# Tx Power Service (TPS)

## Application Programming Interface Reference Manual

**Profile Version: 1.0**

**Release: 4.0.1**
**January 10, 2013**

Louisville, KY www.stonestreetone.com

Bluetooth and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc., USA and licensed to Stonestreet One, LLC. Bluetopia®, Stonestreet One™, and the Stonestreet One logo are registered trademarks of Stonestreet One, LLC, Louisville, Kentucky, USA. All other trademarks are property of their respective owners.
Copyright © 2000-2013 by Stonestreet One, LLC. All rights reserved.

# Table of Contents

# 1. Introduction

Bluetopia®+LE is Stonestreet One's Bluetooth protocol stack that supports the adopted Bluetooth low energy specification. Stonestreet One's upper level protocol stack that supports Single Mode devices is Bluetopia®+LE Single. More specifically, this stack is a software solution that resides above the Physical HCI (Host Controller Interface) Transport Layer and extends through the L2CAP (Logical Link Control and Adaptation Protocol), ATT (Attribute Protocol) Link Layers, the GAP (Generic Attribute Profile) Layer and the Genetic Attribute Protocol (GATT) Layer. In addition to basic functionality of these layers, the Bluetooth Protocol Stack by Stonestreet One provides implementations of the Device Information Service (DIS), GLS (Glucose Service), and several of the Bluetooth Profiles. Program access to these layers, services, and profiles is handled via Application Programming Interface (API) calls.

The remainder of this chapter has sections on the scope of this document, other documents applicable to this document, and a listing of acronyms and abbreviations. Chapter 2 is the API reference that contains a description of all programming interfaces for the Tx Power Service Profile Stack provided by Bluetopia®+LE Single. And, Chapter 3 contains the header file name list for the Tx Power Service library.

## 1.1 Scope

This reference manual provides information on the TPS API. This API is available on the full range of platforms supported by Stonestreet One:

- Windows
- Linux
- Windows Mobile
- QNX
- Windows CE
- Other Embedded OS

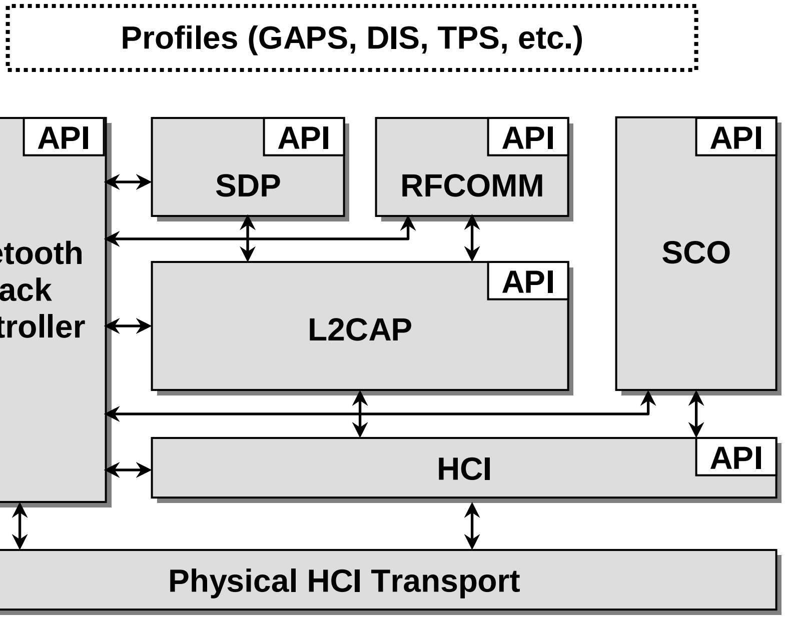

**Figure 1-1 The Stonestreet One Bluetooth Protocol Stack**

## 1.2 Applicable Documents

The following documents may be used for additional background and technical depth regarding the Bluetooth technology.

1. *Specification of the Bluetooth System, Volume 1, Architecture and Terminology Overview*, version 4.0, June 30, 2010.
2. *Specification of the Bluetooth System, Volume 6, Core System Package [Low Energy Controller Volume]*, version 4.0, June 30, 2010.
3. *Bluetopia® Protocol Stack, Application Programming Interface Reference Manual*, version 4.0.1, January 10, 2013.
4. *Bluetooth Tx Power  Service Specification*, version v10r00, April 3, 2012.

Possible error returns are listed for each API function call.  These are the *most likely* errors, but in fact programmers should allow for the possibility of any error listed in the BTErrors.h header file to occur as the value of a function return.

## 1.3 Acronyms and Abbreviations

Acronyms and abbreviations used in this document and other Bluetooth specifications are listed in the table below.

| Term | Meaning |
|---|---|
| API | Application Programming Interface |
| ATT | Attribute Protocol |
| BD_ADDR | Bluetooth Device Address |
| BT | Bluetooth |
| GAPS | Generic Access Profile Service |
| GATT | Generic Attribute Protocol |
| HCI | Host Controller Interface |
| HS | High Speed |
| L2CAP | Logical Link Control and Adaptation Protocol |
| LE | Low Energy |
| LSB | Least Significant Bit |
| MSB | Most Significant Bit |
| TPS | Tx Power Service |

# 2. TPS Programming Interface

The Tx Power Service, TPS, programming interface defines the protocols and procedures to be used to implement TPS capabilities for both Server and Client services. The TPS commands are listed in section 2.1, the event callback prototypes are described in section 2.2, the TPS events are itemized in section 2.3. The actual prototypes and constants outlines in this section can be found in the **TPSAPI.h** header file in the Bluetopia distribution.

## 2.1 Tx Power Service Commands

The available TPS command functions are listed in the table below and are described in the text that follows.

| **Server Commands** | |
|---|---|
| **Function** | **Description** |
| TPS_Initialize_Service | Opens a TPS Server. |
| TPS _Cleanup_Service | Closes an opened TPS Server. |
| TPS_Set_Tx_Power_Level | Sets the Tx Power Level on the specified TPS Instace. |
| TPS_Query_Tx_Power_Level | Gets the current Tx Power Level on the specified TPS Instance. |

### TPS_Initialize_Service

This function opens a TPS Server on a specified Bluetooth Stack.

**Notes:**

1. Only one TPS Server, per Bluetooth Stack ID, may be open at a time.
2. All Client Requests will be dispatched to the EventCallback function that is specified by the second parameter to this function.

**Prototype:**

int BTPSAPI **TPS_Initialize_Service**(unsigned int BluetoothStackID, unsigned int *ServiceID);

**Parameters:**

| | |
|---|---|
| BluetoothStackID[1] | Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize. |
| ServiceID | Unique GATT Service ID of the registered TPS service returned from GATT_Register_Service API. |

**Return:**

Positive non-zero if successful. The return value will be the Service ID of TPS Server that was successfully opened on the specified Bluetooth Stack ID. This is the value that should be used in all subsequent function calls that require Instance ID.

Negative if an error occurred. Possible values are:

TPS_ERROR_INSUFFICIENT_RESOURCES
TPS_ERROR_SERVICE_ALREADY_REGISTERED
TPS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_INVALID_SERVICE_TABLE_FORMAT
BTGATT_ERROR_INSUFFICIENT_RESOURCES
BTGATT_ERROR_INVALID_PARAMETER
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_NOT_INITIALIZED

**Notes:**

1. The BluetoothStackID parameter is not included in versions of Bluetopia that have been optimized to only control a single Bluetooth device, such as some embedded versions of Bluetopia. Please refer to the appropriate header file to determine if this parameter is part of the function call or not.

## TPS_Cleanup_Service

This function is responsible for cleaning up and freeing all resources associated with a TPS Service Instance. After this function is called, no other TPS Service function can be called until after a successful call to the TPS_Initialize_Service() function is performed.

**Prototype:**

int BTPSAPI **TPS_Cleanup_Service**(unsigned int BluetoothStackID, unsigned int InstanceID);

**Parameters:**

| | |
|---|---|
| BluetoothStackID[1] | Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize. |
| InstanceID | The Service Instance ID to close. This InstanceID was returned from the TPS_Initialize_Service(). |

**Return:**

Zero if successful.

Negative if an error occurred. Possible values are:

TPS_ERROR_INVALID_PARAMETER
TPS_ERROR_INVALID_INSTANCE_ID

**Notes:**

1. The BluetoothStackID parameter is not included in versions of Bluetopia that have been optimized to only control a single Bluetooth device, such as some embedded versions of Bluetopia. Please refer to the appropriate header file to determine if this parameter is part of the function call or not.

## TPS_Set_Tx_Power_Level

This function is responsible for setting the Tx Power Level on the specified TPS Instance.

**Prototype:**

int BTPSAPI **TPS_Set_Tx_Power_Level**(unsigned int BluetoothStackID,
    unsigned int InstanceID, SByte_t Tx_Power_Level);

**Parameters:**

| | |
|---|---|
| BluetoothStackID[1] | Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize. |
| InstanceID | The Service Instance ID to close. This InstanceID was returned from the TPS_Initialize_Service(). |
| Tx_Power_Level | The Tx Power Level to set for the specified TPS Instance. |

**Return:**

Zero if successful.

Negative if an error occurred. Possible values are:

TPS_ERROR_INVALID_INSTANCE_ID
TPS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER

**Notes:**

1. The BluetoothStackID parameter is not included in versions of Bluetopia that have been optimized to only control a single Bluetooth device, such as some embedded versions of Bluetopia. Please refer to the appropriate header file to determine if this parameter is part of the function call or not.

## TPS_Query_Tx_Power_Level

This function is responsible for querying the current Tx Power Level on the specified TPS Instance.

**Prototype:**

int BTPSAPI **TPS_Query_Tx_Power_Level**(unsigned int BluetoothStackID,
    unsigned int InstanceID, SByte_t *Tx_Power_Level);

**Parameters:**

BluetoothStackID[1] Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.

InstanceID The Service Instance ID to close. This InstanceID was returned from the TPS_Initialize_Service().

Tx_Power_Level A pointer to return the current Tx Power Level for the specified TPS Instance.

**Return:**

Zero if successful.

An error code if negative; one of the following values:

TPS_ERROR_INVALID_INSTANCE_ID
TPS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER

**Notes:**

1. The BluetoothStackID parameter is not included in versions of Bluetopia that have been optimized to only control a single Bluetooth device, such as some embedded versions of Bluetopia. Please refer to the appropriate header file to determine if this parameter is part of the function call or not.

# 3. File Distributions

The header files that are distributed with the Bluetooth Tx Power Service Library are listed in the table below

| File | Contents/Description |
|---|---|
| TPSAPI.h | Tx Power Service (GATT based) API Type Definitions, Constants, and Prototypes. |
| TPSTYPES.h | Tx Power Service Types. |
| SS1BTTPS.h | Tx Power Service Include file |