Human Interface Device Service (HIDS)

Application Programming Interface Reference Manual

Profile Version: 1.0

Release: 4.0.1
January 10, 2013

Louisville, KY www.stonestreetone.com

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 Access Profile) Layer and the Generic Attribute Profile (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), HIDS (Human Interface Device 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 Human Interface Device Service Profile Stack provided by Bluetopia®+LE Single. And, Chapter 3 contains the header file name list for the Human Interface Device Service library.

1.1 Scope

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

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

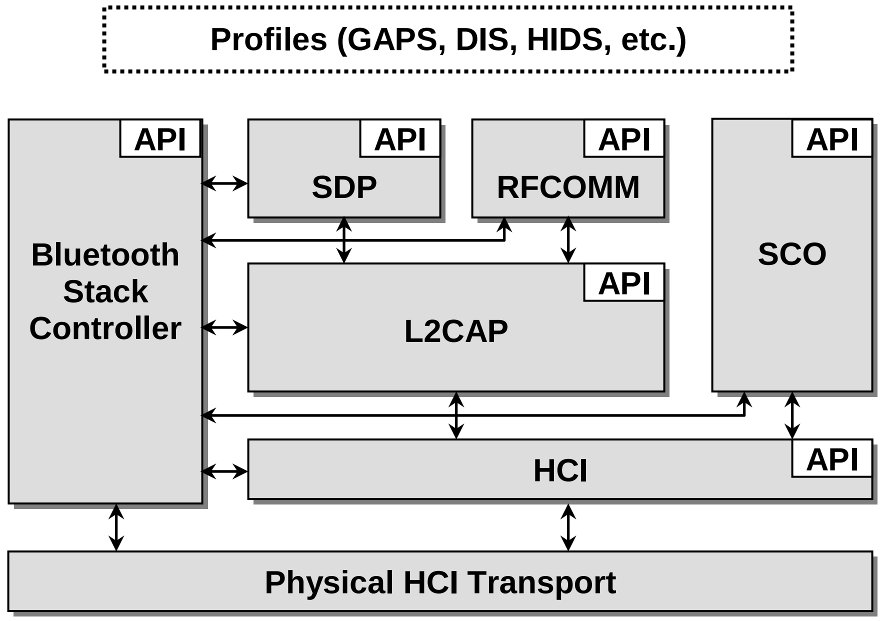

Figure 1-1 The Stonestreet One Bluetooth Protocol Stack

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 Human Interface Device Service Specification*, version v10r00, May 22, 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 Profile
HCI	Host Controller Interface
HIDS	Human Interface Device Service
HS	High Speed
L2CAP	Logical Link Control and Adaptation Protocol
LE	Low Energy
LSB	Least Significant Bit
MSB	Most Significant Bit

2. HIDS Programming Interface

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

2.1 Human Interface Device Service Commands

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

Server Commands	
Function	**Description**
HIDS_Initialize_Service	Opens a HIDS Server.
HIDS_Cleanup_Service	Closes an opened HIDS Server.
HIDS_Read_Client_Configuration_Response	Responds to a HIDS Read Client Configuartion Request.
HIDS_Get_Protocol_Mode_Response	Responds to a HIDS Get Protocol Mode Request.
HIDS_Get_Report_Map_Response	Responds to a HIDS Get Report Map Request.
HIDS_Get_Report_Response	Responds to a HIDS Get Report Map Request.
HIDS_Set_Report_Response	Repsonds to a HIDS Set Report Map Request.
HIDS_Notify_Input_Report	Sends an Input Report Notification to a specified remote device.
Client Commands	
Function	**Description**
HIDS_Decode_HID_Information	Parses a calue received form a remote HIDS server and interpret it as a HID Information value.
HIDS_Decode_Report_Reference	Parses a value received from a remote HIDS Server and interprets it as a Report Referecne value.
HIDS_Decode_External_Report_Reference	Parses a value received from a remote HIDS Server and interprets it as an

	External Report Reference value.
HIDS_Format_Protocol_Mode	Formats a HIDS Protocol Mode into a user specified buffer.
HIDS_Format_Control_Point_Command	Formats a HIDS Control Point Command into a user specified buffer.

HIDS_Initialize_Service

The following fuction is responsible for opening a HID Server over GATT Service.

Notes:

1. The Flags parameter must be a bit mask made of bits of the form HIDS_FLAGS_XXX.
2. The ServiceIDList parameter must contain valid ServiceIDs of services that have already been registered with GATT.

Prototype:

int BTPSAPI **HIDS_Initialize_Service**(unsigned int BluetoothStackID, Byte_t Flags, HIDS_HID_Information_Data_t *HIDInformation, unsigned int NumIncludedServices, unsigned int *ServiceIDList, unsigned int NumExternalReportReferences, GATT_UUID_t *ReferenceUUID, unsigned int NumReports, HIDS_Report_Reference_Data_t *ReportReference, HIDS_Event_Callback_t EventCallback, unsigned long CallbackParameter, unsigned int *ServiceID)

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
Flags	A bit mask of flags which is used to control what the HID Service supports.
HIDInformation	A pointer to a HID Information structure containing information tbout the HID Service. The HID Information Data structure is as follows:

```
typedef struct
{
   Word_t  Version;
   Byte_t       CountryCode;
   Byte_t       Flags;
} HIDS_HID_Information_Data_t;
```

NumIncludedServices	The number of Services that are included by this HID Instance.
ServiceIDList	A list of Service IDs that contain the Service IDS of the Services to be included by this HID Instance.

NumExternalReportReferences	The number of GATT UUIDs referenced by this HID Instance.
ReferenceUUID	A list of GATT UUIDs that contain a list of UUIDs characteristics referenced by this HID Instance.
NumReports	The number of reports that will be contained in this HID Instance.
ReportReference	A list of reports that will be contained in this HID Instance. The Report Reference Data structure is as follow:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t      ReportType;
} HIDS_Report_Reference_Data_t;
```

EventCallback	Callback function that is registered to receive events that are associated with the specified service.
CallbackParameter	A user-defined parameter that will be passed back to the user in the callback function.
ServiceID	Unique GATT Service ID of the registered HIDS service returned from GATT_Register_Service API.

Return:

Positive non-zero if successful. The return value will be the Service ID of HIDS 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:

HIDS_ERROR_INSUFFICIENT_RESOURCES
HIDS_ERROR_INVALID_BLUETOOTH_STACK_ID
HIDS_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.

HIDS_Cleanup_Service

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

Prototype:

int BTPSAPI **HIDS_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 HIDS_Initialize_Service().

Return:

Zero if successful.

Negative if an error occurred. Possible values are:

HIDS_ERROR_INVALID_PARAMETER
HIDS_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.

HIDS_Read_Client_Configuration_Response

The following function is responsible for responding to a HIDS Read Client Configuration Request.

Prototype:

int BTPSAPI **HIDS_Read_Client_Configuration_Response**(unsigned int
    BluetoothStackID, unsigned int InstanceID, unsigned int TransactionID,
    Word_t ClientConfiguration);

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 HIDS_Initialize_Service().

TransactionID	The Transaction ID of the original read request. This value was received in the etHIDS_Read_Client_Configuration_Request event.
ClientConfiguration	The Client Configuration to send to the remote device.

Return:

Zero if successful.

Negative if an error occurred. Possible values are:

HIDS_ERROR_INVALID_INSTANCE_ID
HIDS_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.

HIDS_Get_Server_Mode

The following function is responsible for responding to a HIDS Get Protocol Mode Request.

Prototype:

int BTPSAPI **HIDS_Get_Protocol_Mode_Response**(unsigned int BluetoothStackID, unsigned int InstanceID, unsigned int TransactionID, Byte_t ErrorCode, HIDS_Protocol_Mode_t CurrentProtocolMode);

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 HIDS_Initialize_Service().
TransactionID	The Transaction ID of the original read request. This value was received in the etHIDS_Read_Client_Configuration_Request event.
ErrorCode	ErrorCode is used to determine if the Request is being accepted by the server or if an error response is issued instead. This function returns a zero if successful or a negative return error code if an error occurs.
CurrentProtocolMode	This contains the Protocol Mode to respond with. The Protocol Mode enum is as follows:

```
typedef  enum
{
      pmBoot,
      pmReport
} HIDS_Protocol_Mode_t;
```

Return:

Zero if successful.

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

HIDS_ERROR_INVALID_INSTANCE_ID
HIDS_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.

HIDS_Get_Report_Map_Response

The following function is responsible for responding to a HIDS Get Report Map Request.

Prototype:

int BTPSAPI **HIDS_Get_Report_Map_Response**(unsigned int BluetoothStackID, unsigned int InstanceID, unsigned int TransactionID, Byte_t ErrorCode, unsigned int ReportMapLength, Byte_t *ReportMap);

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 HIDS_Initialize_Service().
TransactionID	The Transaction ID of the original read request. This value was received in the etHIDS_Read_Client_Configuration_Request event.
ErrorCode	ErrorCode is used to determine if the Request is being accepted by the server or if an error response is issued instead. This function returns a zero if successful or a negative return error code if an error occurs.
ReportMapLength	If ErrorCode is 0, this specifies the Report Map length to respond with.
ReportMap	If ErrorCode is 0, this speciafies the data of the Report Map to respond with.

Return:

Zero if successful.

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

HIDS_ERROR_INVALID_INSTANCE_ID
HIDS_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.

HIDS_Get_Report_Response

The following function is responsible for responding to a HIDS Get Report Map Request.

Prototype:

int BTPSAPI **HIDS_Get_Report_Response**(unsigned int BluetoothStackID, unsigned int InstanceID, unsigned int TransactionID, HIDS_Report_Type_t ReportType, HIDS_Report_Reference_Data_t *ReportReferenceData, Byte_t ErrorCode, unsigned int ReportLength, Byte_t *Report);

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 HIDS_Initialize_Service().
TransactionID	The Transaction ID of the original read request. This value was received in the etHIDS_Read_Client_Configuration_Request event.
ReportType	The Report Type that the client is trying to get. The Report Type enum is as follows:

```
typedef enum
{
    rtReport,
    rtBootKeyboardInputReport,
    rtBootKeyboardOutputReport,
    rtBootMouseInputReport
} HIDS_Report_Type_t;
```

ReportReferenceData	Only valid if ReportType is rtReport, the report reference data of the Report that ths client is attempting to get. The Report Reference Data structure is as follow:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t        ReportType;
} HIDS_Report_Reference_Data_t;
```

ErrorCode	ErrorCode is used to determine if the Request is being accepted by the server or if an error response is issued instead. This function returns a zero if successful or a negative return error code if an error occurs.
ReportLength	If ErrorCode is 0, this specifies the Report Map length to respond with.
Report	If ErrorCode is 0, this speciafies the data of the Report Map to respond with.

Return:

Zero if successful.

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

HIDS_ERROR_INVALID_INSTANCE_ID
HIDS_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.

HIDS_Set_Report_Response

This function is responsible for responding to a HIDS Set Report Map Request.

Prototype:

int BTPSAPI **HIDS_Set_Report_Response**(unsigned int BluetoothStackID, unsigned int InstanceID, unsigned int TransactionID, HIDS_Report_Type_t ReportType, HIDS_Report_Reference_Data_t *ReportReferenceData, Byte_t ErrorCode);

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 HIDS_Initialize_Service().
TransactionID	The Transaction ID of the original read request. This value was received in the etHIDS_Read_Client_Configuration_Request event.
ReportType	The Report Type that the client is trying to set. The Report Type enum is as follows:

```
typedef enum
{
        rtReport,
        rtBootKeyboardInputReport,
        rtBootKeyboardOutputReport,
        rtBootMouseInputReport
    } HIDS_Report_Type_t;
```

ReportReferenceData — Only valid if ReportType is rtReport, the report reference data of the Report that ths client is attempting to set. The Report Reference Data structure is as follow:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t      ReportType;
} HIDS_Report_Reference_Data_t;
```

ErrorCode — ErrorCode is used to determine if the Request is being accepted by the server or if an error response is issued instead. This function returns a zero if successful or a negative return error code if an error occurs.

Return:

Zero if successful.

Negative if an error occurred. Possible values are:

HIDS_ERROR_INVALID_INSTANCE_ID
HIDS_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.

HIDS_Notify_Input_Report

The following function is responsible for sending an Input Report notification to a specified remote device.

Prototype:

int BTPSAPI **HIDS_Notify_Input_Report**(unsigned int BluetoothStackID, unsigned int InstanceID, unsigned int ConnectionID, HIDS_Report_Type_t ReportType, HIDS_Report_Reference_Data_t *ReportReferenceData, Word_t InputReportLength, Byte_t *InputReportData)

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 HIDS_Initialize_Service().
ConnectionID	Connection ID of the currently connected remote client device to send the handle/value notification.
ReportType	The Report Type that the client is trying to be notified. The Report Type enum is as follows:

```
typedef enum
{
    rtReport,
    rtBootKeyboardInputReport,
    rtBootKeyboardOutputReport,
    rtBootMouseInputReport
} HIDS_Report_Type_t;
```

ReportReferenceData	A pointer to a Report Reference structure that is only used (and must be specified only if) the ReportType is reInputReport. The Report Reference Data structure is as follow:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t      ReportType;
} HIDS_Report_Reference_Data_t;
```

InputReportLength	The length of the Input Report.
InputReportData	A pointer to the Input Report that is to be notified.

Return:

Zero if successful.

Negative if an error occurred. Possible values are:

```
HIDS_ERROR_INVALID_INSTANCE_ID
HIDS_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.

HIDS_Decode_HID_Information

The following function is responsible for parsing a value received from a remote HIDS Server interpreting it as a HID Information value.

Prototype:

int BTPSAPI **HIDS_Decode_HID_Information**(unsigned int ValueLength, Byte_t *Value, HIDS_HID_Information_Data_t *HIDSHIDInformation);

Parameters:

ValueLength	Specifies the length of the value returned by the remote HIDS Server.
Value	Value is a pointer to the data returned by the remote HIDS Server.
HIDSHIDInformation	A pointer to store the parsed HID Information value (if successful). The HID Information Data structure is as follows:

```
typedef struct
{
    Word_t    Version;
    Byte_t    CountryCode;
    Byte_t    Flags;
} HIDS_HID_Information_Data_t;
```

Return:

Zero if successful.

Negative if an error occurred. Possible values are:

HIDS_ERROR_MALFORMATTED_DATA
HIDS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER

HIDS_Decode_Report_Reference

The following function is responsible for parsing a value received from a remote HIDS Server interpreting it as a Report Reference value.

Prototype:

int BTPSAPI **HIDS_Decode_Report_Reference**(unsigned int ValueLength, Byte_t *Value, HIDS_Report_Reference_Data_t *ReportReferenceData);

Parameters:

ValueLength	Specifies the length of the value returned by the remote HIDS Server.
Value	Value is a pointer to the data returned by the remote HIDS Server.

ReportReferenceData	A pointer to store the parsed Report Reference data (if successful). The Report Reference Data structure is as follows:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t      ReportType;
} HIDS_Report_Reference_Data_t;
```

Return:

Zero if successful.

Negative if an error occurred. Possible values are:

```
HIDS_ERROR_MALFORMATTED_DATA
HIDS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER
```

HIDS_Decode_External_Report_Reference

The following function is responsible for parsing a value received from a remote HIDS Server interpreting it as a External Report Reference value.

Prototype:

int BTPSAPI **HIDS_Decode_External_Report_Reference**(unsigned int ValueLength, Byte_t *Value, GATT_UUID_t *ExternalReportReferenceUUID);

Parameters:

ValueLength	Specifies the length of the value returned by the remote HIDS Server.
Value	Value is a pointer to the data returned by the remote HIDS Server.
ReportReferenceData	A pointer to store the parsed External Report Reference data (if successful). The GATT UUID structure is as follows:

```
typedef struct _tagGATT_UUID_t
{
    GATT_UUID_Type_t   UUID_Type;
    union
    {
        UUID_16_t      UUID_16;
        UUID_128_t     UUID_128;
    } UUID;
} GATT_UUID_t;
```

With the GATT UUID Type enum being defied as follows:

```
typedef enum
{
    guUUID_16,
    guUUID_128
} GATT_UUID_Type_t;
```

Return:

Zero if successful.

Negative if an error occurred. Possible values are:

HIDS_ERROR_MALFORMATTED_DATA
HIDS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER

HIDS_Format_Protocol_Mode

The following function is responsible for formatting a HIDS Protocol Mode into a user specified buffer.

Prototype:

int BTPSAPI **HIDS_Format_Protocol_Mode**(HIDS_Protocol_Mode_t ProtocolMode,
    unsigned int BufferLength, Byte_t *Buffer);

Parameters:

ProtocolMode	This is the user specified command to format. The Protocol Mode enum is as follows: typedef enum { pmBoot, pmReport } **HIDS_Protocol_Mode_t**;
BufferLength	Specifies the Length of the Buffer. The buffer must be of at least HIDS_PROTOCOL_MODE_VALUE_LENGTH in length.
Buffer	A pointer to the buffer to format the Protocol Mode into. The buffer must be of at least HIDS_PROTOCOL_MODE_VALUE_LENGTH in size.

Return:

Zero if successful.

Negative if an error occurred. Possible values are:

HIDS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER

HIDS_Format_Control_Point_Command

The following function is responsible for formatting a HIDS Control Point Command into a user specified buffer.

Prototype:

int BTPSAPI **HIDS_Format_Control_Point_Command**
(HIDS_Control_Point_Command_t Command, unsigned int BufferLength, Byte_t *Buffer);

Parameters:

Command	The command to format. The Control Point Command enum is as follows:

```
typedef enum
{
    pcSuspend,
    pcExitSuspend
} HIDS_Control_Point_Command_t;
```

BufferLength	Specifies the Length of the Buffer. The buffer must be of at least HIDS_PROTOCOL_MODE_VALUE_LENGTH in length.
Buffer	A pointer to the buffer to format the Protocol Mode into. The buffer must be of at least HIDS_PROTOCOL_MODE_VALUE_LENGTH in size.

Return:

Zero if successful.

Negative if an error occurred. Possible values are:
HIDS_ERROR_INVALID_PARAMETER
BTGATT_ERROR_NOT_INITIALIZED
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_INVALID_PARAMETER

2.2 Human Interface Device Service Event Callback Prototypes

2.2.1 Server Event Callback

The event callback function mentioned in the HIDS_Initialize_Service command accepts the callback function described by the following prototype.

HIDS_Event_Callback_t

This The event callback function mentioned in the HIDS_Initialize_Service command accepts the callback function described by the following prototype.

Note:

This function MUST NOT Block and wait for events that can only be satisfied by Receiving HID Service Event Packets. A Deadlock WILL occur because NO HIDS Event Callbacks will be issued while this function is currently outstanding.

Prototype:

typedef void (BTPSAPI ***HIDS_Event_Callback_t**)(unsigned int BluetoothStackID, HIDS_Event_Data_t *HIDS_Event_Data, unsigned long CallbackParameter);

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
HIDS_Event_Data_t	Data describing the event for which the callback function is called. This is defined by the following structure:

```
typedef struct
{
    HIDS_Event_Type_t       Event_Data_Type;
    Word_t                  Event_Data_Size;
    union
    {
        HIDS_Read_Client_Configuration_Data_t
            *HIDS_Read_Client_Configuration_Data;
        HIDS_Client_Configuration_Update_Data_t
            *HIDS_Client_Configuration_Update_Data;
        HIDS_Get_Protocol_Mode_Request_Data_t
            *HIDS_Get_Protocol_Mode_Request_Data;
        HIDS_Set_Protocol_Mode_Request_Data_t
            *HIDS_Set_Protocol_Mode_Request_Data;
        HIDS_Get_Report_Map_Request_Data_t
            *HIDS_Get_Report_Map_Data;
        HIDS_Get_Report_Request_Data_t
            *HIDS_Get_Report_Request_Data;
        HIDS_Set_Report_Request_Data_t
            *HIDS_Set_Report_Request_Data;
        HIDS_Control_Point_Command_Data_t
            *HIDS_Control_Point_Command_Data;
    } Event_Data;
} HIDS_Event_Data_t;
```

Where, Event_Data_Type is one of the enumerations of the event types listed in the table in section 2.3, and each data structure in the union is described with its event in that section as well.

CallbackParameter	User-defined parameter that was defined in the callback registration.

Return:

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.

2.3 Human Interface Device Service Events

The Human Interface Device Service contains events that are received by the Server. The following sections detail those events.

2.3.1 Human Interface Device Service Server Events

The possible Human Interface Device Service Server Events from the Bluetooth stack are listed in the table below and are described in the text which follows:

Server Commands	
Function	**Description**
etHIDS_Read_Client_Configuration_Request	Dispatched to a HIDS Server when a HIDS Client is attempting to read a descriptor.
etHIDS_Server_Client_Configuration_Update_ Request	Dispathed to a HIDS Server when a HIDS Client is writing a Client Configuration descriptor.
etHIDS_Server_Get_Protocol_Mode_Request	Dispathced to a HIDS Server when a HIDS Client is attempting to get the current Protocol Mode.
etHIDS_Server_Set_Protocol_Mode_Request	Dispathced to a HIDS Server when a HIDS Client is attempting to set the current Protocol Mode.
etHIDS_Server_Get_Report_Map_Request	Dispathced to a HIDS Server when a HIDS Client is attempting to get the Report Map value.
etHIDS_Server_Get_Report_Request	Dispathced to a HIDS Server when a HIDS Client is attempting to get the specified Report value.
etHIDS_Server_Set_Report_Request	Dispathced to a HIDS Server when a HIDS Client is attempting to set the Report value.

etHIDS_Server_Control_Point_Command_ Indication	Dispathced to a HIDS Server in response to the reception of a request from a Client to write the Control Point Command.

etHIDS_Read_Client_Configuartion_Request

The following HIDS Profile Event is dispatched to a HIDS Server when a HIDS Client is attempting to read a descriptor.

Note:

Only the following characteristic types may be returned in this event: rtReport (Input Report Type Only), reBootKeyboardInputReport, and rtBootMouseInputReport.

Return Structure:

```
typedef struct
{
    unsigned int                      InstanceID;
    unsigned int                      ConnectionID;
    unsigned int                      TransactionID;
    GATT_Connection_Type_t            ConnectionType;
    BD_ADDR_t                         RemoteDevice;
    HIDS_Report_Type_t                ReportType;
    HIDS_Report_Reference_Data_t      ReportReferenceData;
} HIDS_Read_Client_Configuration_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected.
ConnectionID	Connection ID of the currently connected remote HIDS server device.
TransactionID	The TransactionID identifies the transaction between a client and server. This identifier should be used to respond to the current request.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.
RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.
ReportType	Specifies the Descriptor that the Client is attempting to read. The Report Type enum is defined as follows:

```
typedef enum
{
    rtReport,
    rtBootKeyboardInputReport,
    rtBootKeyboardOutputReport,
    rtBootMouseInputReport
} HIDS_Report_Type_t;
```

ReportReferenceData	A report reference structure (only valid if ReportType is set to rtReport) that contains the Report ID and Report type of the characteristic value whose CCCD is being read. The Report Reference Data structure is as follow:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t      ReportType;
} HIDS_Report_Reference_Data_t;
```

etHIDS_Server_Client_Configuration_Update_Request

The following HIDS Profile Event is dispatched to a HIDS Server when a HIDS Client has written a Client Configuration descriptor.

Return Structure:

```
typedef struct
{
    unsigned int                    InstanceID;
    unsigned int                    ConnectionID;
    GATT_Connection_Type_t          ConnectionType;
    BD_ADDR_t                       RemoteDevice;
    HIDS_Report_Type_t              ReportType;
    HIDS_Report_Reference_Data_t    ReportReferenceData;
    Word_t                          ClientConfiguration;
} HIDS_Client_Configuration_Update_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected.
ConnectionID	Connection ID of the currently connected remote HIDS server device.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.
RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.
ReportType	Specifies the descriptor that the Client is writing. The Report Type enum is defined as follows:

```
typedef enum
{
    rtReport,
    rtBootKeyboardInputReport,
    rtBootKeyboardOutputReport,
    rtBootMouseInputReport
} HIDS_Report_Type_t;
```

ReportReferenceData	A report reference structure (Only valid if the Report Type is set to rtReport) that contains the Report ID and the report type of the characteristic value whose CCCD is being read. The Report Reference Data structure is as follow:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t      ReportType;
} HIDS_Report_Reference_Data_t;
```

ClientConfiguration	The New Client Configuration for the specified characteristic.

etHIDS_Server_Get_Protocol_Mode_Request

The following HIDS Profile Event is dispatched to a HIDS Server when a HIDS client is attempting to get the current Protocol Mode.

Return Structure:

```
typedef struct
{
    unsigned int              InstanceID;
    unsigned int              ConnectionID;
    unsigned int              TransactionID;
    GATT_Connection_Type_t    ConnectionType;
    BD_ADDR_t                 RemoteDevice;
} HIDS_Get_Protocol_Mode_Request_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected.
ConnectionID	Connection ID of the currently connected remote HIDS server device.
TransactionID	The TransactionID identifies the transaction between a client and server. This identifier should be used to respond to the current request.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.
RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.

etHIDS_Server_Set_Protocol_Mode_Request

The following HIDS Profile Event is dispatched to a HIDS Server when a HIDS Client is attempting to set the current Protocol Mode.

Return Structure:

```
typedef struct
```

```
{
    unsigned int                 InstanceID;
    unsigned int                 ConnectionID;
    unsigned int                 TransactionID;
    GATT_Connection_Type_t       ConnectionType;
    BD_ADDR_t                    RemoteDevice;
    HIDS_Protocol_Mode_t         ProtocolMode;
} HIDS_Set_Protocol_Mode_Request_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected..
ConnectionID	Connection ID of the currently connected remote HIDS server device.
TransactionID	The TransactionID identifies the transaction between a client and server. This identifier should be used to respond to the current request.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.
RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.
ProtocolMode	The Protocol Mode that the HIDS client is attempting to set.The Protocol Mode structure is as follows:

```
typedef enum
{
    pmBoot,
    pmReport
} HIDS_Protocol_Mode_t;
```

etHIDS_Server_Get_Report_Map_Request

The following HIDS Profile Event is dispatched to a HIDS Server when a HIDS Client is attempting to get the Report Map value.

Return Structure:

```
typedef struct
{
    unsigned int                 InstanceID;
    unsigned int                 ConnectionID;
    unsigned int                 TransactionID;
    GATT_Connection_Type_t       ConnectionType;
    BD_ADDR_t                    RemoteDevice;
    Word_t                       ReportMapOffset;
} HIDS_Get_Report_Map_Request_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected..
ConnectionID	Connection ID of the currently connected remote HIDS server device.
TransactionID	The TransactionID identifies the transaction between a client and server. This identifier should be used to respond to the current request.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.
RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.
ReportMapOffest	The offset into the Report Map that HIDS Client is attempting to read.

etHIDS_Server_Get_Report_Request

The following HIDS Profile Event is dispatched to a HIDS Server when a HIDS Client is attempting to get a Report value.

Return Structure:

```
typedef struct
{
   unsigned int                    InstanceID;
   unsigned int                    ConnectionID;
   unsigned int                    TransactionID;
   GATT_Connection_Type_t          ConnectionType;
   BD_ADDR_t                       RemoteDevice;
   Word_t                          ReportOffest;
   HIDS_Report_Type_t              ReportType;
   HIDS_Report_Reference_Data_t    ReportReferenceData;
} HIDS_ Get_Report_Request_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected..
ConnectionID	Connection ID of the currently connected remote HIDS server device.
TransactionID	The TransactionID identifies the transaction between a client and server. This identifier should be used to respond to the current request.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.

RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.
ReportOffset	The offset into the Report that HIDS Client is attempting to read.
ReportType	Specifies the report that the HIDS Client is attempting to get. The Report Type enum is as follows:

```
typedef enum
{
    rtReport,
    rtBootKeyboardInputReport,
    rtBootKeyboardOutputReport,
    rtBootMouseInputReport
} HIDS_Report_Type_t;
```

ReportReferenceData	A report reference structure (Only valid if the ReportType is set to rtReport) that contains the Report ID and Report Type of the Report that is being read. The Report Reference Data structure is as follow:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t      ReportType;
} HIDS_Report_Reference_Data_t;
```

etHIDS_Server_Set_Report_Request

The following HIDS Profile Event is dispatched to a HIDS Server when a HIDS Client is attempting to set the Report value.

Return Structure:

```
typedef struct
{
    unsigned int                    InstanceID;
    unsigned int                    ConnectionID;
    unsigned int                    TransactionID;
    GATT_Connection_Type_t          ConnectionType;
    BD_ADDR_t                       RemoteDevice;
    HIDS_Report_Type_t              ReportType;
    HIDS_Report_Reference_Data_t    ReportReferenceData;
    unsigned int                    ReportLength;
    Byte_t                         *Report
} HIDS_Set_Report_Request_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected..

ConnectionID	Connection ID of the currently connected remote HIDS server device.
TransactionID	The TransactionID identifies the transaction between a client and server. This identifier should be used to respond to the current request.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.
RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.
ReportType	Specifies the report that the HIDS Client is attempting to set. The Report Type enum is as follows:

```
typedef enum
{
    rtReport,
    rtBootKeyboardInputReport,
    rtBootKeyboardOutputReport,
    rtBootMouseInputReport
} HIDS_Report_Type_t;
```

ReportReferenceData	A report reference structure (Only valid if the ReportType is set to rtReport) that contains the Report ID and Report Type of the Report that is being written. The Report Reference Data structure is as follow:

```
typedef struct
{
    Byte_t      ReportID;
    Byte_t      ReportType;
} HIDS_Report_Reference_Data_t;
```

ReportLength	The length of the data that the HIDS Client is attempting to write.
Report	A pointer to the data that the HIDS Client is attempting to write.

etHIDS_Server_Control_Point_Command_Indication

The following is dispatched to a HIDS Server in response to the reception of a request from a Client to write to the Control Point Command.

Return Structure:

```
typedef struct
{
    unsigned int                    InstanceID;
    unsigned int                    ConnectionID;
    GATT_Connection_Type_t          ConnectionType;
    BD_ADDR_t                       RemoteDevice;
    HIDS_Control_Point_Command_t    ControlPointCommand
} HIDS_Read_Reference_Time_Information_Request_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected..
ConnectionID	Connection ID of the currently connected remote HIDS server device.
ConnectionType	Identifies the type of remote Bluetooth device that is connected. Currently this value will be gctLE only.
RemoteDevice	Specifies the address of the Client Bluetooth device that has connected to the specified Server.
ControlPointCommand	The Control Point Command that the Client has requested to write to. The Control Point Command enum is as follows:

```
typedef enum
{
   pcSuspend,
   pcExitSuspend
} HIDS_Control_Point_Command_t;
```

3. File Distributions

The header files that are distributed with the Bluetooth Human Interface Device Service Library are listed in the table below

File	Contents/Description
HIDSAPI.h	Bluetooth Human Interface Device Service (GATT based) API Type Definitions, Constants, and Prototypes.
HIDSTypes.h	Bluetooth Human Interface Device Service Types.
SS1BTHIDS.h	Bluetooth Human Interface Device Service Include file