidEnter**
ビーコン領域に入りました
- **TbBTEventTypeDidExit**

ビーコン領域から出ました

- **TbBTEventTypesIsInside**
ビーコン領域内にいる状態です
- **TbBTEventTypesIsOutside**
ビーコン領域外にいる状態です

TbBTManagedRegionType: 管理システム側でアプリ用に登録されている、サービス専用以外のビーコン領域の管理区分

- **TbBTManagedRegionTypeWhole**
全ての管理領域
- **TbBTManagedRegionTypeTest**
テスト用のビーコン領域
- **TbBTManagedRegionTypeOwned**
デベロッパー専用のビーコン領域（デベロッパー側用意）
- **TbBTManagedRegionType3rd**
サービス連動用意されている第3者ビーコン領域（運営側用意）

TbBTBeaconType: 管理システム側で登録されている、ビーコン領域用のビーコンの管理区分

- **TbBTBeaconTypeServiceStatic**
3bitter サービス専用の固定値ビーコン
- **TbBTBeaconTypeServiceDynamic**
3bitter サービス専用の変化値ビーコン
- **TbBTBeaconTypeOwnedStatic**
デベロッパー専用の固定値ビーコン（デベロッパー側用意）
- **TbBTBeaconTypeOwnedDynamic**
デベロッパー専用の変化値ビーコン（デベロッパー側用意）

TbBTPrepareResultType: SDK とサーバでの連携によるデータ準備結果区分

- **TbBTPrepareResultTypeNoDifference**
最新データは、SDK で保持していたデータとの差分がありません
- **TbBTPrepareResultTypeHasNew**

最新データに、SDK で保持していたデータと比較して追加データがあります

- **TbBTPrepareResultTypeHasAbandoned**

最新データに、SDK で保持していたデータと比較して削除されたデータがあります

- **TbBTPrepareResultTypeHasNewAndAbandoned**

最新データに、SDK で保持していたデータと比較して追加及び削除されたデータの両方があります

TbBTContactCheckResultType: サーバへのビーコン連動処理の結果区分

- **TbBTContactCheckResultTypeDoNothing**

ビーコン領域に対して処理すべきことはありません

- ・ 該当領域に対して必要な処理が既に実行済みである場合は、この区分が返ります

- **TbBTContactCheckResultTypeMarked**

ビーコン領域に対して、キャンペーンに必要なマーキング処理が実施されました

- **TbBTContactCheckResultTypeTotallyBlocked**

チェック処理の結果通知内容が 1 つ以上見つかりましたが、制約条件により全てのキャンペーン通知内容はブロックされました

- ・ 同じ端末に本 SDK を使用したアプリが 2 つ以上インストールされていて、同一のビーコン領域をモニタリングしているケースで発生する可能性があります

- **TbBTContactCheckResultTypePartBlocked**

チェック処理の結果通知内容が 2 つ以上見つかりましたが、制約条件によりいずれかのキャンペーン通知内容がブロックされました

- **TbBTContactCheckResultTypeUnmarkable**

キャンペーンが見つかりましたが、マーキング処理できない状態でした

- **TbBTContactCheckResultTypeTargetNotFound**

ビーコン領域に対して対象としているサービスもキャンペーンも見つかりませんでした

- ・ **TbBTContactCheckResultTypeSkipped**

ビーコン領域に対して、キャンペーンのマーキング要件が設定されていますが、既に全ての要件は満たされているので処理はスキップされました（次に通知条件にマッチしたタイミングで、通知内容が応答されます）

[メソッド]

+ (TbBTDefaults *)sharedDefaults

アプリケーションで共有される 1 つの TbBTDefaults のインスタンスを返します。

- (NSString *)SDKIdentifier

SDK の簡易識別子を返します。（バージョン情報は含みません）

5 - 3 TbBTPreliminary クラス

[役割]

ビーコン領域連動型トラッキングサービスを実行するための準備をします。

[プロパティ]

無し

[メソッド]

+ (void)setUpWithCompletionHandler:(void (^)(BOOL success))completionHandler

ビーコン領域情報等の準備処理を行います。

アプリケーション起動時に一度だけ呼び出しが必要です。

何らかの問題により、ビーコンネットワークサービスが使用できない状態で処理が中断すると、completionHandler 処理ブロック(※)の引数としての success に NO が、問題なく処理が完了すると success = YES が返ります。

completionHandler に nil を渡した場合は、結果を応答せず処理を終了します。

※ 必要に応じてビーコンネットワークを使用するサービス用の処理スキップ等の処理を実装してください。

[New] ver.1.3.0～

+(BOOL)clearSetupData

setUpWithCompletionHandler で準備した SWAMP サービス用のデータの消去が必要な場合に、実行します。（※ 開発時、OS バージョンアップに伴う仕様変更があった場合等）

5 - 4 TbBTDeveloperPreference クラス

[役割]

アプリケーションで固有のサービス設定値を保持します。
(TbBTDeveloperPreference.plist 設定ファイルの値を保持)

[プロパティ]

NSString appCode SDK を導入するアプリ毎に発行された、アクセス・トークン（読み取り専用）

NSString sdkEdition SDK のタイプ（読み取り専用）

NSString sdkVersion SDK のバージョン（読み取り専用）

NSUInteger maxDesignatableBeaconRegions SDK を使ったビーコン領域のモニタリング管理で個別に指定登録できるビーコン（領域）の最大数（読み取り専用）

NSUInteger regionRefreshIntervalHours 領域更新を実行してから再実行が受けつけられるまでのインターバル時間（読み取り専用）

BOOL testMode テスト使用状態か否か（読み取り専用）

[メソッド]

+(instancetype)sharedPreference

アプリケーションで共有される本クラスのインスタンスを返す

[New ver. 1.3.0 ～]

+(instancetype)isValidPreference

アプリ用 SDK 設定ファイルが正しく記載されているかチェックするメソッドです。

※そのままコールすることもできますが、[TbBTPreliminary setUpWithCompletionHandler]内で実行され、不備があればエラーログを出力します。

5-5 TbBTServiceBeaconData クラス

[役割]

サービス用ビーコンの制御用情報を保持します

[プロパティ]

NSString regionID サービス提供上割り当てられているビーコンの ID (読み取り専用)

NSInteger segment サービス提供上割り当てられているビーコンのセグメント (読み取り専用)

NSString keycode ユーザによる排他使用のためビーコンに割り振られているキーコード文字列 (読み取り専用)

BOOL switcher 他のビーコンのモニタリングのオン・オフ制御用に使用する場合に、YES を指定するための使用オプション (デフォルト : NO)。

[メソッド]

- (instancetype)initWithRegionID:(NSString *)regionID

segment:(NSInteger)segmentValue keycode:(NSString *)keycode

渡されたセグメント値 segmentValue と、キーコード keycode で初期化します

- (void)setSwitcher:(BOOL)switcher

アクセサメソッド。このビーコンは、他のビーコンのモニタリングの制御用であることを示す、switcher フラグを立てます

- (BOOL)isSwitcher

アクセサメソッド。このビーコンは、他のビーコンのモニタリングの制御用であるか否かを示す、switcher フラグ値を返します

5 - 6 TbBTRegionNotificationSettingOptions クラス

[役割]

サービス専用のビーコン領域のモニタリング系処理を実行する場合に、アプリに通知されるイベントの設定

(CLRegion.notifyOnEntry/CLRegion.notifyOnExit/CLBeaconRegion.notifyEntryStateOnDisplay) を指定します

※ TbBTManager の関連するメソッド参照

※ CLRegion/CLBeaconRegion クラスのデフォルト設定で良い場合は使用する必要はありません

(デフォルト)

notifyOnEntry = YES, notifyOnExit = YES, notifyEntryStateOnDisplay = NO

[プロパティ]

NSNumber supportingTypes 指定されている設定区分のビットマスク値（読み取り専用）

[タイプ定義]

TbBTRegionNotificationType: モニタリングする領域の状態検知設定区分

- **TbBTRegionNotificationTypeOnEntry**
領域に入ったことを検知したら（アプリに）通知します
(CLRegion.notifyOnEntry)
- **TbBTRegionNotificationTypeOnExit**
領域から出たことを検知したら（アプリに）通知します
(CLRegion.notifyOnExit)
- **TbBTRegionNotificationTypeEntryStateOnDisplay**
領域内にいるときにディスプレイがオンになったタイミングで領域内にいることを（アプリに）通知します
(CLBeaconRegion.notifyEntryStateOnDisplay)

[メソッド]

+

(instancetype)settingWithTypes:(TbBTRegionNotificationType)supportTypes

「YES」にする区分を指定してインスタンスを生成します

supportTypes の指定は、YES にする区分のビットマスクです。

Ex.) *notifyOnEntry = YES, notifyOnExit = NO, notifyEntryStateOnDisplay = YES にしたい場合、*

[TbBTRegionNotificationSettingOptions

settingWithTypes:TbBTRegionNotificationTypeOnEntry / TbBTRegionNotificationTypeEntryStateOnDisplay]

5 - 7 TbBTBeaconizer クラス

[役割]

アプリがアクティブなときにスマートフォンからビーコン発信をする処理を実行します

※ 事前に Bluetooth の有効化のユーザ許可が必要です

[プロパティ]

id<TbBTBeaconizerDelegate> delegate TbBTBeaconizerDelegate プロトコルメソッドを実装するクラスのインスタンス 必須

[メソッド]

**+ (instancetype)instantiateStaticWithMajor:(NSUInteger)majorValue
Minor:(NSUInteger)minorValue**

10 進数の major 番号 (majorValue)、minor 番号(minorValue)を渡して、固定値ビーコン化します。ビーコン用発信 UUID はアプリ専用に管理サーバに登録されたものを使用します。

+ (instancetype)sharedBeaconizer

instantiate....メソッドで初期化されたシングルトンインスタンスを返します。

instantiate...が実行されていなければ、nil を返します。

- (void)tryToActivateAsBeacon

実行アプリ (の載ったスマートフォン) の擬似ビーコン発信を非同期で試行します。処理後、TbBTBeaconizerDelegate メソッドをコールバックします。

- **(void)resignActiveBeacon**

擬似ビーコン発信を停止します（非同期）。処理後、TbBTBeaconizerDelegate メソッドをコールバックします。

- **(CLBeaconRegion *)myBeaconRegion**

発信する/している自身の擬似ビーコン領域情報を返します。

- **(BOOL)isActivatable**

アプリ（スマートフォン）がビーコン発信可能か否かをチェックします。YES であれば、ビーコン発信が可能、NO が返る場合は、本 SDK によるビーコン発信機能が使えない状態です。

発信できない事由としては、スマートフォン端末や OS がビーコン発信に対応していない、ユーザによる Bluetooth 使用がオフになっている、等の状態があります。

- **(BOOL)isActive**

現在アプリからビーコン発信しているか否かを返します。

YES... 発信中 NO... 停止中

[デリゲートプロトコルメソッド (TbBTBeaconizerDelegate)]

(ビーコン発信開始処理結果 : cf. tryToActivateAsBeacon)

- **(void)didBeacomeActiveBeacon**

発信中の擬似ビーコンになった場合にコールバックされます。

- **(void)didBlockToBeacomeActiveBeaconWithReason:(NSString *)reason**

SDK 判断で擬似ビーコン化を明示的に中止した場合にコールバックされます。reason が制限した事由です。

(例 : ユーザによる Bluetooth 使用の許可がされていない場合)

- **(void)didFailToBeacomeActiveBeaconWithError:(NSString *)error**

error で示されるエラーにより、ビーコン発信ができなかった場合にコールバックされます。

(ビーコン発信停止処理結果 : cf. resignActivateBeacon)

- **(void)didResignActiveBeacon**

擬似ビーコンとしての発信を中断した場合にコールバックされます。

- **(void)didFailToResignActiveBeaconWithReason:(NSString *)reason**

reason で示される事由により、ビーコン発信を停止できなかった場合にコールバックされます。

[New ver. 1.3.0～]

5-8 TbBTOnetimeValidationClient クラス

[役割]

3bitter SWAMP プラットフォームベースの可変値ビーコン（タイプ 1 ワンタイム型）ソリューションで、可変値ビーコン（現行ではタイプ 1 ワンタイム型）の正当性のチェックを行います。

[プロパティ]

id<TbBTOnetimeValidationClientDelegate> delegate

TbBTOnetimeValidationClientDelegate プロトコルメソッドを実装するクラスのインスタンス 必須

[タイプ定義]

TbBTOnetimeValidationError: 可変値ビーコンの妥当性チェック結果エラー区分

- **TbBTOnetimeValidationErrorInvalidApp (= 2001)**
送信元のアプリが可変値ビーコンを使用するように登録されていません
- **TbBTOnetimeValidationErrorInvalidRegion (= 2002)**
送信されたビーコン情報は 3bitter 可変値ビーコンで定義するビーコン領域のものではありません
- **TbBTOnetimeValidationErrorInvalidBeacon (= 2003)**
送信されたビーコン情報は有効な 3bitter 可変値ビーコンではありません
- **TbBTOnetimeValidationErrorInternalServerError (= 2004)**
可変値ビーコンの正当性チェックをするサーバ内でエラーが発生しました

た

- ・ **TbBTONetimeValidationErrorInternetUnconnectable** (= 2005)

[メソッド]

- **(void)requestValidationForOnetimeBeacon:(CLBeacon *)beaconInfo**

チェックサーバに対し、 **beaconInfo** で指定するビーコンが正当な可変値ビーコンであるか否かのチェックをリクエストします (非同期: チェック完了後に、以下のデリゲートメソッドをコールバック)

[デリゲートプロトコルメソッド (TbBTONetimeValidationClientDelegate)]

- **(void)confirmedAsValidBeacon**

渡されたビーコンが正当な可変値ビーコンであると確認されました。

- **(void)decidedAsInvalidBeacon**

渡されたビーコンは有効な可変値ビーコンとしては不当と判定されました。

- **(void)didFailToValidateBeaconWithError:(NSError *)error**

リクエストされたビーコンのチェックに **error** のエラー発生により失敗しました。

error.code のエラーコードに、 *TbBTONetimeValidationError* の区分がエラー入ります。

6 端末言語設定対応

現時点で、ビルトインビューでの表記は日本語、英語を用意しています。
こちらを書き換えることで、ビルトインビューの表記カスタマイズが可能です

6-1 リソースファイル

[Supporting Files – Resources – i18n]

- `en.lproj/TbBTLocalizable.strings` 英語設定用の表示文字列定義
- `ja.lproj/TbBTLocalizable.strings` 日本語設定用の表示文字列定義