



Immediate Alert Service (IAS)

Application Programming Interface Reference Manual

Profile Version: 1.0

Release: 4.0.1
January 10, 2014



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 © 2000-2014 by Stonestreet One, LLC. All rights reserved.

Table of Contents

1. INTRODUCTION.....	3
1.1 Scope	3
1.2 Applicable Documents	4
1.3 Acronyms and Abbreviations	4
2. IMMEDIATE ALERT SERVICE PROGRAMMING INTERFACES.....	5
2.1 Immediate Alert Service Commands	5
IAS_Initialize_Service.....	5
IAS_Initialize_Service_Handle_Range	6
IAS_Cleanup_Service.....	7
IAS_Query_Number_Attributes	7
IAS_Format_Control_Point_Command	8
2.2 Immediate Alert Service Event Callback Prototypes	8
2.2.1 SERVER EVENT CALLBACK	8
IAS_Event_Callback_t	8
2.3 Immediate Alert Service Events	9
2.3.1 IMMEDIATE ALERT SERVICE SERVER EVENTS	9
etIAS_Server_Alert_Level_Control_Point_Command	9
3. FILE DISTRIBUTIONS.....	11

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 Generic 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), IAS (Immediate Alert 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 Immediate Alert Service Profile Stack provided by Bluetopia®+LE Single. And, Chapter 3 contains the header file name list for the Immediate Alert Service Profile library.

1.1 Scope

This reference manual provides information on the APIs identified in Figure 1-1 below. These APIs are available on the full range of platforms supported by Stonestreet One:

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

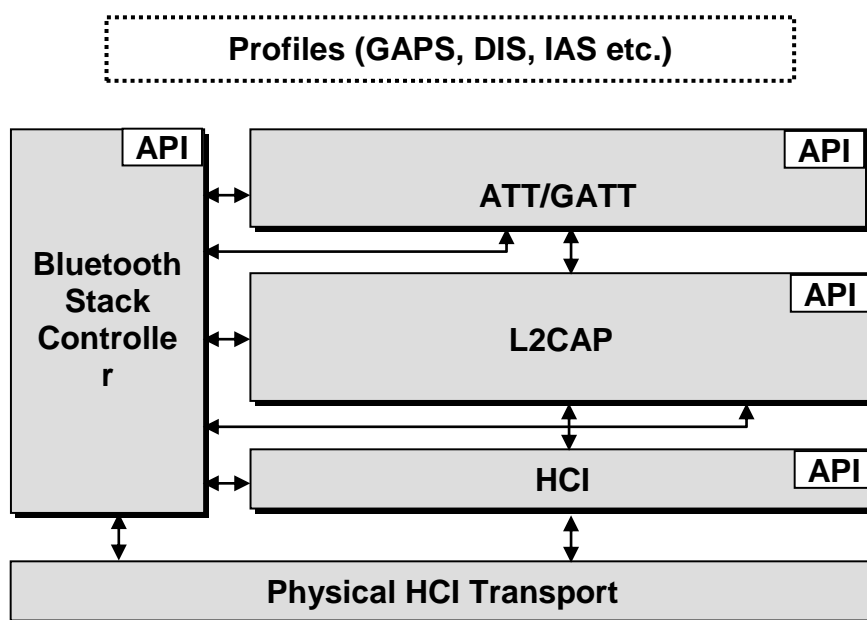


Figure 1-1 The Stonestreet One Bluetooth Protocol Stack

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 Doc Immediate Alert Service Specification*, version v10r00, June 21, 2011

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
IAS	Immediate Alert Service
BD_ADDR	Bluetooth Device Address
BT	Bluetooth
DIS	Device Information Service
GATT	Generic Attribute Protocol
GAPS	Generic Access Profile Service
HCI	Host Controller Interface
HS	High Speed
L2CAP	Logical Link Control and Adaptation Protocol
LE	Low Energy

2. Immediate Alert Service Programming Interfaces

The Immediate Alert Service programming interface defines the protocols and procedures to be used to implement Immediate Alert Service capabilities. The Immediate Alert Service commands are listed in section 2.1, the event callback prototypes are described in section 2.2, and the Immediate Alert Service events are itemized in section 2.3. The actual prototypes and constants outlined in this section can be found in the **IASAPI.H** header file in the Bluetopia distribution.

2.1 Immediate Alert Service Commands

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

Function	Description
IAS_Initialize_Service	Opens an IAS Server.
IAS_Initialize_Service_Handle_Range	Opens an IAS Server with the ability to control the location of the service in the GATT database.
IAS_Cleanup_Service	Closes an opened IAS Server.
IAS_Query_Number_Attributes	Queries the number of attributes.
IAS_Query_Time_Update_State	Gets the Time Update State from the specified IAS Instance
IAS_Format_Control_Point_Command	Responsible for formatting an Alert Level Control Point Command into a user specified buffer

IAS_Initialize_Service

This function opens an IAS Server on a specified Bluetooth Stack.

Prototype:

```
int BTPSAPI IAS_Initialize_Service (unsigned int BluetoothStackID,  
    IAS_Event_Callback_t EventCallback, unsigned long CallbackParameter, unsigned int  
    *ServiceID);
```

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
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 IAS service returned from GATT_Register_Service API
-----------	--

Return:

Positive, non-zero if successful. The return value will be the Service Instance ID of IAS 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.

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

- IAS_ERROR_INSUFFICIENT_RESOURCES
- IAS_ERROR_INVALID_PARAMETER
- IAS_ERROR_SERVICE_ALREADY_REGISTERED
- BTGATT_ERROR_INVALID_SERVICE_TABLE_FORMAT
- BTGATT_ERROR_INSUFFICIENT_RESOURCES
- BTGATT_ERROR_INVALID_PARAMETER
- BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
- BTGATT_ERROR_NOT_INITIALIZED

Possible Events:**IAS_Initialize_Service_Handle_Range**

This function is responsible for opening an IAS Server with the ability to control the location of the service in the GATT database.

Prototype:

```
int BTPSAPI IAS_Initialize_Service_Handle_Range(unsigned int BluetoothStackID,  
        IAS_Event_Callback_t EventCallback, unsigned long CallbackParameter,  
        unsigned int *ServiceID, GATT_Attribute_Handle_Group_t *ServiceHandleRange);
```

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
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 IAS service returned from GATT_Register_Service API
ServiceHandleRange	Pointer to a Service Handle Range structure, that on input can be used to control the location of the service in the GATT database, and on output returns the handle range that the service is using in the GATT database.

Return:

Positive, non-zero if successful. The return value will be the Service Instance ID of IAS 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.

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

IAS_ERROR_INSUFFICIENT_RESOURCES
IAS_ERROR_INVALID_PARAMETER
IAS_ERROR_SERVICE_ALREADY_REGISTERED
BTGATT_ERROR_INVALID_SERVICE_TABLE_FORMAT
BTGATT_ERROR_INSUFFICIENT_RESOURCES
BTGATT_ERROR_INVALID_PARAMETER
BTGATT_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGATT_ERROR_NOT_INITIALIZED

Possible Events:**IAS_Cleanup_Service**

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

Prototype:

```
int BTPSAPI IAS_Cleanup_Service(unsigned int BluetoothStackID, unsigned int InstanceID);
```

Parameters:

BluetoothStackID	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
InstanceID	The Service Instance ID to close. This is the value that was returned from the IAS_Initialize_Service() function.

Return: Zero if successful.

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

IAS_ERROR_INVALID_PARAMETER
IAS_ERROR_INVALID_INSTANCE_ID

Possible Events:**IAS_Query_Number_Attributes**

This function is responsible for querying the number of attributes that are contained in the IAS Service that is registered with a call to IAS_Initialize_Service() or IAS_Initialize_Service_Handle_Range().

Prototype:

```
unsigned int BTPSAPI IAS_Query_Number_Attributes(void)
```

Parameters:**Return:** Zero if successful.

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

IAS_ERROR_INVALID_PARAMETER
IAS_ERROR_INVALID_INSTANCE_ID

Possible Events:**IAS_Format_Control_Point_Command**

This function is responsible for formatting Alert Level Control Point Command into a user specified buffer.

Prototype:

```
int BTPSAPI IAS_Format_Control_Point_Command(IAS_Control_Point_Command_t  
Command, unsigned int BufferLength, Byte_t *Buffer);
```

Parameters:

IAS_Control_Point_Command_t Valid Commands for Immediate Alert Level.
BufferLength Specifies the length of the user specified buffer.
Buffer Specifies the user specified buffer to format the data into.

Return:

Zero if successful.

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

IAS_ERROR_INVALID_PARAMETER

Possible Events:

2.2 Immediate Alert Service Event Callback Prototypes

2.2.1 Server Event Callback

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

IAS_Event_Callback_t

Prototype of callback function passed in the IAS_Initialize_Service command.

Prototype:

```
typedef void (BTPSAPI *IAS_Event_Callback_t)(unsigned int BluetoothStackID,  
IAS_Event_Data_t *IAS_Event_Data, unsigned long CallbackParameter);
```


Parameters:

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

```
typedef struct _tagIAS_Event_Data_t
{
    IAS_Event_Type_t Event_Data_Type;
    Byte_t          Event_Data_Size;
    union
    {
        IAS_Alert_Level_Control_Point_Command_Data_t
            *IAS_Alert_Level_Control_Point_Command_Data;
    } Event_Data;
} IAS_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:

2.3 Immediate Alert Service Events

The Immediate Alert Service contains events that are received by the Server. The following sections detail those events.

2.3.1 Immediate Alert Service Server Events

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

Event	Description
etIAS_Server_Alert_Level_Control_Point_Command	Dispatched when an IAS Client requests write Alert Level Control Point to a registered IAS Server.

etIAS_Server_Alert_Level_Control_Point_Command

Dispatched when an IAS Client has sent write Alert Level Control Point Command to a registered IAS Server.

Return Structure:

```
typedef struct _tagIAS_Alert_Level_Control_Point_Command_Data_t
{
    unsigned int          InstanceID;
    unsigned int          ConnectionID;
    GATT_Connection_Type_t ConnectionType;
    BD_ADDR_t             RemoteDevice;
    IAS_Control_Point_Command_t Command;
} IAS_Alert_Level_Control_Point_Command_Data_t;
```

Event Parameters:

InstanceID	Identifies the Local Server Instance to which the Remote Client has connected.
ConnectionID	Identifier that uniquely identifies the actual connection of remote device that is making the 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.
Command	Specifies the Control Point Command. The values will be cpNoAlert, cpMildAlert, cpHighAlert only.

3. File Distributions

The header files that are distributed with the Bluetooth Immediate Alert Service Library are listed in the table below.

File	Contents/Description
IASAPI.h	Bluetooth Immediate Alert Service (GATT based) API Type Definitions, Constants, and Prototypes.
IASTypes.h	Bluetooth Immediate Alert Service Types.
SS1BTIAS.h	Bluetooth Immediate Alert Service Include file