



# Bluetooth ライブラリ マニュアル

このマニュアルは、Bluetooth ライブラリの仕様について記載します。

#### **ご注意**

- このソフトウェアおよびマニュアルの、一部または全部を無断で使用、複製することはできません。
- このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。
- このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。
- このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なしに変更することがあります。
- このマニュアルの著作権はカシオ計算機株式会社に帰属します。
- 本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2018 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。

## 変更履歴

バージョン	変更日付	ページ	内容
1.00	2008.11		新規作成
1.01	2009.03	2	開発環境とプログラミング言語の対応表を訂正
		21	BTInquiry 関数の冗長な宣言を削除
		23	BTGetDeviceInfo 関数の冗長な宣言を削除
		23	BTGetDeviceInfo 関数のパラメータ説明を訂正
		25	BTGetServiceInfo 関数の解説内容を追加
		26	BTSelectDevice 関数の冗長な宣言を削除
		28	BTSetPassKey 関数の冗長な宣言を削除
		29	BTrustDevice 関数の冗長な宣言を削除
		37	BTSearchDeviceInfo 関数の冗長な宣言を削除
		41	BTSetDefaultDevice 関数の冗長な宣言を削除
48	BTSetAFHStatus 関数のチャンネル指定方法を追加		
1.02	2009.07	-	対象機種に DT-5300 を追加
1.03	2010.05	-	対象機種に DT-5300WM を追加
		73	シリアルプロファイル使用時の接続方法の手順を修正
1.04	2010.06	26	BTSelectDevice 関数のパラメータを追加
1.05	2011.01	-	対象機種に IT-300 および DT-X8 を追加
1.06	2011.11	-	対象機種に IT-9000 を追加
1.07	2013.09	86	各参照先マニュアル名を追加
		87	LAN プロファイル使用時の接続方法の手順を修正
1.08	2014.07	16	BTInitialize 関数の補足を追加
		17、51	BTWaitForBtReady 関数の解説内容を修正
1.09	2014.11	-	対象機種に IT-G500 を追加
		-	関数一覧表をデバイスライブラリ共通マニュアルへ移動
1.10	2015.02	-	対象機種に DT-X100 および DT-X200 を追加
1.11	2015.12	-	VisualBasic、C#の引数定義に名前空間を表すカシオ独自の記法(C:BluetoothLibNet+)を削除
		44	BTGetConnectionStatus 関数に補足説明を追加
1.12	2016.05	20	BTSetLocalInfo 関数に注意を追加
		53	BTSendSerialData 関数の誤記を修正
		55	BTReceiveSerialData 関数の誤記を修正
		77	ダイヤルアッププロファイルを修正
1.13	2016.06	-	変更履歴修正
1.14	2016.10	37	BTSearchDeviceInfo 関数の説明を追加
		79	「本体電源 ON 直後のエラー発生について」を追加
1.15	2016.12	16	BTInitialize 関数に注意を追加
		20	BTSetLocalInfo 関数に注意を追加
		35	BTRegisterLocalInfo 関数に注意を追加
1.16	2018.07	41	BTSetDefaultDevice 関数の説明を修正
		54	BTSendSerialData 関数の注意を追加

# 目次

1.	概要	1
2.	動作環境	2
3.	構造体	4
3.1	BTST_LOCALINFO	5
3.2	BTST_DEVICEINFO	6
4.	定数	7
4.1	デバイスモード	7
4.2	デバイスクラス	8
4.3	サービス UUID	11
4.4	エラーフラグ	13
5.	関数	16
5.1	BTInitialize	16
5.2	BTDeInitialize	17
5.3	BTGetLocalInfo	18
5.4	BTSetLocalInfo	19
5.5	BTInquiry	21
5.6	BTGetDeviceInfo	23
5.7	BTGetServiceInfo	25
5.8	BTSelectDevice	26
5.9	BTSetPassKey	28
5.10	BTTTrustDevice	29
5.11	BTSetWakeOnStatus	31
5.12	BTGetWakeOnStatus	32
5.13	BTGetDeviceHandle	33
5.14	BTGetLastError	34
5.15	BTRegisterLocalInfo	35
5.16	BTRegisterDeviceInfo	36
5.17	BTSearchDeviceInfo	37
5.18	BTDeleteDeviceInfo	39
5.19	BTGetDefaultDeviceInfo	40
5.20	BTSetDefaultDevice	41
5.21	BTGetLibraryStatus	43
5.22	BTGetDeviceName	44
5.23	BTGetConnectionStatus	45
5.24	BTSetConnectionParameter	46
5.25	BTGetConnectionParameter	47
5.26	BTSetAFHStatus	48
5.27	BTGetAFHStatus	50
5.28	BTWaitForBtReady	51
5.29	BTConnectSerial	52
5.30	BTSendSerialData	53
5.31	BTReceiveSerialData	55
5.32	BTDisconnectSerial	56
5.33	BTSetPANStatus	57
5.34	BTGetPANStatus	58

5.35	BTConnectPAN	59
5.36	BTDisconnectPAN	60
5.37	BTConnectHeadset	61
5.38	BTDisconnectHeadset	62
5.39	BTSetSoundPath	63
5.40	BTGetSoundPath	64
5.41	BTSetHeadsetGain	65
5.42	BTGetHeadsetGain	66
5.43	BTSetHeadsetServerStatus	67
5.44	BTGetHeadsetServerStatus	68
5.45	BTSetOBEXFolder	69
5.46	BTGetOBEXFolder	70
5.47	BTSendOBEXFile	71
6.	各プロファイルの Bluetooth 機器接続手順	72
6.1	通信する Bluetooth 機器情報の登録	72
6.2	シリアルプロファイル使用時の接続方法	73
6.3	ダイヤルアッププロファイル使用時の接続方法	75
6.4	LAN プロファイル使用時の接続方法	76
7.	プログラミング上の注意点	77
7.1	サポート通信プロファイルについて	77
7.2	Bluetooth の通信動作モードについて	78
7.3	本体電源 ON 直後のエラー発生について	79
8.	Device Emulator	80
8.1	BTInit.ini	80
8.2	BTDeviceInfo[n].ini	82
8.3	BTReg.ini	83

## 1. 概要

Bluetooth ライブラリは、本体に内蔵している Bluetooth モジュールを利用して、他の Bluetooth 機器との接続および通信を行う機能を提供します。

Bluetooth クラスライブラリは、Bluetooth ライブラリを .NET Compact Framework アプリケーションから直接利用できるようにする、ラッパーライブラリです。

Bluetooth ライブラリを使用することにより、機種を意識することなく、アプリケーションのソースコード互換性を高めることができます。

Bluetooth ライブラリでは、機種を問わず、すべての関数を用意し、アプリケーションから見た「仮想マシン」としての振る舞いを提供します。

Bluetooth ライブラリの各関数は、アプリケーションからの要求に対して、対象のデバイス機能が制御できない場合は、「未サポートエラー」を返します。また搭載デバイスの機能差によって利用できないパラメータを設定した場合は、「パラメータエラー」を返します。

※ Bluetooth ライブラリは、アプリケーションのソースコード互換性の向上を目的としたライブラリであり、搭載デバイスの機能互換性を保障するものではありません。

「未サポートエラー」および「パラメータエラー」を正しく判定し、操作者に対して機能が未サポートである旨を通知する、あるいは処理そのものを無効としてください。

## 2. 動作環境

Bluetooth ライブラリの動作環境を以下に示します。

### 対象機種

DT-5200 / DT-X7 / DT-9800 / DT-5300 / IT-300 / DT-X8 / IT-9000 / IT-G500 / DT-X100 / DT-X200

### 対象 OS

- Microsoft Windows CE 5.0
- Microsoft Windows CE 6.0
- Microsoft Windows Embedded Compact 7
- Microsoft Windows Mobile 6.5
- Microsoft Windows Mobile 6.5.3
- Microsoft Windows Embedded Handheld 6.5

### 開発環境とプログラミング言語

表 2-1

開発環境	Visual C++	Visual Basic, Visual C#
Microsoft embedded Visual C++ Version 4.0 + SP4	○	-
Microsoft Visual Studio.NET 2003 + SP1	×	○
Microsoft Visual Studio 2005 + SP1	○	○
Microsoft Visual Studio 2008	○	○

(○:利用可、×:利用不可、-:機能なし)

### 提供ファイル

表 2-2

ファイル	Visual C++	Visual Basic, Visual C#
BluetoothLib.h	○	-
BluetoothLib.lib	○	-
BluetoothLib.dll	○	○
BluetoothLibNet.dll (クラスライブラリ)	-	○

(○:必要、-:不要)

## 使用方法

### Visual C++ の場合

- プログラムソース内に BluetoothLib.h をインクルードし、リンカの依存ファイルとして BluetoothLib.lib を指定してください
- BluetoothLib.dll は本体に内蔵されています

### Visual Basic または Visual C# の場合

- BluetoothLibNet.dll をプロジェクトの参照に追加してください
- BluetoothLib.dll は本体に内蔵されています
- BluetoothLibNet.dll を実行モジュールと同じフォルダにコピーしてください

## 名前空間とクラス

クラスライブラリ BluetoothLibNet.dll では、関数および定数の参照用として、下記のクラスが用意されています。

表 2-3

名前空間	クラス名	内容
CaLib	BluetoothLibNet.Api	関数参照用クラス
	BluetoothLibNet.Def	定数参照用クラス

クラス定義の詳細については、Microsoft Visual Studio で BluetoothLibNet.dll を参照設定し、オブジェクトブラウザで確認してください。



### 3. 構造体

Bluetooth ライブラリで提供する構造体を以下に示します。

表 3-1 構造体一覧

構造体名	内容	使用する関数
BTST_LOCALINFO	本体の Bluetooth デバイス情報を格納する構造体	BTGetLocalInfo BTSetLocalInfo
BTST_DEVICEINFO	通信する Bluetooth デバイスの情報を格納する構造体	BTGetDeviceInfo BTGetServiceInfo BTSelectDevice BTTrustDevice BTRegisterDeviceInfo BTSearchDeviceInfo BTDeleteDeviceInfo BTGetDefaultDeviceInfo BTSetDefaultDevice

## 3.1 BTST\_LOCALINFO

本体の Bluetooth デバイス情報を格納します。

```
struct _btlocalinfo {
    TCHAR    LocalName[82];    Bluetooth 機器名
    TCHAR    LocalAddress[18]; Bluetooth 機器アドレス
    LONG     LocalDeviceMode; Bluetooth デバイスモード
    LONG     LocalClass1;     Bluetooth デバイスクラス 1 (ローカル)
    LONG     LocalClass2;     Bluetooth デバイスクラス 2 (ローカル)
    LONG     LocalClass3;     Bluetooth デバイスクラス 3 (ローカル)
    BOOL     Authentication;   Bluetooth 認証設定フラグ
    BOOL     Encryption;      暗号設定フラグ
} BTST_LOCALINFO;
```

BTST\_LOCALINFO 構造体を使用する関数は以下の通りです。

BTSetLocalInfo  
BTGetLocalInfo

## 3.2 BTST\_DEVICEINFO

通信する Bluetooth 機器のデバイス情報を格納します。

```
struct _btdeviceinfo {
    LONG      DeviceErrorFlag;    エラーフラグ
    HANDLE    DeviceHandle;      デバイスハンドル
    TCHAR     DeviceName[82];     Bluetooth 機器名
    TCHAR     DeviceAddress[18];  Bluetooth 機器アドレス
    LONG      DeviceClass1;       Bluetooth デバイスクラス 1 (通信先)
    LONG      DeviceClass2;       Bluetooth デバイスクラス 2 (通信先)
    LONG      DeviceClass3;       Bluetooth デバイスクラス 3 (通信先)
    DWORD     ProfileNumber;      使用可能なプロファイルの数
    WORD      ProfileUUID[16];    使用可能なプロファイルの種類
} BTST_DEVICEINFO;
```

BTST\_DEVICEINFO 構造体を使用する関数は以下の通りです。

- BTGetDeviceInfo
- BTGetServiceInfo
- BTSelectDevice
- BTTTrustDevice
- BTRegisterLocalInfo
- BTSearchDeviceInfo
- BTDeleteDeviceInfo
- BTGetDefaultDeviceInfo
- BTSetDefaultDevice

## 4. 定数

Bluetooth ライブラリで提供する定数を以下に示します。

表 4-1 定数一覧

定数名	内容
デバイスモード	他の Bluetooth デバイスに対してのアクセス許可
デバイスクラス	Bluetooth 機器の属性
サービス UUID	Bluetooth デバイスでの使用可能なサービス
エラーフラグ	エラー詳細情報

### 4.1 デバイスモード

デバイスモードは、本体の Bluetooth が他の Bluetooth デバイスに対してアクセス許可するかどうかを決めるパラメータです。以下のパラメータが使用できます。

表 4-2 デバイスモード一覧

設定値	設定内容	
BTMODE_NO_SCANS	他の Bluetooth デバイスからの Inquiry および接続を許可しない	○
BTMODE_INQUIRY_ENABLED	他の Bluetooth デバイスからの Inquiry のみ許可する	○
BTMODE_PAGE_ENABLED	他の Bluetooth デバイスからの接続のみ許可する	○
BTMODE_BOTH_ENABLED	他の Bluetooth デバイスからの Inquiry および接続を許可する	○
BTMODE_LIMITED_ACCESSIBLE	指定した相手からの接続のみ許可する	-

デバイスモードは、BTST\_LOCALINFO の LocalDeviceMode メンバを使用して、パラメータの取得および設定を行います。デフォルトの設定は、BTMODE\_BOTH\_ENABLED です。

## 4.2 デバイスクラス

デバイスクラスは、Bluetooth 機器がどのような属性を持っているかを示すパラメータです。デバイスクラスのパラメータは、メジャーサービスクラス、メジャーデバイスクラス、マイナーデバイスクラスの 3 種類があります。BTST\_LOCALINFO のメンバに以下の様に設定してください。

LocalClass1 ←       メジャーデバイスクラスを設定  
LocalClass2 ←       マイナーデバイスクラスを設定  
LocalClass3 ←       メジャーサービスクラスを設定

### ■メジャーデバイスクラス

メジャーデバイスクラスは、Bluetooth デバイスの大まかな機器属性を示すパラメータです。以下のパラメータのうち、どれか 1 種類が使用可能です。

BTCOD\_MAJOR\_MISC  
BTCOD\_MAJOR\_COMPUTER  
BTCOD\_MAJOR\_PHONE  
BTCOD\_MAJOR\_LAN\_ACCESS\_POINT  
BTCOD\_MAJOR\_AUDIO  
BTCOD\_MAJOR\_PERIPHERAL  
BTCOD\_MAJOR\_IMAGING  
BTCOD\_MAJOR\_UNCLASSIFIED

### ■マイナーデバイスクラス

マイナーデバイスクラスは、Bluetooth デバイスの詳細な機器属性を示すパラメータです。メジャーデバイスクラスごとに、使用可能なマイナーデバイスクラスが決められています。

以下のパラメータは、BTCOD\_MAJOR\_LAN\_ACCESS\_POINT を除く全てのメジャーデバイスクラスと組み合わせ使用します。

BTCOD\_MINOR\_UNCLASSIFIED

以下のパラメータは、BTCOD\_MAJOR\_COMPUTER デバイスクラスと組み合わせ使用します。

BTCOD\_COMPUTER\_DESKTOP  
BTCOD\_COMPUTER\_SERVER  
BTCOD\_COMPUTER\_LAPTOP  
BTCOD\_COMPUTER\_HANDHELD  
BTCOD\_COMPUTER\_PALM  
BTCOD\_COMPUTER\_WEARABLE

以下のパラメータは、BTCOD\_MAJOR\_PHONE デバイスクラスと組み合わせ使用します。

BTCOD\_PHONE\_CELLULAR  
BTCOD\_PHONE\_CORDLESS  
BTCOD\_PHONE\_SMART  
BTCOD\_PHONE\_MODEM

以下のパラメータは、BTCOD\_MAJOR\_LAN\_ACCESS\_POINT デバイスクラスと組み合わせて使用します。

- BTCOD\_LAP\_FULLY\_AVAILABLE
- BTCOD\_LAP\_USAGE\_1
- BTCOD\_LAP\_USAGE\_2
- BTCOD\_LAP\_USAGE\_3
- BTCOD\_LAP\_USAGE\_4
- BTCOD\_LAP\_USAGE\_5
- BTCOD\_LAP\_USAGE\_6
- BTCOD\_LAP\_NOT\_AVAILABLE

以下のパラメータは、BTCOD\_MAJOR\_AUDIO デバイスクラスと組み合わせて使用します。

- BTCOD\_AUDIO\_HEADSET\_PROFILE
- BTCOD\_AUDIO\_HANDS\_FREE
- BTCOD\_AUDIO\_RESERVED1
- BTCOD\_AUDIO\_MICROPHONE
- BTCOD\_AUDIO\_LOUDSPEAKER
- BTCOD\_AUDIO\_HEADPHONES
- BTCOD\_AUDIO\_PORTABLE\_AUDIO
- BTCOD\_AUDIO\_CAR\_AUDIO
- BTCOD\_AUDIO\_SET\_TOP\_BOX
- BTCOD\_AUDIO\_HI\_FI\_DEVICE
- BTCOD\_AUDIO\_VCR
- BTCOD\_AUDIO\_VIDEO\_CAMERA
- BTCOD\_AUDIO\_CAMCORDER
- BTCOD\_AUDIO\_VIDEO\_MONITOR
- BTCOD\_AUDIO\_DISPLAY\_LOUDSPEAKER
- BTCOD\_AUDIO\_VIDEO\_CONFERENCING
- BTCOD\_AUDIO\_RESERVED
- BTCOD\_AUDIO\_GAMING\_TOY

以下のパラメータは、BTCOD\_MAJOR\_PERIPHERAL デバイスクラスと組み合わせて使用します。

- BTCOD\_PP\_KEYBOARD
- BTCOD\_PP\_POINTING\_DEVICE
- BTCOD\_PP\_COMBO\_DEVICE

以下のパラメータは、BTCOD\_MAJOR\_PERIPHERAL デバイスクラスのマイナーデバイスクラスと組み合わせて使用します。

- BTCOD\_PP\_UNCATEGORIZED
- BTCOD\_PP\_JOYSTICK
- BTCOD\_PP\_GAMEPAD
- BTCOD\_PP\_REMOTE\_CONTROL
- BTCOD\_PP\_SENSING\_DEVICE

以下のパラメータは、BTCOD\_MAJOR\_IMAGING デバイスクラスと組み合わせて使用します。

BTCOD\_IMAGING\_RESERVED1  
BTCOD\_IMAGING\_RESERVED2  
BTCOD\_IMAGING\_DISPLAY  
BTCOD\_IMAGING\_CAMERA  
BTCOD\_IMAGING\_SCANNER  
BTCOD\_IMAGING\_PRINTER

### ■メジャーサービスクラス

メジャーサービスクラスは、Bluetooth デバイスのサービス属性を示すパラメータです。

以下のパラメータが使用可能です。各パラメータの論理和を取ることで、複数のメジャーサービスを設定することが可能です。

BTCOD\_LIMITED\_DISCOVERABLE  
BTCOD\_RESERVED\_1  
BTCOD\_RESERVED\_2  
BTCOD\_POSITIONING  
BTCOD\_NETWORKING  
BTCOD\_RENDERING  
BTCOD\_CAPTURING  
BTCOD\_OBJECT\_TRANSFER  
BTCOD\_AUDIO  
BTCOD\_TELEPHONY  
BTCOD\_INFORMATION

### ■デバイスクラスパラメータの取得・設定方法

本体のデバイスクラスを取得する場合は、BTGetLocalInfo 関数を実行した後、引数の

BTST\_LOCALINFO の LocalClass1, LocalClass2, LocalClass3 の各メンバを参照します。

他の Bluetooth デバイスのデバイスクラスを取得する場合は、BTGetDeviceInfo 関数を実行した後、引数の BTST\_DEVICEINFO の DeviceClass1, DeviceClass2, DeviceClass3 の各メンバを参照します。

本体のデバイスクラスを設定する場合は、BTST\_LOCALINFO の LocalClass メンバにデバイスクラスパラメータを設定後、BTSetLocalInfo 関数を実行します。

※デバイスクラスパラメータの詳細につきましては、Bluetooth Assigned Numbers を参照してください。

<http://www.bluetooth.org/foundry/assignednumb/document/baseband/>

## 4.3 サービス UUID

サービス UUID は、通信先の Bluetooth デバイスで使用可能なサービスを取得する際に使用されるパラメータです。以下のパラメータが使用できます。

表 4-3 サービス UUID 一覧

サービス UUID	
BTUUID_SDP_PROTOCOL	○
BTUUID_RFCOMM_PROTOCOL	○
BTUUID_TCS_PROTOCOL	○
BTUUID_CTP_PROTOCOL	○
BTUUID_L2CAP_PROTOCOL	○
BTUUID_IP_PROTOCOL	○
BTUUID_UDP_PROTOCOL	○
BTUUID_TCP_PROTOCOL	○
BTUUID_TCS_AT_PROTOCOL	○
BTUUID_OBEX_PROTOCOL	○
BTUUID_FTP_PROTOCOL	○
BTUUID_HTTP_PROTOCOL	○
BTUUID_WSP_PROTOCOL	○
BTUUID_BNEP_PROTOCOL	○
BTUUID_UPNP_PROTOCOL	○
BTUUID_HIDP_PROTOCOL	○
BTUUID_SERVICE_DISCOVERY_SERVER	○
BTUUID_BROWSE_GROUP_DESCRIPTOR	○
BTUUID_PUBLIC_BROWSE_ROOT	○
BTUUID_PUBLIC_BROWSE_GROUP	○
BTUUID_SERIAL_PORT	○
BTUUID_LAN_ACCESS_USING_PPP	○
BTUUID_DIALUP_NETWORKING	○
BTUUID_IR_MC_SYNC	○
BTUUID_OBEX_OBJECT_PUSH	○
BTUUID_OBEX_FILE_TRANSFER	○
BTUUID_IR_MC_SYNC_COMMAND	○
BTUUID_HEADSET	○
BTUUID_CORDLESS_TELEPHONY	○
BTUUID_INTERCOM	○
BTUUID_FAX	○
BTUUID_HEADSET_AUDIO_GATEWAY	○
BTUUID_WAP	○
BTUUID_WAP_CLIENT	○
BTUUID_PANU	○
BTUUID_NAP	○
BTUUID_GN	○
BTUUID_DIRECT_PRINTING	○
BTUUID_REFERENCE_PRINTING	○



BTUUID_IMAGING	○
BTUUID_IMAGING_RESPONDER	○
BTUUID_IMAGING_AUTOMATIC_ARCHIVE	○
BTUUID_IMAGING_REFERENCE_OBJECTS	○
BTUUID_HANDSFREE	○
BTUUID_HANDSFREE_AUDIO_GATEWAY	○
BTUUID_DIRECT_PRINTING_REFERENCE_OBJECTS	○
BTUUID_REFLECTED_UI	○
BTUUID_BASIC_PRINTING	○
BTUUID_PRINTING_STATUS	○
BTUUID_HUMAN_INTERFACE_DEVICE_SERVICE	○
BTUUID_HARDCOPY_CABLE_REPLACEMENT	○
BTUUID_CHR_PRINT	○
BTUUID_HCR_SCAN	○
BTUUID_COMMON_ISDN_ACCESS	○
BTUUID_VIDEO_CONFERENCING_GW	○
BTUUID_PNP_INFORMATION	○
BTUUID_GENERIC_NETWORKING	○
BTUUID_GENERIC_FILE_TRANSFER	○
BTUUID_GENERIC_AUDIO	○
BTUUID_GENERIC_TELEPHONY	○
BTUUID_CTP	○

サービスクラスは、BTGetServiceInfo 関数を実行したとき、BTST\_DEVICEINFO の ProfileUUID メンバに値が格納されます。

## 4.4 エラーフラグ

エラーフラグは、エラーカテゴリフラグとエラーステータスフラグの組み合わせとして、関数から戻ります。

### エラーカテゴリフラグ

エラーカテゴリフラグは、エラーが発生したカテゴリをあらわします。

```
BTERR_CAT_NO_CATEGORY
BTERR_CAT_UART
BTERR_CAT_OSIF
BTERR_CAT_L2HCI
BTERR_CAT_RFCOMM
BTERR_CAT_SDK
BTERR_CAT_LYM
BTERR_CAT_IPC_RPC
BTERR_CAT_OBEX
BTERR_CAT_BLUETOOTH
BTERR_CAT_WSADAPTER
BTERR_CAT_WINSOCK_2X
BTERR_CAT_BNEP
BTERR_CAT_WINDOWS_SYSTEM
BTERR_CAT_WINDOWS_REGISTRY
```

### エラーステータスフラグ

エラーステータスフラグは、エラーの状態をあらわします。

```
BTERR_SUCCESS
BTERR_INVALID_PARAMETER_1
BTERR_INVALID_PARAMETER_2
BTERR_INVALID_PARAMETER_3
BTERR_INVALID_PARAMETER_4
BTERR_INVALID_PARAMETER_5
BTERR_INVALID_PARAMETER_6
BTERR_INVALID_PARAMETER_7
BTERR_INVALID_PARAMETER_8
BTERR_INVALID_PARAMETER_9
BTERR_INVALID_PARAMETER_10
BTERR_INVALID_PARAMETER_11_OR_MORE
BTERR_FAILED
BTERR_PENDING
BTERR_NO_MEMORY
BTERR_INVALID_PARAMETER
BTERR_OPERATION_FAILED
BTERR_INVALID_HANDLE
BTERR_CONNECTION_CLOSED
BTERR_BUFFER_TOO_SMALL
BTERR_END_OF_LIST
BTERR_ALREADY_EXISTS
BTERR_NOT_FOUND
BTERR_OVERFLOW
```

BTERR\_TIMEOUT  
BTERR\_NOT\_IMPLEMENTED  
BTERR\_NO\_RESOURCES  
BTERR\_INVALID\_CONNECTION  
BTERR\_UNINITIALIZED  
BTERR\_UNLOADING  
BTERR\_NO\_SERVER  
BTERR\_INVALID\_STATE  
BTERR\_HW\_ERROR  
BTERR\_DOES\_NOT\_EXIST  
BTERR\_CONNECTION\_FAILED  
BTERR\_CONNECTION\_LOST  
BTERR\_EARLY\_RETURN  
BTERR\_CANCELLED\_BY\_USER  
BTERR\_UNAUTHORIZED  
BTERR\_INVALID\_CHANNEL  
BTERR\_CONFLICT  
BTERR\_COULD\_NOT\_WRITE\_TO\_FILE  
BTERR\_SHARE\_DOES\_NOT\_EXIST  
BTERR\_SCATTERNET  
BTERR\_PACKET\_DROPPED  
BTERR\_MALFORMED\_PACKET  
BTERR\_REDUNDANT  
BTERR\_COULD\_NOT\_OPEN\_FILE  
BTERR\_TCPIP\_NOT\_AVAILABLE  
BTERR\_INVALID\_CRITICAL\_SECTION  
BTERR\_BIND  
BTERR\_OPENING\_SOCKET  
BTERR\_MAXIMUM\_RECURSION  
BTERR\_NO\_MATCH  
BTERR\_PROTOCOL\_UNAVAILABLE  
BTERR\_VERSION  
BTERR\_VALUE\_NOT\_FOUND  
BTERR\_SET\_STRING\_VALUE  
BTERR\_SET\_UINT\_VALUE  
BTERR\_SET\_BIN\_VALUE  
BTERR\_UNK\_VALUE\_TYPE  
BTERR\_MALFORMED\_ADDRESS  
BTERR\_INVALID\_PORT  
BTERR\_INVALID\_UUID  
BTERR\_SERVICE\_DOES\_NOT\_EXIST  
BTERR\_OBJECT\_TYPE\_INVALID  
BTERR\_DEFAULT\_OBJECT\_NOT\_SET  
BTERR\_MALFORMED\_PROPERTY  
BTERR\_COULD\_NOT\_READ\_FILE  
BTERR\_FILE\_NOT\_FOUND  
BTERR\_DIRECTORY\_NOT\_FOUND  
BTERR\_CONNECTED  
BTERR\_MALFORMED\_PRINTABLE\_STRING  
BTERR\_MAX\_FILESIZE\_REACH

---

BTERR\_LIB\_INIT  
BTERR\_APP\_EXIST  
BTERR\_DEVICE\_LIST  
BTERR\_DEVICE\_ADDRESS  
BTERR\_POWER\_MODULE  
BTERR\_LIB\_REINIT  
BTERR\_REG\_OPEN  
BTERR\_REG\_WRITE  
BTERR\_REG\_READ  
BTERR\_REG\_DELETE  
BTERR\_REG\_NO\_DATA  
BTERR\_REG\_NOT\_FOUND

## 5. 関数

### 5.1 BTInitialize

Bluetooth プロトコルスタックを初期化します。

```
[C++]  
LONG BTInitialize()
```

```
[Visual Basic]  
Public Shared Function BTInitialize() As Int32
```

```
[C#]  
public static Int32 BTInitialize()
```

#### 解説

本体の Bluetooth デバイスの使用を開始します。Bluetooth 仮想 COM ポートも初期化します。他の Bluetooth 機器との接続および通信を行う前に、最初にこの関数を必ず実行してください。Device Emulator では、BTInit.ini ファイルで指定した動作をします。詳細は BTInit.ini を参照してください。

#### パラメータ

なし

#### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: 初期化に失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

#### 補足

DT-5200 / DT-X7 / DT-9800 / DT-5300CE では、BTDeInitialize 関数直後に本関数を呼び出す場合、3 秒以上の Wait が必要です。

#### 注意

BTRegisterLocalInfo 関数でデバイス情報 (BTST\_LOCALINFO 構造体) をレジストリに登録している状態で、本関数を実行するとシステムのデバイス名を BTST\_LOCALINFO 構造体の LocalName に変更します。

(DT-5200、DT-X7、DT-9800、DT-5300CE では、デバイス名を変更しません。)

## 5.2 BTDeInitialize

Bluetooth プロトコルスタックを開放します。

```
[C++]  
LONG BTDeInitialize()
```

```
[Visual Basic]  
Public Shared Function BTDeInitialize() As Int32
```

```
[C#]  
public static Int32 BTDeInitialize()
```

### 解説

本体の Bluetooth デバイスの使用を終了します。Bluetooth 仮想 COM ポートも使用できなくなります。他の Bluetooth 機器との通信終了後、最後にこの関数を必ず実行してください。Device Emulator では、BTSetPassKey 関数で設定した PassKey の削除も行います。

### パラメータ

なし

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: リソース解放に失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

本関数を実行しないでユーザアプリケーションを終了した場合、次回 BTInitialize 関数を実行したときにエラーが発生して、本体の Bluetooth デバイスが使用できなくなる可能性があります。

## 5.3 BTGetLocalInfo

Bluetooth デバイス情報を取得します。

```
[C++]
LONG BTGetLocalInfo(
    BTST_LOCALINFO *LocalInfo
)
```

```
[Visual Basic]
Public Shared Function BTGetLocalInfo( _
    ByVal LocalInfo As BTST_LOCALINFO _
) As Int32
```

```
[C#]
public static Int32 BTGetLocalInfo(
    BTST_LOCALINFO LocalInfo);
```

### 解説

本体の Bluetooth デバイス情報を取得します。本関数では、BTST\_LOCALINFO の全てのメンバを取得します。Bluetooth デバイスモードおよび Bluetooth デバイスクラスにおいて取得可能なパラメータについては、「定数」を参照してください。

Device Emulator では、BTInit.ini ファイルの情報を読み込みます。詳細は BTInit.ini を参照してください。

### パラメータ

#### *LocalInfo*

Bluetooth デバイス情報構造体を指定します。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報取得失敗
BTERR_DRIVER	: ドライバーエラー
	DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.4 BTSetLocalInfo

Bluetooth デバイス情報を設定します。

```
[C++]
LONG BTSetLocalInfo(
    BTST_LOCALINFO *LocalInfo
)
```

```
[Visual Basic]
Public Shared Function BTSetLocalInfo( _
    ByVal LocalInfo As BTST_LOCALINFO _
) As Int32
```

```
[C#]
public static Int32 BTSetLocalInfo(
    BTST_LOCALINFO LocalInfo
);
```

### 解説

本体の Bluetooth デバイス情報を設定します。本関数を実行する前に、BTGetLocalInfo 関数を実行して、現在の Bluetooth デバイス情報を取得してください。

本関数では、BTST\_LOCALINFO のメンバのうち、Bluetooth アドレスを除く全てのデバイス情報が設定されます。Bluetooth デバイスマードおよび Bluetooth デバイスクラスにおいて設定可能なパラメータについては、「定数」を参照してください。

Device Emulator では、BTInit.ini ファイルに情報を書き込みます。詳細は BTInit.ini を参照してください。

### パラメータ

#### LocalInfo

設定する Bluetooth デバイス情報構造体を指定します。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報取得失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

Bluetooth デバイス情報のうち、Bluetooth 機器アドレスは変更しないでください。Bluetooth 機器アドレスを変更して本関数を実行すると、エラーとなります。

暗号設定フラグは、Bluetooth 認証フラグが有効の場合のみ、設定が可能です。Bluetooth 認証フラグが無効の場合は、暗号設定フラグを無効に設定してください。



## 注意

コントロールパネルのシステムのデバイス名とBTST\_LOCALINFO構造体のLocalNameは共通のため、本関数を実行するとシステムのデバイス名をBTST\_LOCALINFO構造体のLocalNameに変更します。  
(DT-5200、DT-X7、DT-9800、DT-5300CEでは、デバイス名を変更しません。)

## 5.5 BTInquiry

Bluetooth 機器の Inquiry(問い合わせ)を実行します。

```
[C++]
LONG BTInquiry(
    HANDLE *DeviceHandle,
    DWORD *DeviceNumber,
    DWORD InquiryTime
)
```

```
[Visual Basic]
Overloads Public Shared Function BTInquiry( _
    ByVal DeviceHandle As Int32(), _
    ByRef DeviceNumber As Int32, _
    ByVal InquiryTime As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTInquiry(
    Int32[] DeviceHandle,
    ref Int32 DeviceNumber,
    Int32 InquiryTime
);
```

### 解説

Bluetooth 機器の Inquiry(問い合わせ)を実行します。  
Device Emulator では、BTDeviceInfo[n].ini ファイルを検索し、そのファイル数を取得します。

### パラメータ

#### *DeviceHandle*

常に NULL (C++ の場合) または IntPtr.Zero (Visual Basic、C# の場合) を指定してください。

#### *DeviceNumber*

Inquiry で発見された Bluetooth 機器の数が格納されます。

#### *InquiryTime*

Inquiry を実行する時間を、ミリ秒単位で指定します。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: Bluetooth 機器の Inquiry 失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

本関数を実行した後、BTGetDeviceInfo 関数を実行して、Bluetooth 機器のデバイス情報を取得して

---

ください。

## 5.6 BTGetDeviceInfo

Bluetooth 機器のデバイス情報を取得します。

```
[C++]
LONG BTGetDeviceInfo(
    BTST_DEVICEINFO *DeviceInfo,
    DWORD DeviceNumber,
    HANDLE *DeviceHandle
)
```

```
[Visual Basic]
Overloads Public Shared Function BTGetDeviceInfo( _
    ByVal DeviceInfo As BTST_DEVICEINFO(), _
    ByVal DeviceNumber As Int32, _
    ByVal DeviceHandle As Int32() _
) As Int32
```

```
[C#]
public static Int32 BTGetDeviceInfo(
    BTST_DEVICEINFO[] DeviceInfo,
    Int32 DeviceNumber,
    Int32[] DeviceHandle
);
```

### 解説

BTInquiry 関数で発見された Bluetooth 機器のデバイス情報を取得します。BTInquiry 関数を実行後、必ず本関数を実行して Bluetooth 機器情報を取得してください。

Device Emulator では、BTDeviceInfo[n].ini ファイルの情報を読み込みます。詳細は BTDeviceInfo[n].ini を参照してください。

本関数で取得できるデバイス情報は、以下の通りです。

- 機器名
- 機器アドレス
- デバイスクラス

## パラメータ

### *DeviceInfo*

デバイス情報を格納する構造体を、配列で指定します。

構造体は、BTInquiry 関数で取得した Bluetooth 機器の数だけ用意してください。

### *DeviceNumber*

デバイス情報を格納する配列 DeviceInfo の構造体要素数。

通常は、BTInquiry 関数で取得した Bluetooth 機器の数を指定します。

### *DeviceHandle*

常に NULL (C++の場合) または IntPtr.Zero (Visual Basic、C#の場合) を指定してください。

## 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: Bluetooth 機器情報取得失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 補足

Bluetooth 機器名が取得できなかった場合、Bluetooth 機器名の変数に Bluetooth 機器アドレスを格納します。取得した Bluetooth 機器情報のうち、Bluetooth 機器名以外の情報は書き換えないでください。Bluetooth 機器との接続および通信ができなくなる可能性があります。

本関数を実行すると、Bluetooth 機器のサービス情報が初期化されます。Bluetooth 機器のサービス情報を取得する場合は、本関数を実行後、BTGetServiceInfo 関数を実行してください。

## 5.7 BTGetServiceInfo

Bluetooth 機器のサービス情報を取得します。

```
[C++]
LONG BTGetServiceInfo(
    BTST_DEVICEINFO *DeviceInfo
)
```

```
[Visual Basic]
Public Shared Function BTGetServiceInfo( _
    ByVal DeviceInfo As BTST_DEVICEINFO _
) As Int32
```

```
[C#]
public static Int32 BTGetServiceInfo(
    BTST_DEVICEINFO DeviceInfo
);
```

### 解説

通信する Bluetooth 機器のサービス情報を取得します。

本関数は、BTGetDeviceInfo 関数で取得した Bluetooth 機器のデバイス情報に加え、使用可能なプロファイルの種類 (BTST\_DEVICEINFO 構造体の ProfileUUID) を取得します。取得可能なサービスクラスパラメータについては、「定数」を参照してください。

本関数を実行する前に、BTGetDeviceInfo 関数を実行して、サービス情報以外のデバイス情報をあらかじめ取得してください。

Device Emulator では、BTDeviceInfo[n].ini ファイルの情報を読み込みます。詳細は BTDeviceInfo[n].ini を参照してください。

### パラメータ

#### *DeviceInfo*

Bluetooth 機器情報構造体を指定します。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: Bluetooth 機器情報取得失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

本関数を実行しなくても、Bluetooth 機器との接続を行うことは可能です。サポートしている通信プロファイルの種類をチェックする場合に使用してください。Bluetooth 機器によっては、SDP(Service Discovery Profile) をサポートしない等の理由で、使用可能なプロファイルの情報を取得できない場合があります。

本関数を実行後、再び BTGetDeviceInfo 関数を実行すると、引数に格納された Bluetooth サービス情報は消去されます。

## 5.8 BTSelectDevice

Bluetooth 機器を指定します。

```
[C++]
LONG BTSelectDevice(
    BTST_DEVICEINFO *DeviceInfo,
    LPTSTR PortName
)
```

```
[Visual Basic]
Overloads Public Shared Function BTSelectDevice( _
    ByVal DeviceInfo As BTST_DEVICEINFO, _
    ByVal PortName As String _
) As Int32
```

```
[C#]
public static Int32 BTSelectDevice(
    BTST_DEVICEINFO DeviceInfo,
    string PortName
);
```

### 解説

接続する Bluetooth 機器を指定します。Bluetooth 通信を開始する前にこの関数を実行して、接続する Bluetooth 機器を選択してください。

### パラメータ

#### *DeviceInfo*

接続する Bluetooth 機器の情報を指定します。

デフォルトの Bluetooth 機器を指定する場合は、NULL (C++ の場合) または `IntPtr.Zero` (Visual Basic、C# の場合) を指定します。

#### *PortName*

接続する通信ポートを、以下の値で指定します。

BTPORT_SERIAL	: 仮想シリアル
BTPORT_DIALUP	: ダイヤルアップ
BTPORT_PAN	: Bluetooth PAN
BTPORT_OBEX	: OBEX Object Push
BTPORT_HEADSET	: ヘッドセット

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: Bluetooth 機器指定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

---

## 補足

この関数を実行しない場合、`BTSetDefaultDevice` 関数で設定したデフォルトの Bluetooth 機器が、接続する Bluetooth 機器として指定されます。



## 5.9 BTSetPassKey

PassKey を設定します。

```
[C++]
LONG BTSetPassKey (
    LPTSTR PassKey
)
```

```
[Visual Basic]
Overloads Public Shared Function BTSetPassKey ( _
    ByVal PassKey As String _
) As Int32
```

```
[C#]
public static Int32 BTSetPassKey (
    string PassKey
);
```

### 解説

本体の PassKey を設定します。PassKey は、本体で他の Bluetooth 機器とのボンドを実行する場合、および他の Bluetooth 機器から PassKey 要求があったときに使用されます。

パラメータの設定により、他の Bluetooth 機器からの PassKey 要求を拒否することが可能です。設定した PassKey は、BTSetPassKey 関数および BTDeInitialize 関数を実行するまで有効です。

Device Emulator では、BTDeviceInfo[n].ini ファイルに情報を書き込みます。

### パラメータ

#### *PassKey*

設定するパスキーを指定します。

NULL (C++ の場合) または IntPtr.Zero (Visual Basic、C# の場合) を指定した場合、他の Bluetooth 機器からの PassKey 要求を拒否します。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: PassKey の設定に失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

BTTrustDevice 関数を実行する前に、必ず本関数を実行してください。他の Bluetooth 機器から HT へ PassKey 要求が必要となる場合は、事前に本関数を実行して PassKey を設定してください。

## 5.10 BTTrustDevice

Bluetooth 機器との信頼を実行します。

```
[C++]
LONG BTTrustDevice(
    BTST_DEVICEINFO *DeviceInfo,
    LPTSTR PortName
)
```

```
[Visual Basic]
Overloads Public Shared Function BTTrustDevice( _
    ByVal DeviceInfo As BTST_DEVICEINFO, _
    ByVal PortName As String _
) As Int32
```

```
[C#]
public static Int32 BTTrustDevice(
    BTST_DEVICEINFO DeviceInfo,
    string PortName
);
```

### 解説

Bluetooth 機器との信頼を実行します。  
本体から、指定した Bluetooth 機器との信頼を行います。

### パラメータ

#### *DeviceInfo*

デバイスの信頼を実行する Bluetooth 機器の情報を指定します。  
NULL (C++ の場合) または `IntPtr.Zero` (Visual Basic、C# の場合) を指定した場合、デフォルトの Bluetooth 機器とのデバイスの信頼を実行します。

#### *PortName*

デフォルトで通信する設定をした通信ポートを指定します。 `DeviceInfo` で NULL を指定した場合のみ、以下の中から一つを指定します。

`BTPORT_SERIAL` : 仮想シリアル  
`BTPORT_DIALUP` : ダイヤルアップ

### 戻り値

`BTERR_SUCCESS` : 正常終了  
`BTERR_FAILED` : Bluetooth 機器とのボンドに失敗  
`BTERR_DRIVER` : ドライバーエラー  
DeviceEmulator では発生しません  
`FUNCTION_UNSUPPORTED` : 未サポートエラー

### 補足

関数を実行する前に、`BTSetPassKey` 関数を必ず実行して、ボンド実行時に使用される `PassKey` を設定しておいてください。 `PassKey` を設定していない場合、または `BTSetPassKey` 関数でパラメータに

---

NULL または空文字列("")を指定した場合、本関数はエラーを返します。  
通信先の Bluetooth 機器からボンド要求があった場合、BTSetPassKey で設定した PassKey を自動的に送信するので、本関数を実行する必要はありません。

## 5.11 BTSetWakeOnStatus

Bluetooth WakeOn 機能を設定します。

```
[C++]
LONG BTSetWakeOnStatus(
    DWORD WakeOnStatus
)
```

```
[Visual Basic]
Public Shared Function BTSetWakeOnStatus( _
    ByVal WakeOnStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTSetWakeOnStatus(
    Int32 WakeOnStatus
);
```

### 解説

本体の Bluetooth WakeOn 機能を設定します。

### パラメータ

#### *WakeOnStatus*

本体の Bluetooth WakeOn を以下の値で指定します。

- BTWAKEON\_ENABLE : Bluetooth WakeOn 設定を有効にする
- BTWAKEON\_DISABLE : Bluetooth WakeOn 設定を無効にする

### 戻り値

- BTERR\_SUCCESS : 正常終了
- BTERR\_FAILED : デバイス情報設定失敗
- BTERR\_DRIVER : ドライバーエラー  
DeviceEmulator では発生しません
- FUNCTION\_UNSUPPORTED : 未サポートエラー

### 補足

Bluetooth WakeOn 設定は、本体をサスペンドする前に実行してください。

## 5.12 BTGetWakeOnStatus

Bluetooth WakeOn 設定を取得します。

```
[C++]
LONG BTGetWakeOnStatus(
    DWORD *WakeOnStatus
)
```

```
[Visual Basic]
Public Shared Function BTGetWakeOnStatus( _
    ByRef WakeOnStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTGetWakeOnStatus(
    ref Int32 WakeOnStatus
);
```

### 解説

本体の Bluetooth WakeOn 設定を取得します。

Device Emulator では、BTInit.ini ファイルの情報を読み込みます。詳細は BTInit.ini を参照してください。

### パラメータ

#### *WakeOnStatus*

本体の Bluetooth Wake On 設定を取得します。取得する値の詳細は BTSetWakeOnStatus 関数を参照してください。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報取得失敗
BTERR_DRIVER	: ドライバーエラー
	DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.13 BTGetDeviceHandle

Bluetooth 機器のデバイスハンドルを取得します。

```
[C++]
LONG BTGetDeviceHandle(
    HANDLE *DeviceHandle,
    LPTSTR BTAddress
)
```

```
[Visual Basic]
Public Shared Function BTGetDeviceHandle( _
    ByRef DeviceHandle As Int32, _
    ByVal BTAddress As String _
) As Int32
```

```
[C#]
public static Int32 BTGetDeviceHandle(
    ref Int32 DeviceHandle,
    string BTAddress
);
```

### 解説

Bluetooth アドレスを指定して、Bluetooth 機器のデバイスハンドルを取得します。

### パラメータ

#### *DeviceHandle*

Bluetooth 機器のデバイスハンドルを取得します。

#### *BTAddress*

Bluetooth アドレスを指定します。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: Bluetooth 機器のデバイスハンドル取得失敗
BTERR_DRIVER	: ドライバーエラー
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

Bluetooth 機器のデバイスハンドルを取得後、必ず `BTGetDeviceInfo` 関数を実行して、Bluetooth 機器のデバイス情報を取得してください。指定した Bluetooth アドレスを持つ Bluetooth 機器のデバイスハンドルが、`BTGetDeviceHandle` 関数または `BTInquiry` 関数によってすでに取得されている場合、本関数を実行するとエラーとなります。すでに取得したデバイスハンドルを使用してください。

## 5.14 BTGetLastError

エラー詳細情報を取得します。

```
[C++]  
LONG BTGetLastError ()
```

```
[Visual Basic]  
Public Shared Function BTGetLastError () As Int32
```

```
[C#]  
public static Int32 BTGetLastError ()
```

### 解説

Bluetooth ライブラリ関数呼び出しで発生したエラー詳細を取得します。  
この関数で取得したエラー詳細は機種により異なります。このため、システムライブラリの  
SysGetModelName 関数で機種を確認する必要があります。

### パラメータ

なし

### 戻り値

エラーコード詳細を返します。詳細は「エラーフラグ」を参照してください。  
または

FUNCTION\_UNSUPPORTED : 未サポートエラー

## 5.15 BTRegisterLocalInfo

Bluetooth デバイス情報をレジストリに登録します。

```
[C++]  
LONG BTRegisterLocalInfo()
```

```
[Visual Basic]  
Public Shared Function BTRegisterLocalInfo() As Int32
```

```
[C#]  
public static Int32 BTRegisterLocalInfo()
```

### 解説

本体の Bluetooth デバイス情報をレジストリに登録します。登録したデバイス情報は、次回 **BTInitialize** 関数を実行したときに再設定されます。

Device Emulator では、BTReg.ini ファイルに情報を書き込みます。

### パラメータ

なし

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報登録失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

Bluetooth デバイス情報を変更してレジストリに登録する場合は、本関数を実行する前に **BTSetLocalInfo** 関数を実行して、Bluetooth デバイス情報を変更してください。

### 注意

本関数でデバイス情報 (BTST\_LOCALINFO 構造体) をレジストリに登録している状態で、**BTInitialize** 関数を実行するとシステムのデバイス名を **BTST\_LOCALINFO** 構造体の **LocalName** に変更します。(DT-5200、DT-X7、DT-9800、DT-5300CE では、デバイス名を変更しません。)



## 5.16 BTRegisterDeviceInfo

Bluetooth 機器の情報をレジストリに登録します。

```
[C++]
LONG BTRegisterDeviceInfo(
    BTST_DEVICEINFO *DeviceInfo
)
```

```
[Visual Basic]
Public Shared Function BTRegisterDeviceInfo( _
    ByVal DeviceInfo As BTST_DEVICEINFO _
) As Int32
```

```
[C#]
public static Int32 BTRegisterDeviceInfo(
    BTST_DEVICEINFO DeviceInfo
);
```

### 解説

Bluetooth 機器の情報をレジストリに登録します。登録する Bluetooth 機器情報の Bluetooth アドレスがすでにレジストリに登録されている場合、以前登録した Bluetooth 機器情報は上書きされます。Device Emulator では、BTReg.ini ファイルに情報を書き込みます。

### パラメータ

#### *DeviceInfo*

登録する Bluetooth 機器の情報を指定します。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: レジストリへの登録失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.17 BTSearchDeviceInfo

Bluetooth 機器情報をレジストリから取得します。

```
[C++]
LONG BTSearchDeviceInfo(
    BTST_DEVICEINFO *DeviceInfo,
    DWORD *DeviceNumber,
    LPTSTR SearchKey
)
```

```
[Visual Basic]
Overloads Public Shared Function BTSearchDeviceInfo( _
    ByVal DeviceInfo As BTST_DEVICEINFO(), _
    ByRef DeviceNumber As Int32, _
    ByVal SearchKey As String _
) As Int32
```

```
[C#]
public static Int32 BTSearchDeviceInfo(
    BTST_DEVICEINFO[] DeviceInfo,
    ref Int32 DeviceNumber,
    string SearchKey
);
```

### 解説

検索キーを指定して、Bluetooth 機器の情報をレジストリから検索して取得します。検索キーと一致する Bluetooth 機器情報の数のみを取得することも可能です。

Device Emulator では、BTReg.ini ファイルの情報を読み込みます。詳細は BTReg.ini を参照してください。

### パラメータ

#### *DeviceInfo*

キーと一致した Bluetooth 機器の情報を指定します。

DeviceNumber で指定した数以上の配列を用意します。(配列はゼロクリアしてください)

NULL (C++ の場合) または IntPtr.Zero (Visual Basic、C# の場合) を指定した場合、キーと一致した Bluetooth 機器の数のみを返します。

#### *DeviceNumber*

取得する Bluetooth 機器情報の最大値を指定します。実行後、キーと一致した Bluetooth 機器の数が格納されます。

#### *SearchKey*

Bluetooth 機器情報検索キーを指定します。Bluetooth アドレスまたは Bluetooth 機器名を指定します。NULL (C++ の場合) または IntPtr.Zero (Visual Basic、C# の場合) を指定した場合、レジストリに登録されている全ての Bluetooth 機器の情報を返します。

## 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: レジストリからの検索失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 補足

Bluetooth機器情報を格納する構造体の変数は、配列として用意してください。DeviceNumberで小さい値を指定すると、キーと一致した Bluetooth 機器情報が完全には取得できない場合があります。キーと一致すると想定される Bluetooth デバイスの個数よりも大きい値を指定してください。検索キーと一致する Bluetooth 機器情報の数が予想できない場合には、最初に検索キーと一致する Bluetooth 機器情報の数のみを取得してください。その後、Bluetooth 機器情報を格納する構造体の変数を動的に確保し、再度同じ検索キーを使用して Bluetooth 機器情報を取得してください。

## 5.18 BTDeleteDeviceInfo

Bluetooth 機器情報をレジストリから削除します。

```
[C++]
LONG BTDeleteDeviceInfo(
    BT_DEVICEINFO *DeviceInfo
)
```

```
[Visual Basic]
Public Shared Function BTDeleteDeviceInfo( _
    ByVal DeviceInfo As BTST_DEVICEINFO _
) As Int32
```

```
[C#]
public static Int32 BTDeleteDeviceInfo(
    BTST_DEVICEINFO DeviceInfo
);
```

### 解説

指定した Bluetooth 機器の情報をレジストリから削除します。

### パラメータ

#### *DeviceInfo*

削除する Bluetooth 機器の情報を格納する構造体を指定します。  
Device Emulator では、BTReg.ini ファイルに情報を書き込みます。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: レジストリからの削除失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

指定する Bluetooth 機器情報は、BTSearchDeviceInfo 関数を使用して取得してください。引数として使用する Bluetooth 機器情報が、レジストリに登録されている Bluetooth 機器情報と一致しない場合、エラーとなります。

## 5.19 BTGetDefaultDeviceInfo

デフォルトで通信する Bluetooth 機器の情報を取得します。

```
[C++]
LONG BTGetDefaultDeviceInfo(
    BTST_DEVICEINFO *DeviceInfo,
    LPTSTR PortName
)
```

```
[Visual Basic]
Public Shared Function BTGetDefaultDeviceInfo( _
    ByVal DeviceInfo As BTST_DEVICEINFO, _
    ByVal PortName As String _
) As Int32
```

```
[C#]
public static Int32 BTGetDefaultDeviceInfo(
    BTST_DEVICEINFO DeviceInfo,
    string PortName
);
```

### 解説

デフォルトで通信する Bluetooth 機器の情報を取得します。

### パラメータ

#### *DeviceInfo*

デフォルトの Bluetooth 機器の情報を取得する構造体を指定します。

#### *PortName*

デフォルトの Bluetooth 機器と通信するときの通信ポートを、以下の値で指定します。

BTPORT\_SERIAL : 仮想シリアル  
BTPORT\_DIALUP : ダイヤルアップ

### 戻り値

BTERR\_SUCCESS : 正常終了  
BTERR\_FAILED : デフォルトで通信する Bluetooth 機器情報の取得失敗  
BTERR\_DRIVER : ドライバーエラー  
DeviceEmulator では発生しません  
FUNCTION\_UNSUPPORTED : 未サポートエラー

### 補足

デフォルトで通信する Bluetooth 機器を設定していない場合、エラーとなります。

## 5.20 BTSetDefaultDevice

デフォルトで通信する Bluetooth 機器を設定します。

```
[C++]
LONG BTSetDefaultDevice(
    BTST_DEVICEINFO *DeviceInfo,
    LPTSTR PortName
)
```

```
[Visual Basic]
Overloads Public Shared Function BTSetDefaultDevice( _
    ByVal DeviceInfo As BTST_DEVICEINFO, _
    ByVal PortName As String _
) As Int32
```

```
[C#]
public static Int32 BTSetDefaultDevice(
    BTST_DEVICEINFO DeviceInfo,
    string PortName
);
```

### 解説

デフォルトで通信する Bluetooth 機器を設定します。本関数で設定したデフォルトの Bluetooth 機器は、次回 `BTInitialize` 関数を実行したときに有効となります。このため、この関数を一度呼び出してデフォルトで通信する機器を指定すると、次の通信から `BTSearchDeviceInfo` 関数、`BTSelectDevice` 関数で接続相手を指定する必要がなくなります。

`Device Emulator` では、設定値を内部変数として格納するため、何も動作しませんが、`BTGetDefaultDeviceInfo` 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *DeviceInfo*

デフォルトに設定する Bluetooth 機器の情報を格納する構造体を指定します。

`NULL` (C++ の場合) または `IntPtr.Zero` (Visual Basic、C# の場合) を指定した場合、デフォルトで通信するデバイスを設定しません、

#### *PortName*

デフォルトの Bluetooth 機器と通信するときの通信ポートを、以下の値で指定します。

`BTPORT_SERIAL` : 仮想シリアル  
`BTPORT_DIALUP` : ダイアルアップ

### 戻り値

`BTERR_SUCCESS` : 正常終了  
`BTERR_FAILED` : デフォルトで通信する Bluetooth 機器情報の設定失敗  
`BTERR_DRIVER` : ドライバーエラー  
`DeviceEmulator` では発生しません  
`FUNCTION_UNSUPPORTED` : 未サポートエラー

## 補足

本関数を実行しても、**BTSelectDevice** 関数を使用して接続先として指定した **Bluetooth** 機器は変更されません。関数を実行する前に、**BTRegisterDeviceInfo** 関数を実行して、**Bluetooth** 機器情報をレジストリに登録しておく必要があります。

関数で使用する **Bluetooth** 機器情報は、**BTSearchDeviceInfo** 関数を使用して取得してください。

**Bluetooth** 機器情報がレジストリに登録されている **Bluetooth** 機器情報と一致しない場合、エラーとなります。

## 5.21 BTGetLibraryStatus

Bluetooth ライブラリの現在の状態を取得します。

```
[C++]
LONG BTGetLibraryStatus (
    DWORD *LibraryStatus
)
```

```
[Visual Basic]
Public Shared Function BTGetLibraryStatus( _
    ByRef LibraryStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTGetLibraryStatus(
    ref Int32 LibraryStatus
);
```

### 解説

Bluetooth ライブラリの現在の状態を取得します。

本関数は、BTInitialize 関数を実行していない状態でも、実行することが可能です。

### パラメータ

#### *LibraryStatus*

Bluetooth ライブラリの現在の状態を取得する変数

BTSTATUS_NOT_INITIALIZED	: 初期化されていない
BTSTATUS_INITIALIZED	: 初期化が完了している
BTSTATUS_REINITIALIZING	: 再初期化処理を実行中 DeviceEmulator では発生しません
BTSTATUS_REINIT_FAILED	: 再初期化に失敗した DeviceEmulator では発生しません

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー



## 5.22 BTGetDeviceName

Bluetooth 機器の Bluetooth アドレスを直接指定して、Bluetooth 機器名を取得します。

```
[C++]
LONG BTGetDeviceName(
    BTST_DEVICEINFO *DeviceInfo,
    LPTSTR DeviceAddress
)
```

```
[Visual Basic]
Public Shared Function BTGetDeviceName( _
    ByVal DeviceInfo As BTST_DEVICEINFO, _
    ByVal DeviceAddress As String _
) As Int32
```

```
[C#]
public static Int32 BTGetDeviceName(
    BTST_DEVICEINFO DeviceInfo,
    string DeviceAddress
);
```

### 解説

Bluetooth アドレスが既知の場合、BTInquiry 関数を実行せずに、本関数を実行して Bluetooth 機器名を取得することができます。

Device Emulator では、BTDeviceInfo[n].ini ファイルの情報を読み込みます。詳細は BTDeviceInfo[n].ini を参照してください。

### パラメータ

#### *DeviceInfo*

Bluetooth 機器名を格納する構造体変数

#### *DeviceAddress*

機器名を取得する Bluetooth 機器アドレスを指定します。(例:"00:80:37:17:78:DA")

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報取得失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.23 BTGetConnectionStatus

他の Bluetooth 機器との接続状態を取得します。

```
[C++]
LONG BTGetConnectionStatus(
    DWORD *ConnectionStatus
)
```

```
[Visual Basic]
Public Shared Function BTGetConnectionStatus( _
    ByRef ConnectionStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTGetConnectionStatus(
    ref Int32 ConnectionStatus
);
```

### 解説

他の Bluetooth 機器との接続状態を取得します。

### パラメータ

#### *ConnectionStatus*

他の Bluetooth 機器との接続状態を取得する変数

BTCONNECT_NO_CONNECTION	: 接続していない
BTCONNECT_SERIAL_CLIENT	: 仮想シリアルで接続
BTCONNECT_SERIAL_SERVER	: 仮想シリアルで接続
BTCONNECT_PAN	: PAN プロファイルで接続
BTCONNECT_OBEX_CLIENT	: OBEX プロファイルで接続
BTCONNECT_OBEX_SERVER	: OBEX プロファイルで接続
BTCONNECT_HEADSET_CLIENT	: BT ヘッドセットと接続
BTCONNECT_HEADSET_SERVER	: BT ヘッドセットと接続

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー
	DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 補足

- BTConnectHeadset 関数を実行した場合には、本関数を実行して接続が完了したかチェックしてください。
- DT-5300(WM) / IT-300 / DT-X8 / IT-9000 / IT-G500 / DT-X100 / DT-X200 以降の製品では、本 Bluetooth ライブラリでオープンした Bluetooth 接続のみに対応します。

## 5.24 BTSetConnectionParameter

Bluetooth 接続時に使用するパラメータを設定します。

```
[C++]
LONG BTSetConnectionParameter (
    DWORD SRMode,
    DWORD FastConnection
)
```

```
[Visual Basic]
Public Shared Function BTSetConnectionParameter ( _
    ByVal SRMode As Int32, _
    ByVal FastConnection As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTSetConnectionParameter (
    Int32 SRMode,
    Int32 FastConnection
);
```

### 解説

本関数を使用して SR モードまたは FastConnection モードを変更すると、BT モジュールが接続時に使用する電波の種類が変わり、Bluetooth 接続時間が速くなる場合があります。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、BTGetConnectionParameter 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### SRMode

SR モードを設定する変数

- BTSRMODE\_R0 : SR モードを R0 に設定する
- BTSRMODE\_R1 : SR モードを R1 に設定する
- BTSRMODE\_R2 : SR モードを R2 に設定する

#### FastConnection

Fast Connection モードを設定する変数

- BTPARAM\_DISABLE : Fast Connection 設定を無効にする
- BTPARAM\_ENABLE : Fast Connection 設定を有効にする

### 戻り値

- BTERR\_SUCCESS : 正常終了
- BTERR\_FAILED : デバイス情報設定失敗
- BTERR\_DRIVER : ドライバーエラー  
DeviceEmulator では発生しません
- FUNCTION\_UNSUPPORTED : 未サポートエラー

## 5.25 BTGetConnectionParameter

Bluetooth 接続時に使用するパラメータを取得します。

```
[C++]
LONG BTGetConnectionParameter (
    DWORD *SRMode,
    DWORD *FastConnection
)
```

```
[Visual Basic]
Public Shared Function BTGetConnectionParameter ( _
    ByRef SRMode As Int32, _
    ByRef FastConnection As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTGetConnectionParameter (
    ref Int32 SRMode,
    ref Int32 FastConnection
);
```

### 解説

Bluetooth 接続時に使用するパラメータを取得します。

### パラメータ

#### *SRMode*

本体の SR モードを取得します。取得する値の詳細は `BTSetConnectionParameter` 関数を参照してください。

#### *FastConnection*

Fast Connection モードを取得します。取得する値の詳細は `BTSetConnectionParameter` 関数を参照してください。

### 戻り値

<code>BTERR_SUCCESS</code>	: 正常終了
<code>BTERR_FAILED</code>	: デバイス情報設定失敗
<code>BTERR_DRIVER</code>	: ドライバーエラー
	DeviceEmulator では発生しません
<code>FUNCTION_UNSUPPORTED</code>	: 未サポートエラー

## 5.26 BTSetAFHStatus

Bluetooth AFH 機能を設定します。

```
[C++]
LONG BTSetAFHStatus(
    DWORD AFHMode,
    BYTE *AFHChannel
)
```

```
[Visual Basic]
Public Shared Function BTSetAFHStatus( _
    ByVal AFHMode As Int32, _
    ByVal AFHChannel As Byte() _
) As Int32
```

```
[C#]
public static Int32 BTSetAFHStatus(
    Int32 AFHMode,
    Byte[] AFHChannel
);
```

### 解説

Bluetooth AFH 機能を設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、BTGetAFHStatus 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *AFHMode*

- BTA FH\_DISABLE : AFH 設定を無効にする
- BTA FH\_AUTO : AFH 設定を有効にする (自動設定)
- BTA FH\_MANUAL : AFH 設定を有効にする (AFHChannel で周波数を指定する)

#### *AFHChannel*

AFH チャンネルを設定する変数

10 バイトの領域がある BYTE 型の変数を使用して設定する。

AFHMode を BTA FH\_MANUAL に設定した場合のみ有効です。

AFHMode を BTA FH\_DISABLE、BTA FH\_AUTO に設定した場合は、10 バイトの領域がある BYTE 型の変数を用意し、値は未設定で引数として下さい。

## 周波数の指定方法

Bluetooth の 79 個のチャンネルについて、個別に有効/無効を指定します。

10 バイト(80 ビット)の領域に、使用するチャンネル相当ビットは 1、使用しないチャンネル相当ビットは 0 を指定します。

チャンネル番号 (BTch:0~78) とバイト位置の対応は、以下のとおりです。

バイト	0*	1	2	3	4	5	6	7	8	9
BTch	78~72	71~64	63~56	55~48	47~40	39~32	31~24	23~16	15~8	7~0

※ 先頭バイトは、下位 7 ビットのみを使用し、最上位ビットは常に 0 とします。

チャンネル番号 BTch と周波数(2402~2480 MHz)は、以下の関係式で表されます。

$$\text{周波数} = 2402 + \text{BTch}$$

例:

使用する周波数範囲 : 2451 MHz ~ 2473 MHz  
チャンネル番号 BTch : 49 ch ~ 71 ch  
AFHChannel 設定値 : 00 FF FF FE 00 00 00 00 00

## 戻り値

BTERR\_SUCCESS : 正常終了  
BTERR\_FAILED : デバイス情報設定失敗  
BTERR\_DRIVER : ドライバーエラー  
DeviceEmulator では発生しません  
FUNCTION\_UNSUPPORTED : 未サポートエラー

## 5.27 BTGetAFHStatus

Bluetooth AFH 機能を取得します。

```
[C++]
LONG BTGetAFHStatus(
    DWORD *AFHMode,
    BYTE *AFHChannel
)
```

```
[Visual Basic]
Public Shared Function BTGetAFHStatus( _
    ByRef AFHMode As Int32, _
    ByRef AFHChannel( ) As Byte _
) As Int32
```

```
[C#]
public static Int32 BTGetAFHStatus(
    ref Int32 AFHMode,
    ref byte AFHChannel
)
```

### 解説

Bluetooth AFH 機能を取得します。

### パラメータ

#### *AFHMode*

Bluetooth AFH モードを取得する変数。取得する値の詳細は `BTSetAFHStatus` 関数を参照してください。

#### *AFHChannel*

AFH チャンネルを取得する変数。取得する値の詳細は `BTSetAFHStatus` 関数を参照してください。

### 戻り値

<code>BTERR_SUCCESS</code>	: 正常終了
<code>BTERR_FAILED</code>	: デバイス情報設定失敗
<code>BTERR_DRIVER</code>	: ドライバエラー <code>DeviceEmulator</code> では発生しません
<code>FUNCTION_UNSUPPORTED</code>	: 未サポートエラー

## 5.28 BTWaitForBtReady

BTInitialize 関数呼出し後の通信が可能になるまでの時間を待ちます。

```
[C++]  
LONG BTWaitForBtReady( )
```

```
[Visual Basic]  
Public Shared Function BTWaitForBtReady( ) As Int32
```

```
[C#]  
public static Int32 BTWaitForBtReady ( );
```

### 解説

Bluetooth モジュールの電源 ON 後 (BTInitialize 関数呼出し後)、通信が可能になるまでの時間を待ちます。

Device Emulator では、何もせずに BTERR\_SUCCESS を返します。

### パラメータ

なし

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗 DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー



## 5.29 BTConnectSerial

仮想シリアルプロファイルを使用して Bluetooth 機器との接続を実行します。

```
[C++]
LONG BTConnectSerial (
    DWORD ConnectionMode,
    DWORD ConnectionTimeout,
    DWORD ReceiveTimeout
)
```

```
[Visual Basic]
Public Shared Function BTConnectSerial ( _
    ByVal ConnectionMode As Int32, _
    ByVal ConnectionTimeout As Int32, _
    ByVal ReceiveTimeout As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTConnectSerial (
    Int32 ConnectionMode,
    Int32 ConnectionTimeout,
    Int32 ReceiveTimeout
);
```

### 解説

仮想シリアルプロファイルを使用して Bluetooth 機器との接続を実行します。

Device Emulator では、BluetoothLibrary の初期化の確認のみを行います。また、正常終了時は、実行結果を内部変数として格納します。

### パラメータ

#### *ConnectionMode*

接続モードの設定

- BTCONNECT\_SERIAL\_CLIENT : マスターモードで接続する
- BTCONNECT\_SERIAL\_SERVER : スレーブモードで接続する

#### *ConnectionTimeout*

仮想シリアルプロファイル接続のタイムアウト時間をミリ秒で指定する

#### *ReceiveTimeout*

BTReceiveSerialData 関数の受信タイムアウト時間をミリ秒で指定する

### 戻り値

- BTERR\_SUCCESS : 正常終了
- BTERR\_FAILED : デバイス情報設定失敗
- BTERR\_DRIVER : ドライバーエラー  
DeviceEmulator では発生しません
- FUNCTION\_UNSUPPORTED : 未サポートエラー

## 5.30 BTSendSerialData

仮想シリアルプロファイルを使用して、接続した Bluetooth 機器へデータを送信します。

```
[C++]
LONG BTSendSerialData(
    LPVOID Buffer,
    DWORD DataSize,
    DWORD *SendSize
)
```

```
[Visual Basic]
Public Shared Function BTSendSerialData( _
    ByRef Buffer As Byte, _
    ByVal DataSize As Int32, _
    ByRef SendSize As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTSendSerialData(
    ref byte Buffer,
    Int32 DataSize,
    ref Int32 SendSize
);
```

### 解説

仮想シリアルポートに対して Windows API の WriteFile を実行したときと同じ動作をします。  
Device Emulator では、SendSize に DataSize のサイズを代入します。

### パラメータ

#### Buffer

送信するデータのポインタ（※WriteFile 関数の第 2 引数に相当します）

#### DataSize

送信するデータのサイズ(単位は Byte)（※WriteFile 関数の第 3 引数に相当します）

#### SendSize

実際に送信されたデータのサイズ(単位は Byte)（※WriteFile 関数の第 4 引数に相当します）

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 注意

.NET でより安全に使用するためには、ガベージコレクションの再配置を禁止する必要があります。

```
[Visual Basic]
Dim buffer() As Byte = (省略)
Dim gh = GCHandle.Alloc(buffer, GCHandleType.Pinned)
ret = Api.BTSendSerialData(buffer(0), buffer.Length, length)
gh.Free()
```

## 5.31 BTReceiveSerialData

仮想シリアルプロファイルを使用して、接続した Bluetooth 機器からデータを受信します。

```
[C++]
LONG BTReceiveSerialData(
    LPVOID Buffer,
    DWORD DataSize,
    DWORD *ReceiveSize
)
```

```
[Visual Basic]
Public Shared Function BTReceiveSerialData( _
    ByRef Buffer As Byte, _
    ByVal DataSize As Int32, _
    ByRef ReceiveSize As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTReceiveSerialData (
    ref byte Buffer,
    Int32 DataSize,
    ref Int32 ReceiveSize
);
```

### 解説

仮想シリアルポートに対して Windows API の ReadFile 関数を実行したときと同じ動作をします。  
Device Emulator では、SendSize に DataSize のサイズを代入します。

### パラメータ

#### Buffer

受信するデータを格納する変数のポインタ (※ReadFile 関数の第 2 引数に相当します)

#### DataSize

受信可能なデータの最大サイズ(単位は Byte) (※ReadFile 関数の第 3 引数に相当します)

#### ReceiveSize

実際に受信されたデータのサイズ(単位は Byte) (※ReadFile 関数の第 4 引数に相当します)

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.32 BTDisconnectSerial

仮想シリアルプロファイルを使用した Bluetooth 接続を切断します。

```
[C++]  
LONG BTDisconnectSerial (Void)
```

```
[Visual Basic]  
Public Shared Function BTDisconnectSerial () As Int32
```

```
[C#]  
public static Int32 BTDisconnectSerial ()
```

### 解説

仮想シリアルプロファイルを使用した Bluetooth 接続を切断します。  
Device Emulator では、BTConnectSerial 関数実行済の確認のみを行います。

### パラメータ

なし

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.33 BTSetPANStatus

Bluetooth PAN アダプタの状態を設定します。

```
[C++]
LONG BTSetPANStatus (
    DWORD BTPANStatus
)
```

```
[Visual Basic]
Public Shared Function BTSetPANStatus( _
    ByVal BTPANStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTSetPANStatus(
    Int32 BTPANStatus
);
```

### 解説

Bluetooth PAN アダプタの状態を設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、BTGetPANStatus 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *BTPANStatus*

PAN アダプタの状態を設定する変数

BTPARAM\_DISABLE : BT PAN アダプタを使用不可にする  
BTPARAM\_ENABLE : BT PAN アダプタを使用可能にする

### 戻り値

BTERR\_SUCCESS : 正常終了  
BTERR\_FAILED : デバイス情報設定失敗  
BTERR\_DRIVER : ドライバーエラー  
DeviceEmulator では発生しません  
FUNCTION\_UNSUPPORTED : 未サポートエラー

## 5.34 BTGetPANStatus

Bluetooth PAN アダプタの状態を取得します。

```
[C++]  
LONG BTGetPANStatus (  
    DWORD *BTPANStatus  
)
```

```
[Visual Basic]  
Public Shared Function BTGetPANStatus( _  
    ByRef BTPANStatus As Int32 _  
) As Int32
```

```
[C#]  
public static Int32 BTGetPANStatus(  
    ref Int32 BTPANStatus  
);
```

### 解説

Bluetooth PAN アダプタの状態を取得します。

### パラメータ

#### *BTPANStatus*

PAN アダプタの状態を取得する変数 取得する値の詳細は **BTSetPANStatus** 関数を参照してください。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.35 BTConnectPAN

Bluetooth PAN プロファイルを使用して Bluetooth 機器との接続を実行します。

```
[C++]
LONG BTConnectPAN(
    DWORD ConnectionTimeout
)
```

```
[Visual Basic]
Public Shared Function BTConnectPAN( _
    ByVal ConnectionTimeout As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTConnectPAN(
    Int32 ConnectionTimeout
);
```

### 解説

Bluetooth PAN アダプタの状態を設定します。

Device Emulator では、BluetoothLibrary の初期化の確認のみを行います。また、正常終了時は、実行結果を内部変数として格納します。

### パラメータ

#### *ConnectionTimeout*

Bluetooth PAN 接続のタイムアウト時間をミリ秒で指定する

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー



## 5.36 BTDisconnectPAN

Bluetooth PAN プロファイルを使用した Bluetooth 機器との接続を切断します。

```
[C++]  
LONG BTDisconnectPAN(void)
```

```
[Visual Basic]  
Public Shared Function BTDisconnectPAN() As Int32
```

```
[C#]  
public static Int32 BTDisconnectPAN()
```

### 解説

Bluetooth PAN アダプタの状態を設定します。

Device Emulator では、BTConnectPAN 関数実行済の確認のみを行います。

### パラメータ

なし

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.37 BTConnectHeadset

Bluetooth ヘッドセットへの接続を実行します。

```
[C++]  
LONG BTConnectHeadset(void)
```

```
[Visual Basic]  
Public Shared Function BTConnectHeadset() As Int32
```

```
[C#]  
public static Int32 BTConnectHeadset()
```

### 解説

Bluetooth ヘッドセットへの接続を実行します。

Device Emulator では、BluetoothLibrary の初期化の確認のみを行います。また、正常終了時は、実行結果を内部変数として格納します。

### パラメータ

なし

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSupport	: 未サポートエラー

### 備考

本関数を実行する前に、BTSelectDevice 関数または BTSetDefaultDevice 関数によって、接続する Bluetooth 機器が設定されている必要があります。

接続する Bluetooth 機器が設定されていない場合、エラーとなります。

本関数を実行しただけでは、Bluetooth ヘッドセットとの接続は完了しません。関数を実行後、通信先の Bluetooth ヘッドセットを操作して、接続を完了させる必要があります。

Bluetooth ヘッドセットとの接続が完了したかをチェックするには、本関数を実行後に BTGetConnectionStatus 関数を実行してください。

## 5.38 BTDisconnectHeadset

Bluetooth ヘッドセットとの接続を切断します。

```
[C++]  
LONG BTDisconnectHeadset(void)
```

```
[Visual Basic]  
Public Shared Function BTDisconnectHeadset() As Int32
```

```
[C#]  
public static Int32 BTDisconnectHeadset()
```

### 解説

Bluetooth ヘッドセットとの接続を切断します。

Device Emulator では、BTConnectHeadset 関数実行済の確認のみを行います。

### パラメータ

なし

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

### 備考

BTConnectHeadset 関数による Bluetooth 接続が実行されていない場合、エラーとなります。

通信先の Bluetooth 機器の状態によっては、切断に時間がかかる場合があります。

Bluetooth ヘッドセットを操作して接続の切断を実行した場合、本関数を実行する必要はありません。

## 5.39 BTSetSoundPath

音声経路の状態を設定します。

```
[C++]  
LONG BTSetSoundPath(  
    DWORD SoundPath  
)
```

```
[Visual Basic]  
Public Shared Function BTSetSoundPath( _  
    ByVal SoundPath As Int32 _  
) As Int32
```

```
[C#]  
public static Int32 BTSetSoundPath(  
    Int32 SoundPath  
);
```

### 解説

音声経路の状態を設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、BTGetSoundPath 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *SoundPath*

音声経路の状態を設定する変数

BTSOUND\_INTERNAL : 内蔵スピーカとマイクを使用する  
BTSOUND\_HEADSET : Bluetooth ヘッドセットを使用する

### 戻り値

BTERR\_SUCCESS : 正常終了  
BTERR\_FAILED : デバイス情報設定失敗  
BTERR\_DRIVER : ドライバーエラー  
DeviceEmulator では発生しません  
FUNCTION\_UNSUPPORTED : 未サポートエラー

## 5.40 BTGetSoundPath

本体の音声経路の状態を取得します。

```
[C++]
LONG BTGetSoundPath(
    DWORD *SoundPath
)
```

```
[Visual Basic]
Public Shared Function BTGetSoundPath( _
    ByRef SoundPath As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTGetSoundPath(
    ref Int32 SoundPath
);
```

### 解説

本体の音声経路の状態を取得します。

### パラメータ

#### *SoundPath*

音声経路の状態を取得する変数 取得する値の詳細は `BTSetSoundPath` 関数を参照してください。

### 戻り値

<code>BTERR_SUCCESS</code>	: 正常終了
<code>BTERR_FAILED</code>	: デバイス情報設定失敗
<code>BTERR_DRIVER</code>	: ドライバーエラー
	<code>DeviceEmulator</code> では発生しません
<code>FUNCTION_UNSUPPORTED</code>	: 未サポートエラー

## 5.41 BTSetHeadsetGain

接続している Bluetooth ヘッドセットのゲインを設定します。

```
[C++]
LONG BTSetHeadsetGain(
    DWORD SpeakerGain,
    DWORD MicrophoneGain
)
```

```
[Visual Basic]
Public Shared Function BTSetHeadsetGain( _
    ByVal SpeakerGain As Int32, _
    ByVal MicrophoneGain As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTSetHeadsetGain(
    Int32 SpeakerGain,
    Int32 MicrophoneGain
);
```

### 解説

音声経路の状態を設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、BTGetHeadsetGain 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *SpeakerGain*

Bluetooth ヘッドセットのスピーカゲインを設定する変数 0 から 15 の範囲の値を設定する。

#### *MicrophoneGain*

Bluetooth ヘッドセットのマイクゲインを設定する変数 0 から 15 の範囲の値を設定する

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー
	DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.42 BTGetHeadsetGain

接続している Bluetooth ヘッドセットのゲインを取得します。

```
[C++]
LONG BTGetHeadsetGain(
    DWORD *SpeakerGain,
    DWORD *MicrophoneGain
)
```

```
[Visual Basic]
Public Shared Function BTGetHeadsetGain( _
    ByRef SpeakerGain As Int32, _
    ByRef MicrophoneGain As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTGetHeadsetGain(
    ref Int32 SpeakerGain,
    ref Int32 MicrophoneGain
);
```

### 解説

音声経路の状態を取得します。

### パラメータ

#### *SpeakerGain*

取得する値の詳細は `BTSetHeadsetGain` 関数を参照してください。

#### *MicrophoneGain*

取得する値の詳細は `BTSetHeadsetGain` 関数を参照してください。

### 戻り値

<code>BTERR_SUCCESS</code>	: 正常終了
<code>BTERR_FAILED</code>	: デバイス情報設定失敗
<code>BTERR_DRIVER</code>	: ドライバーエラー
	<code>DeviceEmulator</code> では発生しません
<code>FUNCTION_UNSUPPORTED</code>	: 未サポートエラー

## 5.43 BTSetHeadsetServerStatus

Bluetooth ヘッドセットサーバの動作状態を設定します。

```
[C++]
LONG BTSetHeadsetServerStatus(
    DWORD ServerStatus
)
```

```
[Visual Basic]
Public Shared Function BTSetHeadsetServerStatus( _
    ByVal ServerStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTSetHeadsetServerStatus(
    Int32 ServerStatus
);
```

### 解説

Bluetooth ヘッドセットサーバの動作状態を設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、BTGetHeadsetServerStatus 関数を実行することにより、設定値を確認することができます。

### パラメータ

#### *ServerStatus*

Bluetooth ヘッドセットサーバの動作状態を設定する変数

BTPARAM\_DISABLE : BT ヘッドセットサーバを無効にする  
BTPARAM\_ENABLE : BT ヘッドセットサーバを有効にする

### 戻り値

BTERR\_SUCCESS : 正常終了  
BTERR\_FAILED : デバイス情報設定失敗  
BTERR\_DRIVER : ドライバーエラー  
DeviceEmulator では発生しません  
FUNCTION\_UNSUPPORTED : 未サポートエラー



## 5.44 BTGetHeadsetServerStatus

Bluetooth ヘッドセットサーバの動作状態を取得します。

```
[C++]
LONG BTGetHeadsetServerStatus (
    DWORD * ServerStatus
)
```

```
[Visual Basic]
Public Shared Function BTGetHeadsetServerStatus( _
    ByRef ServerStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTGetHeadsetServerStatus(
    ref Int32 ServerStatus
);
```

### 解説

Bluetooth ヘッドセットサーバの動作状態を取得します。

### パラメータ

#### *ServerStatus*

取得する値の詳細は [BTSetHeadsetServerStatus](#) を参照してください。

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー
	DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.45 BTSetOBEXFolder

Object Push プロファイル通信で使用するフォルダを設定します。

```
[C++]  
LONG BTSetOBEXFolder(  
    LPTSTR FolderName  
)
```

```
[Visual Basic]  
Public Shared Function BTSetOBEXFolder( _  
    ByVal FolderName As String _  
) As Int32
```

```
[C#]  
public static Int32 BTSetOBEXFolder(  
    string FolderName  
);
```

### 解説

Object Push プロファイル通信で使用するフォルダを設定します。

Device Emulator では、設定値を内部変数として格納するため、何も動作しませんが、BTGetOBEXFolder 関数を実行することにより、設定値を確認することができます。

### パラメータ

*FolderName*

Object Push プロファイル通信で使用するフォルダを取得する変数

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.46 BTGetOBEXFolder

Object Push プロファイル通信で使用するフォルダを取得します。

```
[C++]
LONG BTGetOBEXFolder (
    LPTSTR FolderName,
    DWORD FolderLength
)
```

```
[Visual Basic]
Public Shared Function BTGetOBEXFolder ( _
    ByVal FolderName As String, _
    ByVal FolderLength As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTGetOBEXFolder (
    string FolderName,
    Int32 FolderLength
);
```

### 解説

Object Push プロファイル通信で使用するフォルダを取得します。  
Device Emulator では、BluetoothLibrary の初期化の確認のみを行います。

### パラメータ

#### *FolderName*

Object Push プロファイル通信で使用するフォルダを取得する変数

#### *FolderLength*

*FolderName* 変数の長さを設定する変数

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 5.47 BTSendOBEXFile

Object Push プロファイルを使用してファイル送信を実行します。

```
[C++]
LONG BTSendOBEXFile(
    LPTSTR FileName,
    DWORD ObjectType
)
```

```
[Visual Basic]
Public Shared Function BTSendOBEXFile( _
    ByVal FileName As String, _
    ByVal ObjectType As Int32 _
) As Int32
```

```
[C#]
public static Int32 BTSendOBEXFile(
    string FileName,
    Int32 ObjectType
);
```

### 解説

Object Push プロファイルを使用してファイル送信を実行します。  
Device Emulator では、BluetoothLibrary の初期化の確認のみを行います。

### パラメータ

#### *FileName*

送信するファイル名

#### *ObjectType*

送信するオブジェクトタイプ

BTOBEX_OBJECT_VCARD	: 名刺
BTOBEX_OBJECT_VCALENDAR	: スケジュール
BTOBEX_OBJECT_VNOTE	: メモ
BTOBEX_OBJECT_VMESSAGE	: メール
BTOBEX_OBJECT_ALL	: その他のファイル

### 戻り値

BTERR_SUCCESS	: 正常終了
BTERR_FAILED	: デバイス情報設定失敗
BTERR_DRIVER	: ドライバーエラー DeviceEmulator では発生しません
FUNCTION_UNSUPPORTED	: 未サポートエラー

## 6. 各プロファイルの Bluetooth 機器接続手順

HT がサポートしているプロファイルを使用して、他の Bluetooth 機器との接続および通信を行う手順を示します。

### 6.1 通信する Bluetooth 機器情報の登録

本体の Bluetooth デバイスをマスターモードで使用する場合には、以下の手順に従って通信する Bluetooth 機器の登録を行ってください。

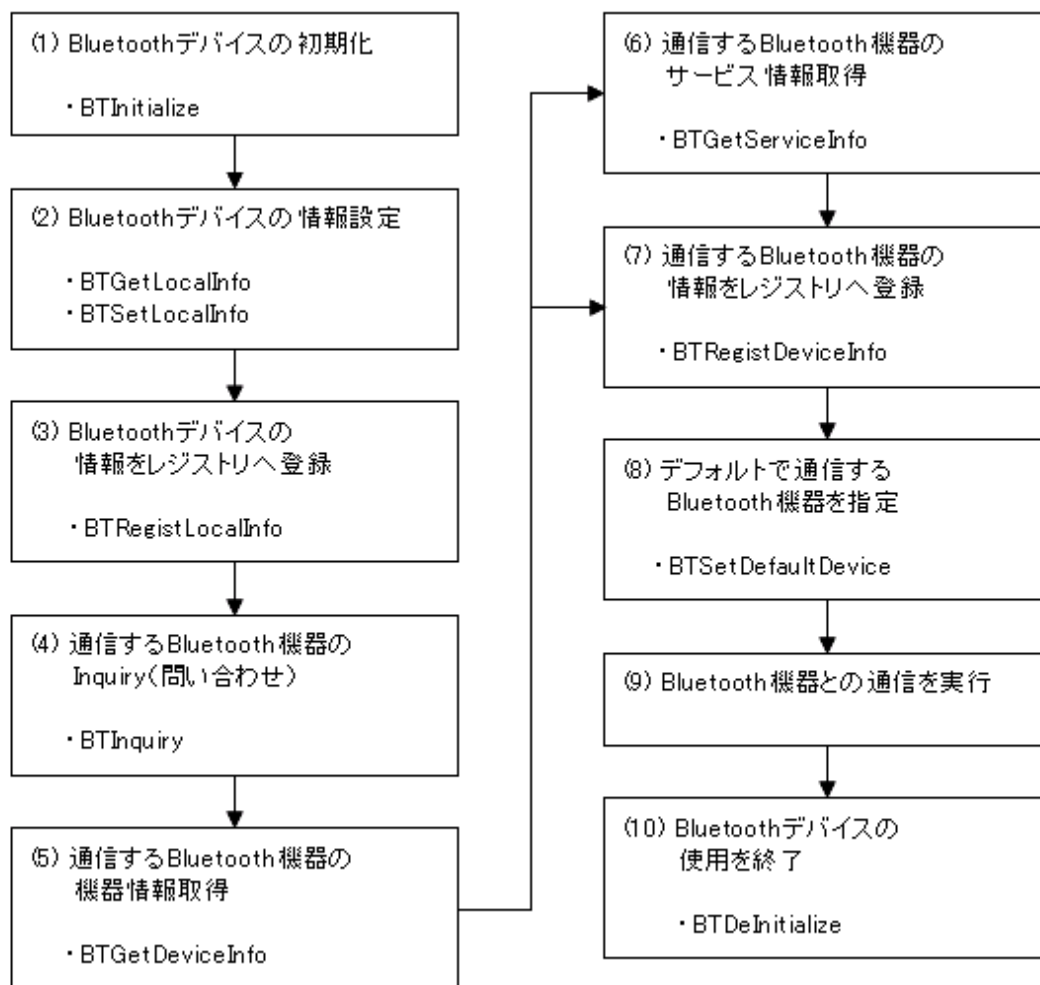


図 6.1

- 手順(6)は省略することが可能です。
- デフォルトで通信する Bluetooth 機器を指定する場合は、手順(8)を実行してください。
- Bluetooth 機器情報の登録終了後直ちに通信を行う場合は、手順(9)を実行してください。

## 6.2 シリアルプロフィール使用時の接続方法

シリアルプロフィールを使用する場合、本体の Bluetooth デバイスの動作モードによって、2 通りの接続方法があります。

### (1)クライアント(マスタ)モードで接続する場合

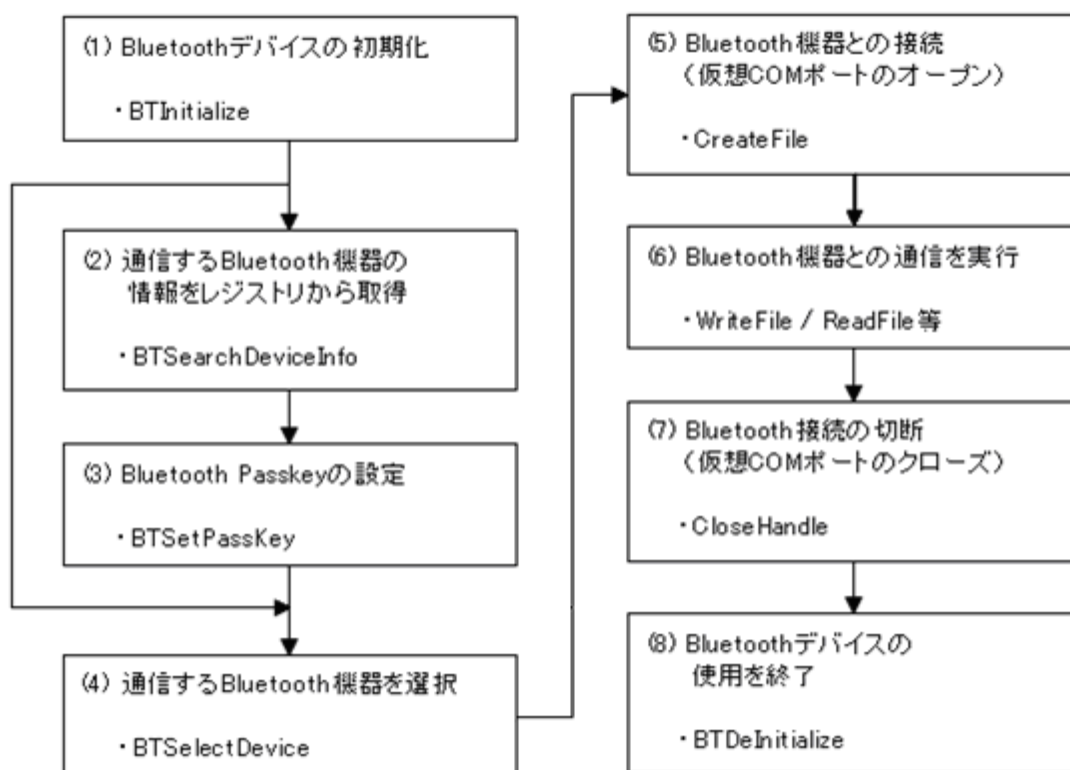


図 6.2

- デフォルトの Bluetooth 機器と接続する場合、手順(2)と手順(4)は不要です。
- 通信先の Bluetooth 機器が PassKey を要求する場合は、手順(3)を実行してください。
- 複数の Bluetooth 機器とシリアル通信を行う場合は、手順(2)から手順(7)を繰り返し実行してください。

## (2)サーバ(スレーブ)モードで接続する場合

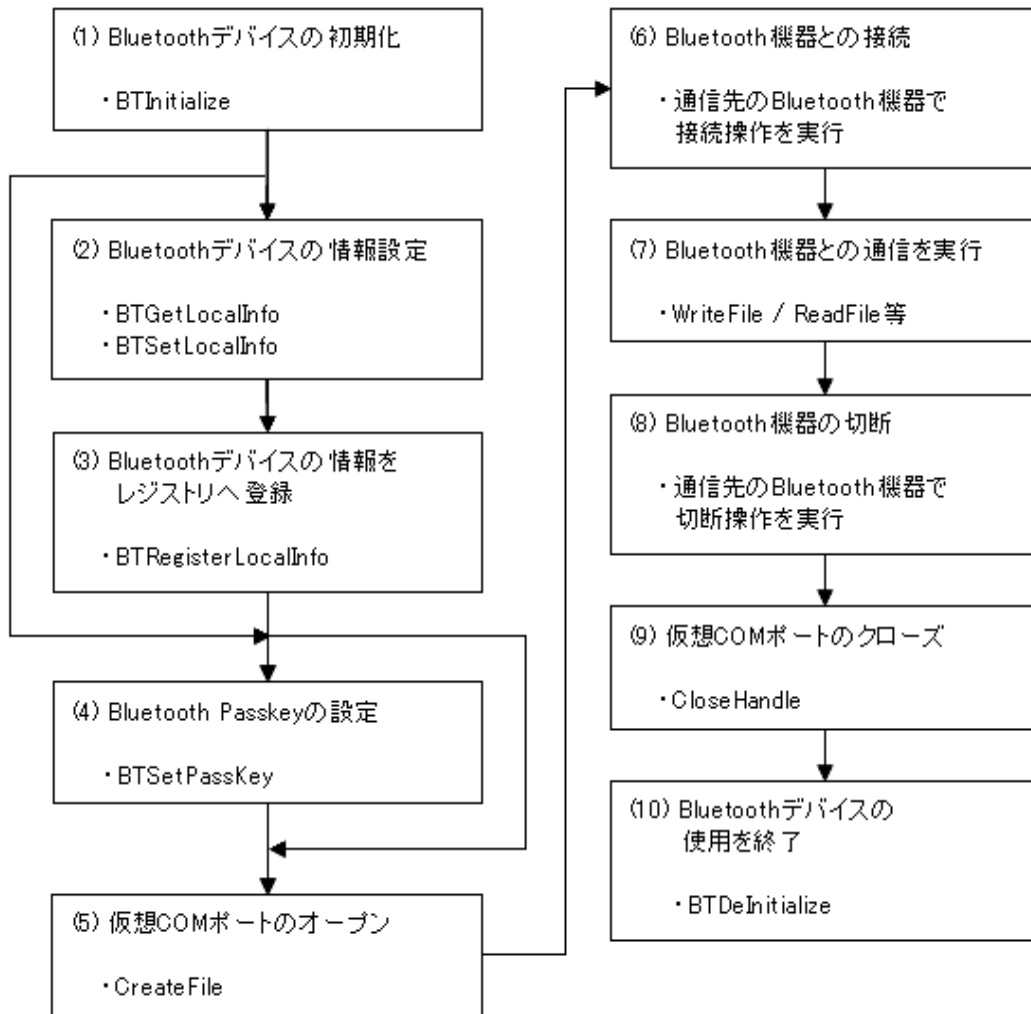


図 6.3

- 複数の Bluetooth 機器とシリアル通信を行う場合は、手順(2)から手順(9)を繰り返し実行してください。

## 6.3 ダイアルアッププロファイル使用時の接続方法

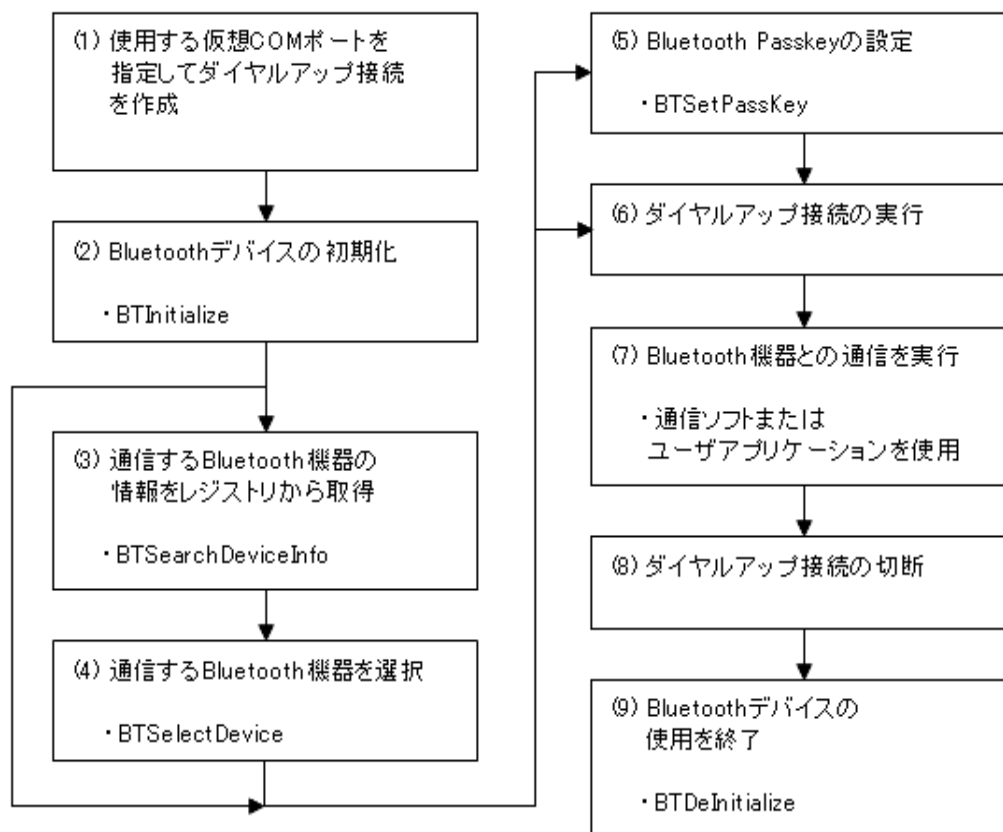


図 6.4

手順(1)、(6)、(7)、(8)はHTのコントロールパネル「ダイアルアップとネットワーク接続」を操作してください。具体的な操作方法については、下記マニュアルを参照してください。

- デフォルトの Bluetooth 機器と接続する場合、手順(3)と手順(4)は不要です。
- 通信先の Bluetooth 機器が PassKey を要求する場合は、手順(4)を実行してください。
- 複数の Bluetooth 機器とダイアルアップ通信を行う場合は、手順(3)から手順(8)を繰り返し実行してください。

機種	参照マニュアル
DT-5200	DT-5200 Bluetooth 接続マニュアル
DT-X7	DT-X7 Bluetooth 接続マニュアル
上記以外	各機種別ソフトウェアマニュアル



## 6.4 LAN プロファイル使用時の接続方法

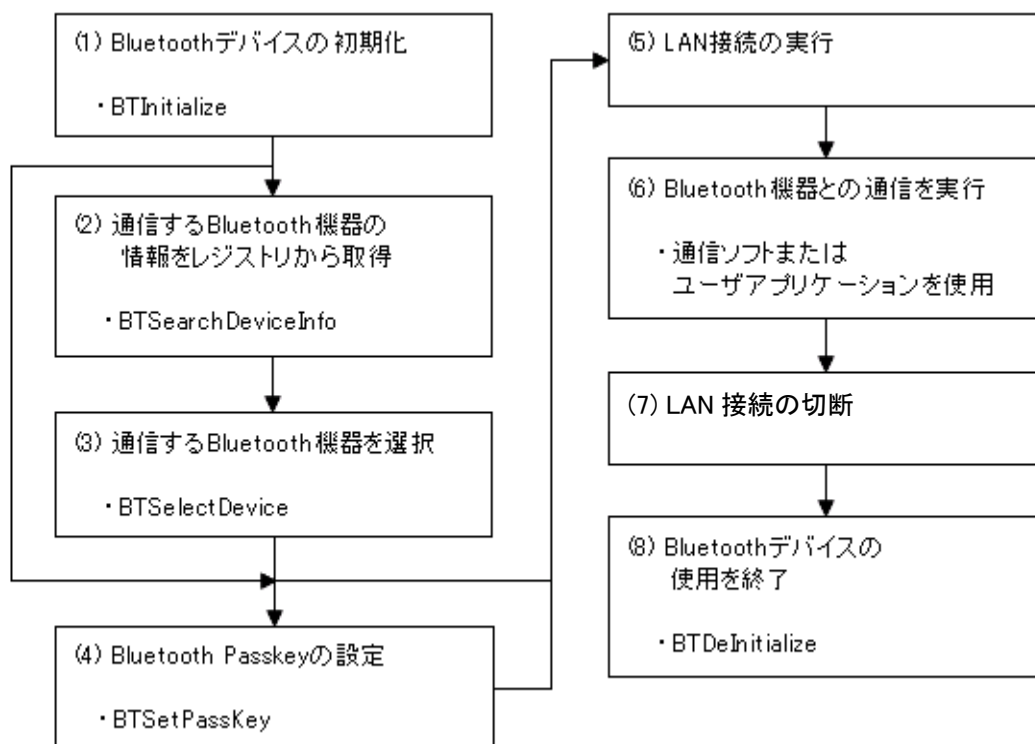


図 6.5

手順(5)、(6)、(7)は、コントロールパネル「ダイヤルアップとネットワーク接続」を操作してください。具体的な操作方法については、下記マニュアルを参照してください。

- デフォルトの Bluetooth 機器と接続する場合、手順(2)と手順(3)は不要です。
- 通信先の Bluetooth 機器が PassKey を要求する場合は、手順(4)を実行してください。
- 複数の Bluetooth 機器とダイヤルアップ通信を行う場合は、手順(2)から手順(7)を繰り返し実行してください。

機種	参照マニュアル
DT-5200	DT-5200 Bluetooth 接続マニュアル
DT-X7	DT-X7 Bluetooth 接続マニュアル
上記以外	各機種別ソフトウェアマニュアル

## 7. プログラミング上の注意点

通常は本体内蔵の Bluetooth 接続ツールを使用して Bluetooth 機器との接続を行います。本ライブラリを使用することにより、ユーザアプリケーションから直接 Bluetooth 機器との接続および通信を行うことができます。なお、Bluetooth ライブラリと Bluetooth 接続ツールを同時に使用することはできません。

Bluetooth 接続ツールが起動している場合、Bluetooth ライブラリを使用したユーザアプリケーションを起動する前に、Bluetooth 設定ツールを終了させてください。

### 7.1 サポート通信プロファイルについて

サポートする通信プロファイルは以下の通りです。いずれのプロファイルにおいても、仮想 COM ポートを使用して通信を行います。

シリアルプロファイル(マスタ)	: COM6
シリアルプロファイル(スレーブ)	: COM7

ダイヤルアッププロファイルでは、OS によりモデム名が異なります。

Windows CE 5.0	: BluetoothDUN
Windows CE 6.0	
Windows Embedded Compact 7	
Windows Mobile 6.5	: BluetoothDUNLIB
Windows Mobile 6.5.3	
Windows Embedded Handheld 6.5	

## 7.2 Bluetooth の通信動作モードについて

Bluetooth 機器間で通信を行う場合、Bluetooth 通信機器はピコネットと呼ばれるワイヤレスネットワークを構成し、その中で各 Bluetooth 機器はマスタまたはスレーブのいずれかのモードで動作します。

### 通信先の Bluetooth 機器が Bluetooth モデム・携帯電話・アクセスポイント・プリンタ等の場合

本体をマスタモード、他の通信機器をスレーブモードで使用します。

### 通信先の Bluetooth 機器が PC・他のHTの場合

一方の Bluetooth 機器をマスタモード、もう一方をスレーブモードで使用します。

#### ■本体をマスタモードにして通信する場合

本体をマスタモードにして通信する場合の概要を示します。

##### (1) 通信する Bluetooth 機器のデバイス情報の登録

- Bluetooth 機器の Inquiry (問い合わせ) を実行
- Inquiry で発見した Bluetooth 機器の情報を取得
- 取得した Bluetooth 機器情報をレジストリに登録
- 必要に応じて、デフォルトで通信する Bluetooth 機器を設定

##### (2) 通信する Bluetooth 機器の選択および通信の実行

- レジストリから Bluetooth 機器情報を取得
- 通信する Bluetooth 機器を選択  
デフォルトに指定した Bluetooth 機器と通信する場合、上記操作は不要
- 選択した Bluetooth 機器との接続および通信を実行

#### ■本体をスレーブモードにして通信する場合

本体をスレーブモードにして通信する場合の概要を示します。

##### (1) HT のデバイス情報の登録

- 通信先の Bluetooth 機器で、HT の Inquiry (問い合わせ) を実行
- HT の Bluetooth デバイス情報を通信先の Bluetooth 機器に登録

##### (2) Bluetooth 通信の実行

- 通信先の Bluetooth 機器で、通信する Bluetooth 機器として HT を指定
- 通信先の Bluetooth 機器で、HT との接続を実行
- 通信先の Bluetooth 機器との通信を実行

## 7.3 本体電源 ON 直後のエラー発生について

BTInitialize 関数呼び出し済み状態からの本体電源 ON 直後は Bluetooth を再初期化します。このタイミングで Bluetooth ライブラリの関数を呼ぶと、BTERR\_DRIVER (BTGetLastError は BTERR\_LIB\_INIT) が発生します。この場合は初期化完了まで待ってから、ライブラリ関数を呼ぶ必要があります。

初期化完了まで待つ方法として、具体的には BTGetLibraryStatus 関数で状態を取得し、再初期化中 (BTSTATUS\_REINITIALIZING) であれば、この関数をポーリングで呼び出して初期化が完了 (BTSTATUS\_INITIALIZED) するまで待ってください。

ポーリング間隔は 1 秒程度を推奨します。

BTGetLibraryStatus 関数では以下の状態を取得できます。(詳細は「5.21 BTGetLibraryStatus」を参照してください。)

BTSTATUS_NOT_INITIALIZED	: 初期化されていない
BTSTATUS_INITIALIZED	: 初期化が完了している
BTSTATUS_REINITIALIZING	: 再初期化処理を実行中
BTSTATUS_REINIT_FAILED	: 再初期化に失敗した

## 8. Device Emulator

Device Emulator 上で本ライブラリを正しく動作させるためには、以下の設定ファイルが必要です。

BTInit.ini

BTDeviceInfo[n].ini ([n]は 0~256 の数値)

BTReg.ini

設定ファイルはあらかじめ以下のフォルダにインストールされています。

¥Storage Card¥Bluetooth

### 8.1 BTInit.ini

デバイスの初期化情報を保存するファイルです。

BTInit.ini のサンプルを以下に示します。

```
[Init]
Already=0
[Local Info]
LocalName="Emulator"
LocalAddress="00:00:00:00:00:00"
LocalDeviceMode=5
LocalClass1=1
LocalClass2=5
LocalClass3=0
Authentication=0
Encryption=0
[WakeOnStatus]
WakeOn=0
```

<b>[Init]</b> 初期化セクション	
Already	Bluetooth プロトコルスタックの未初期化/初期化済を指定します。 BluetoothLib を正しく動作させるためには、必ず <b>0</b> を指定してください。 <b>0</b> :未初期化 <b>1</b> :初期化済
<b>[LocalInfo]</b> ローカルデバイス情報セクション	
LocalName	デバイス名を <b>81</b> 文字以内の文字列で指定します。 (参照:BTST_LOCALINFO 構造体)
LocalAddress	デバイスのアドレスを文字列で指定します。 (参照:BTST_LOCALINFO 構造体)
LocalDeviceMode	モード設定を数値で指定します。 (参照:BTST_LOCALINFO 構造体)
LocalClass1	クラス設定を数値で指定します。 (参照:BTST_LOCALINFO 構造体)
LocalClass2	クラス設定を数値で指定します。 (参照:BTST_LOCALINFO 構造体)
LocalClass3	クラス設定を数値で指定します。 (参照:BTST_LOCALINFO 構造体)
Authentication	認証フラグを指定します。(参照:BTST_LOCALINFO 構造体) <b>0</b> :FALSE <b>1</b> :TRUE
Encryption	暗号化フラグを指定します。(参照:BTST_LOCALINFO 構造体) 認証フラグが <b>0</b> の場合は、暗号化フラグも <b>0</b> を指定してください。 <b>0</b> :FALSE <b>1</b> :TRUE
<b>[WakeOnStatus]</b> WakeOn 設定セクション	
WakeOn	WakeOn フラグを指定します。(参照:BTSetWakeOnStatus 関数) <b>0</b> :FALSE <b>1</b> :TRUE

## 8.2 BTDeviceInfo[n].ini

通信先デバイス情報を保存するファイルです。

フォルダ内にファイルを追加することにより、通信先デバイスを増やすことができます。

BTDeviceInfo[n].ini のサンプルを以下に示します。

```
[DeviceInfo]
DeviceAddress="00:00:00:00:00:01"
DeviceName="BTDevice1"
DeviceClass1=1
DeviceClass2=5
DeviceClass3=0
ProfileUUID0=4353
ProfileUUID1=4354
PassKey="123"
```

[DeviceInfo] 通信先デバイス情報セクション	
DeviceAddress	デバイスのアドレスを文字列で指定します。 (参照:BTST_DEVICEINFO 構造体)
DeviceName	デバイス名を 81 文字以内の文字列で指定します。 (参照:BTST_DEVICEINFO 構造体)
DeviceClass1	クラス設定を数値で指定します。(参照:BTST_DEVICEINFO 構造体)
DeviceClass2	クラス設定を数値で指定します。(参照:BTST_DEVICEINFO 構造体)
DeviceClass3	クラス設定を数値で指定します。(参照:BTST_DEVICEINFO 構造体)
ProfileUUID0 : ProfileUUID16	プロファイルを数値で指定します。(参照:BTST_DEVICEINFO 構造体)
PassKey	認証用 PassKey を文字列で指定します。 (参照:BTST_DEVICEINFO 構造体)

## 8.3 BTReg.ini

BTSetDefaultDevice 関数を実行せずに、BTGetDefaultDeviceInfo 関数を実行した場合に、取得する Bluetooth 機器の情報を保存するファイルです。

また、BTSetDefaultDevice 関数を実行した場合は、本ファイルを更新します。

BTReg.ini のサンプルを以下に示します。

```
[LocalInfo]
LocalName="Emulator"
LocalAddress="00:00:00:00:00:00"
LocalDeviceMode=5
LocalClass1=1
LocalClass2=5
LocalClass3=0
Authentication=0
Encryption=0
[DeviceInfo]
DeviceAddress="00:00:00:00:00:01"
DeviceName="BTDevice1"
DeviceClass1=1
DeviceClass2=5
DeviceClass3=0
ProfileUUID0=4353
ProfileUUID1=4354
PassKey="123"
```

※ 本ファイルの各項目については、BTInit.ini および BTDeviceInfo[n].ini を参照してください。



## カシオ計算機お問い合わせ窓口

### 製品に関する最新情報

- 製品サポートサイト（カシオペア・ハンディターミナル）

<http://casio.jp/support/ht/>

## カシオ計算機株式会社

〒151-8543 東京都渋谷区本町 1-6-2

TEL 03-5334-4638(代)