Bluetopia®

Bluetooth™ Protocol Stack

Supported Commands/Events Application Programming Interface Reference Manual

Protocol Version: 2.1 + EDR

Release: 2.1.3
May 24, 2011

stonestreet one
Louisville, KY www.stonestreetone.com

Table of Contents

1. Introduction

Bluetopia®, the Bluetooth Protocol Stack by Stonestreet One, provides a software architecture that encapsulates the upper functionality of the Bluetooth Protocol Stack. 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) and the SCO/eSCO (Synchronous Connection-Oriented) Link layers. In addition to basic functionality at these layers, Bluetopia by Stonestreet One provides implementations of the Service Discovery Protocol (SDP), RFCOMM (the Radio Frequency serial COMMunications port emulator), 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 documents, and a listing of acronyms and abbreviations. Chapter 2 is the API reference which contains a description of all programming interfaces for Bluetopia. Chapter 3 contains a description of the programming interfaces for the profiles contained in the core Bluetooth Protocol Stack library.

1.1 Scope

This reference manual provides information on the Bluetopia supported Command/Event APIs identified in Figure 1-1 below. These APIs are available on the full range of embedded platforms supported by Stonestreet One. This information is supplied because various embedded devices may have varying RAM/ROM/FLASH requirements that might preclude the support of the entire Bluetopia API.

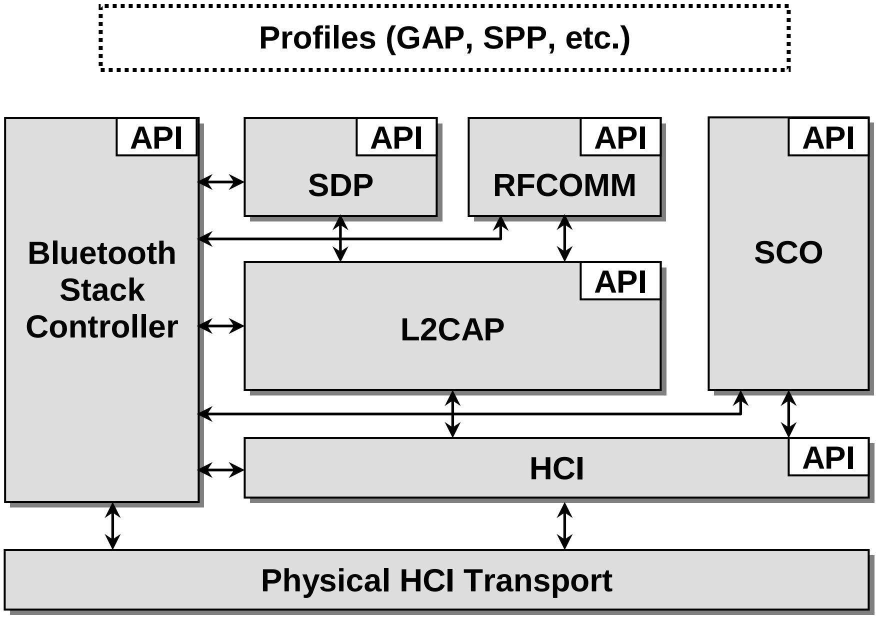

Figure 1-1 The Stonestreet One Bluetooth Protocol Stack

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, Core*, version 1.1, February 22, 2001.
2. *Specification of the Bluetooth System, Volume 2, Profiles*, version 1.1, February 22, 2001.
3. *Specification of the Bluetooth System, Volume 1, Architecture and Terminology Overview*, version 1.2, November 5, 2003.
4. *Specification of the Bluetooth System, Volume 2, Core System Package*, version 1.2, November 5, 2003.
5. *Specification of the Bluetooth System, Volume 3, Core System Package*, version 1.2, November 5, 2003.
6. *Specification of the Bluetooth System, Volume 1, Architecture and Terminology Overview*, version 2.0 + EDR, November 4, 2004.
7. *Specification of the Bluetooth System, Volume 2, Core System Package*, version 2.0 + EDR, November 4, 2004.
8. *Specification of the Bluetooth System, Volume 3, Core System Package*, version 2.0 + EDR, November 4, 2004.
9. *Specification of the Bluetooth System, Volume 0, Master Table of Contents & Compliance Requirements*, version 2.1+EDR, July 26, 2007.
10. *Specification of the Bluetooth System, Volume 1, Architecture and Terminology Overview*, version 2.1+EDR, July 26, 2007.
11. *Specification of the Bluetooth System, Volume 2, Core System Package [Controller Volume]*, version 2.1+EDR, July 26, 2007.
12. *Specification of the Bluetooth System, Volume 3, Core System Package [Host Volume]*, version 2.1+EDR, July 26, 2007.
13. *Specification of the Bluetooth System, Volume 4, Host Controller Interface*, version 2.1+EDR, July 26, 2007.
14. *Specification of the Bluetooth System, Bluetooth Core Specification Addendum 1*, June 26, 2008.
15. *Bluetooth Assigned Numbers*, version 1.1, February 22, 2001.
16. *Digital cellular telecommunications system (Phase 2+); Terminal Equipment to Mobile Station (TE-MS) multiplexer protocol (GSM 07.10)*, version 7.1.0, Release 1998; commonly referred to as: ETSI TS 07.10.
17. *Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX) with Published Errata*, Version 1.2, April 1999.

The Bluetooth Protocol Stack API calls were developed to closely follow the above specifications. Note that in previous versions of this document, the Bluetooth section that was directly applicable to the specified functionality was referenced. With the advent of newer versions of the Bluetooth Specification being served by this document, multiple references would need to be given for the specified function. Because of this, the section references have been dropped from this document. The reader should therefore consult the correct Bluetooth Core specification and determine the applicable section manually. In almost all cases, the determination of the section can easily be found by examining the table of contents of the core specification.

1.3 Acronyms and Abbreviations

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

Term	Meaning
ACL link	Asynchronous Connection-less Link – Provides a packet-switched connection. (Master to any slave)
API	Application Programming Interface
BD_ADDR	Bluetooth Device Address
BSC	Bluetooth Stack Controller
BT	Bluetooth
CID	Channel Identifier
dB	Decibels
DH	Data-High Rate Data packet type for high rate data
DLCI	Data Link Connection Identifier
DM	Data - Medium Rate Data packet type for medium rate data
DUT	Device Under Test
DV	Data Valid (serial interface signal)
DV	Data Voice data packet type for data and voice
ETSI	European Telecommunications Standards Institute
FC	Flow Control (serial interface signal)
FCC	Federal Communications Commission
GAP	Generic Application Profile
HCI	Host Controller Interface
HV	High quality Voice e.g. HV1 packet

Term	Meaning
IAC	Inquiry Access Code
IC	Incoming Call indicator (serial interface signal)
ID	Identifier
L2CA	Logical Link Control and Adaptation Logical Link Control And Management part of the Bluetooth protocol stack
L2CAP	Logical Link Control and Adaptation Protocol
LAP	Lower Address Part (of Bluetooth Device address)
LCID	Local Channel Identifier
LE	Low Energy
LM	Link Manager
LMP	Link Manager Protocol For LM peer to peer communication
LSB	Least Significant Bit
MSB	Most Significant Bit
MSC	Message Sequence Chart
MTU	Maximum Transmission Unit
NAP	Non-significant Address Part
OCF	Opcode Command Field
OGF	Opcode Group Field
PDU	Protocol Data Unit (a message)
PIN	Personal Identification Number
PSM	Protocol/Service Multiplexer
QoS	Quality of Service
RFCOMM	Radio Frequency serial COMMunications – Serial cable emulation protocol based on ETSI TS 07.10
RSSI	Received Signal Strength Indication
RTC	Ready to Communicate (serial interface signal)
RTR	Ready to Receive (serial interface signal)
RX	Receiver
SCO link	Synchronous Connection-Oriented Link – Supports time-bounded information like voice. (Master to single slave)
eSCO link	Extended Synchronous Connection-Oriented Link – Supports

Term	Meaning
	time-bounded information like voice. (Version 1.2)
SDP	Service Discovery Protocol
SPP	Serial Port Protocol
TBD	To Be Defined
TCS	Telephony Control protocol Specification
TEI	Terminal Endpoint Identifier
TX	Transmit
UAP	Upper Address Part
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
UUID	Universally Unique Identifier

2. Stack Application Programming Interface

The various parts of the Bluetooth Protocol Stack implementation are documented in separate sections in this chapter. The sections and their contents are:

2.1 BSC (Bluetooth Stack Controller) API

2.2 HCI API

2.3 L2CAP API

2.4 SDP API

2.5 RFCOMM API

2.6 SCO API

Each section will contain a table that will list each Bluetopia Command/Event API along with the availability of the Command/Event in the precompiled Bluetopia Protocol stack.

2.1 BSC (Bluetooth Stack Controller) API

The functions in this section are not defined in the Bluetooth specification, but have been added to provide some stack management and debugging aids. This section defines the supported commands and events in this distribution from the BSC API.

2.1.1 BSC Commands

The supported BSC API commandss are listed in the table below.

Command	Supported
BSC_Initialize	Yes
BSC_Shutdown	Yes
BSC_RegisterDebugCallback	Yes
BSC_UnRegisterDebugCallback	Yes
BSC_RegisterEventCallback	Yes
BSC_UnRegisterEventCallback	Yes
BSC_LockBluetoothStack	Yes
BSC_UnLockBluetoothStack	Yes
BSC_StartTimer	Yes
BSC_StopTimer	Yes
BSC_AuthenticateDevice	Yes
BSC_QueryStackIdle	Yes
BSC_AcquireListLock	Yes

Command	Supported
BSC_ReleaseListLock	Yes
BSC_AddGenericListEntry_Actual	Yes
BSC_AddGenericListEntry	Yes
BSC_SearchGenericListEntry	Yes
BSC_GetNextGenericListEntry	Yes
BSC_DeleteGenericListEntry	Yes
BSC_FreeGenericListEntryMemory	Yes
BSC_FreeGenericListEntryList	Yes

2.1.1 BSC Events

The supported BSC API events are listed in the table below.

Event	Supported
etAuthenticationRequest	Yes

2.2 HCI API

The Host Controller Interface (HCI) layer API of the Bluetooth Protocol Stack provides software access to the HCI command interface to the baseband controller and link manager. This allows access to hardware status and control registers. This API provides a uniform method of accessing the Bluetooth baseband capabilities.

This API is organized into separate subsections primarily by the six command groups as specified in the Bluetooth Core Specification. In addition, there is a section on miscellaneous commands/parameters and a section on the HCI events. Therefore, the subsections that follow are:

2.2.1 Link Control Commands

i. Link Policy Commands

ii. Host Controller & Baseband Commands

iii. Informational Parameters

iv. Status Parameters

v. Testing Commands

vi. Miscellaneous Commands/Parameters

vii. HCI Event Registration and Callbacks

viii. HCI Events

Every API function has a return that is zero when no error occurs in processing the request, and is one of the error conditions listed in the BTErrors.h Header File. In addition, the StatusResult value returned with every HCI command is only valid if the API function return is zero. The header, **HCITypes.h**, includes the defines for all the possible StatusResults returned from the Bluetooth Controller chip.

2.2.1 Link Control Commands

The Link Control commands are used to control the connections to other Bluetooth devices. These commands direct the Link Manager (LM) portion of the HCI to create and modify the link layer connections, and perform inquiries of other devices.

The supported commands in this group are listed below.

Command	Supported
HCI_Inquiry	Yes
HCI_Inquiry_Cancel	Yes
HCI_Periodic_Inquiry_Mode	Yes
HCI_Exit_Periodic_Inquiry_Mode	Yes
HCI_Create_Connection	Yes
HCI_Disconnect	Yes
HCI_Add_SCO_Connection	No
HCI_Accept_Connection_Request	Yes
HCI_Reject_Connection_Request	Yes
HCI_Link_Key_Request_Reply	Yes
HCI_Link_Key_Request_Negative_Reply	Yes
HCI_PIN_Code_Request_Reply	Yes
HCI_PIN_Code_Request_Negative_Reply	Yes
HCI_Change_Connection_Packet_Type	Yes
HCI_Authentication_Requested	Yes
HCI_Set_Connection_Encryption	Yes
HCI_Change_Connection_Link_Key	Yes
HCI_Master_Link_Key	No
HCI_Remote_Name_Request	Yes
HCI_Read_Remote_Supported_Features	Yes
HCI_Read_Remote_Version_Information	Yes

Command	Supported
HCI_Read_Clock_Offset	Yes
HCI_Create_Connection_Cancel	Yes
HCI_Remote_Name_Request_Cancel	Yes
HCI_Read_Remote_Extended_Features	Yes
HCI_Read_LMP_Handle	Yes
HCI_Setup_Synchronous_Connection	No
HCI_Accept_Synchronous_Connection_Request	No
HCI_Reject_Synchronous_Connection_Request	No
HCI_IO_Capability_Request_Reply	Yes
HCI_User_Confirmation_Request_Reply	Yes
HCI_User_Confirmation_Request_Negative_Reply	Yes
HCI_User_Passkey_Request_Reply	Yes
HCI_User_Passkey_Request_Negative_Reply	Yes
HCI_Remote_OOB_Data_Request_Reply	Yes
HCI_Remote_OOB_Data_Request_Negative_Reply	Yes
HCI_IO_Capability_Request_Negative_Reply	Yes
HCI_Create_Physical_Link	No
HCI_Accept_Physical_Link_Request	No
HCI_Disconnect_Physical_Link	No
HCI_Create_Logical_Link	No
HCI_Accept_Logical_Link	No
HCI_Disconnect_Logical_Link	No
HCI_Logical_Link_Cancel	No
HCI_Flow_Spec_Modify	No

2.2.2 Link Policy Commands

The Link Policy Commands provides a means to affect the Link Manager’s (LM) operation.

The supported commands in this group are listed below.

Command	Supported
HCI_Hold_Mode	Yes

Command	Supported
HCI_Sniff_Mode	Yes
HCI_Exit_Sniff_Mode	Yes
HCI_Park_Mode	Yes
HCI_Exit_Park_Mode	Yes
HCI_QoS_Setup	Yes
HCI_Role_Discovery	Yes
HCI_Switch_Role	Yes
HCI_Read_Link_Policy_Settings	Yes
HCI_Write_Link_Policy_Settings	Yes
HCI_Read_Default_Link_Policy_Settings	Yes
HCI_ Write_Default_Link_Policy_Settings	Yes
HCI_ Flow_Specification	Yes
HCI_Sniff_Subrating	Yes

2.2.3 Host Controller & Baseband Commands

These commands provide access and control over parts of the Bluetooth hardware.

The supported commands in this group are listed below.

Command	Supported
HCI_Set_Event_Mask	Yes
HCI_Reset	Yes
HCI_Set_Event_Filter	Yes
HCI_Flush	Yes
HCI_Read_PIN_Type	Yes
HCI_Write_PIN_Type	Yes
HCI_Create_New_Unit_Key	Yes
HCI_Read_Stored_Link_Key	Yes
HCI_Write_Stored_Link_Key	Yes
HCI_Delete_Stored_Link_Key	Yes
HCI_Change_Local_Name	Yes
HCI_Read_Local_Name	Yes

Command	Supported
HCI_Read_Connection_Accept_Timeout	Yes
HCI_Write_Connection_Accept_Timeout	Yes
HCI_Read_Page_Timeout	Yes
HCI_Write_Page_Timeout	Yes
HCI_Read_Scan_Enable	Yes
HCI_Write_Scan_Enable	Yes
HCI_Read_Page_Scan_Activity	Yes
HCI_Write_Page_Scan_Activity	Yes
HCI_Read_Inquiry_Scan_Activity	Yes
HCI_Write_Inquiry_Scan_Activity	Yes
HCI_Read_Authentication_Enable	Yes
HCI_Write_Authentication_Enable	Yes
HCI_Read_Encryption_Mode	Yes
HCI_Write_Encryption_Mode	Yes
HCI_Read_Class_of_Device	Yes
HCI_Write_Class_of_Device	Yes
HCI_Read_Voice_Setting	Yes
HCI_Write_Voice_Setting	Yes
HCI_Read_Automatic_Flush_Timeout	Yes
HCI_Write_Automatic_Flush_Timeout	Yes
HCI_Read_Num_Broadcast_Retransmissions	Yes
HCI_Write_Num_Broadcast_Retransmissions	Yes
HCI_Read_Hold_Mode_Activity	Yes
HCI_Write_Hold_Mode_Activity	Yes
HCI_Read_Transmit_Power_Level	Yes
HCI_Read_SCO_Flow_Control_Enable	Yes
HCI_Write_SCO_Flow_Control_Enable	Yes
HCI_Set_Host_Controller_To_Host_Flow_Control	Yes
HCI_Host_Buffer_Size	Yes
HCI_Host_Number_Of_Completed_Packets	Yes

Command	Supported
HCI_Read_Link_Supervision_Timeout	Yes
HCI_Write Link_Supervision_Timeout	Yes
HCI_Read_Number_Of_Supported_IAC	Yes
HCI_Read_Current_IAC_LAP	Yes
HCI_Write_Current_IAC_LAP	Yes
HCI_Read_Page_Scan_Period_Mode	Yes
HCI_Write_Page_Scan_Period_Mode	Yes
HCI_Read_Page_Scan_Mode	Yes
HCI_Write_Page_Scan_Mode	Yes
HCI_Set_AFH_Host_Channel_Classification	Yes
HCI_ Read_Inquiry_Scan_Type	Yes
HCI_ Write_Inquiry_Scan_Type	Yes
HCI_ Read_Inquiry_Mode	Yes
HCI_ Write_Inquiry_Mode	Yes
HCI_ Read_Page_Scan_Type	Yes
HCI_ Write_Page_Scan_Type	Yes
HCI_Read_AFH_Channel_Assessment_Mode	Yes
HCI_Write_AFH_Channel_Assessment_Mode	Yes
HCI_Read_Extended_Inquiry_Response	Yes
HCI_Write_Extended_Inquiry_Response	Yes
HCI_Refresh_Encryption_Key	Yes
HCI_Read_Simple_Pairing_Mode	Yes
HCI_Write_Simple_Pairing_Mode	Yes
HCI_Read_Local_OOB_Data	Yes
HCI_Read_Inquiry_Response_Transmit_Power_Level	Yes
HCI_Write_Inquiry_Transmit_Power_Level	Yes
HCI_Send_Keypress_Notification	Yes
HCI_Read_Default_Erroneous_Data_Reporting	Yes
HCI_Write_Default_Erroneous_Data_Reporting	Yes
HCI_Enhanced_Flush	Yes

Command	Supported
HCI_Read_Logical_Link_Accept_Timeout	No
HCI_Write_Logical_Link_Accept_Timeout	No
HCI_Set_Event_Mask_Page_2	Yes
HCI_Read_Location_Data	No
HCI_Write_Location_Data	No
HCI_Read_Flow_Control_Mode	No
HCI_Write_Flow_Control_Mode	No
HCI_Read_Enhanced_Transmit_Power_Level	Yes
HCI_Read_Best_Effort_Flush_Timeout	No
HCI_Write_Best_Effort_Flush_Timeout	No
HCI_Short_Range_Mode	No
HCI_Read_LE_Host_Supported	No
HCI_Write_LE_Host_Supported	No

2.2.4 Informational Parameters

The API functions in this section provide access to the Informational Parameters which are settings established by the Bluetooth hardware manufacturer and which provide information about the Bluetooth device and the capabilities of the Host Controller, Link Manager, and Baseband sections.

The supported commands in this group are listed below.

Command	Supported
HCI_Read_Local_Version_Information	Yes
HCI_Read_Local_Supported_Features	Yes
HCI_Read_Buffer_Size	Yes
HCI_Read_Country_Code	Yes
HCI_Read_BD_ADDR	Yes
HCI_Read_Local_Supported_Commands	Yes
HCI_ Read_Local_Extended_Features	Yes
HCI_Read_Data_Block_Size	No

2.2.5 Status Parameters

The Status Parameters retrieved via the commands in this section provide information about the current state of the Host Controller, Link Manager, and Baseband.

The supported commands in this group are listed below.

Command	Supported
HCI_Read_Failed_Contact_Counter	Yes
HCI_Reset_Failed_Contact_Counter	Yes
HCI_Get_Link_Quality	Yes
HCI_Read_RSSI	Yes
HCI_Read_AFH_Channel_Map	Yes
HCI_Read_Clock	Yes
HCI_Read_Encryption_Key_Size	Yes
HCI_Read_Local_AMP_Info	No
HCI_Read_Local_AMP_ASSOC	No
HCI_Write_Remote_AMP_ASSOC	No

2.2.6 Testing Commands

The Testing commands provide the ability to test various functions of the Bluetooth hardware. These commands provide the ability to arrange various conditions for testing.

The supported commands in this group are listed below.

Command	Supported
HCI_Read_Loopback_Mode	Yes
HCI_Write_Loopback_Mode	Yes
HCI_Enable_Device_Under_Test_Mode	Yes
HCI_Write_Simple_Pairing_Debug_Mode	Yes
HCI_Enable_AMP_Receiver_Reports	No
HCI_AMP_Test_End	No
HCI_AMP_Test_Command	No

2.2.7 LE Controller Commands

The These commands provide access and control over LE functions.

The supported commands in this group are listed below.

Command	Supported
HCI_LE_Set_Event_Mask	No
HCI_LE_Read_Buffer_Size	No
HCI_LE_Read_Local_Supported_Features	No
HCI_LE_Set_Random_Address	No
HCI_LE_Set_Advertising_Parameters	No
HCI_LE_Read_Advertising_Channel_Tx_Power	No
HCI_LE_Set_Advertising_Data	No
HCI_LE_Set_Scan_Response	No
HCI_LE_Set_Advertise_Enable	No
HCI_LE_Set_Scan_Parameters	No
HCI_LE_Set_Scan_Enable	No
HCI_LE_Create_Connection	No
HCI_LE_Create_Connection_Cancel	No
HCI_LE_Read_White_List_Size	No
HCI_LE_Clear_White_List	No
HCI_LE_Add_Device_To_White_List	No
HCI_LE_Remove_Device_From_White_List	No
HCI_LE_Connection_Update	No
HCI_LE_Set_Host_Channel_Classification	No
HCI_LE_Read_Channel_Map	No
HCI_LE_Read_Remote_Used_Features	No
HCI_LE_Encrypt	No
HCI_LE_Rand	No
HCI_LE_Start_Encryption	No
HCI_LE_Long_Term_Key_Request_Reply	No
HCI_LE_Long_Term_Key_Requested_Negative_Reply	No
HCI_LE_Read_Supported_States	No
HCI_LE_Reciever_Test	No
HCI_LE_Transmitter_Test	No
HCI_LE_Test_End	No

2.2.8 Miscellaneous Commands/Parameters

These are commands and parameters which are not called out in the Bluetooth specifications, but are needed to facilitate operation of the Bluetooth Protocol Stack.

The supported commands in this group are listed below.

Command	Supported
HCI_Version_Supported	Yes
HCI_Command_Supported	Yes
HCI_Send_ACL_Data	Yes
HCI_Send_SCO_Data	No
HCI_Register_Event_Callback	Yes
HCI_Register_ACL_Data_Callback	Yes
HCI_Register_SCO_Data_Callback	No
HCI_Un_Register_Callback	Yes
HCI_Send_Raw_Command	Yes
HCI_Change_SCO_Configuration	No
HCI_Reconfigure_Driver	Yes

2.2.9 HCI Events

The table below lists the HCI events supported by the current version of the Bluetooth Stack Protocol API. Each event's parameters are further described in text below. The events are an enumeration instance of the enumeration type: HCI_Event_Type_t. The Bluetooth specification includes references to two events not included in this list: Command Complete event and Command Status event. They are omitted from this list because these events are not visible to the application programmer, but are trapped by the Bluetooth Stack and used to set the function return values.

The supported HCI API events are listed in the table below.

Event	Supported
etInquiry_Complete_Event	Yes
etInquiry_Result_Event	Yes
etConnection_Complete_Event	Yes
etConnection_Request_Event	Yes
etDisconnection_Complete_Event	Yes
etAuthentication_Complete_Event	Yes

Event	Supported
etRemote_Name_Request_Complete_Event	Yes
etEncryption_Change_Event	Yes
etChange_Connection_Link_Key_Complete_Event	Yes
etMaster_Link_Key_Complete_Event	No
etRead_Remote_Supported_Features_Complete_Event	Yes
etRead_Remote_Version_Information_Complete_Event	Yes
etQoS_Setup_Complete_Event	No
etHardware_Error_Event	Yes
etFlush_Occurred_Event	Yes
etRole_Change_Event	Yes
etNumber_Of_Completed_Packets_Event	Yes
etMode_Change_Event	Yes
etReturn_Link_Keys_Event	No
etPIN_Code_Request_Event	Yes
etLink_Key_Request_Event	Yes
etLink_Key_Notification_Event	Yes
etLoopback_Command_Event	Yes
etData_Buffer_Overflow_Event	No
etMax_Slots_Change_Event	No
etRead_Clock_Offset_Complete_Event	No
etConnection_Packet_Type_Changed_Event	No
etQoS_Violation_Event	No
etPage_Scan_Mode_Change_Event	No
etPage_Scan_Repetition_Mode_Change_Event	No
etBluetooth_Logo_Testing_Event	Yes
etVendor_Specific_Debug_Event	Yes
etDevice_Reset_Event	
etFlow_Specification_Complete_Event	No
etInquiry_Result_With_RSSI_Event	Yes
etRead_Remote_Extended_Features_Complete_Event	Yes

Event	Supported
etSynchronous_Connection_Complete_Event	No
etSynchronous_Connection_Changed_Event	No
etSniff_Subrating_Event	Yes
etExtended_Inquiry_Result_Event	Yes
etEncryption_Key_Refresh_Complete_Event	Yes
etIO_Capability_Request_Event	Yes
etIO_Capability_Response_Event	Yes
etUser_Confirmation_Request_Event	Yes
etUser_Passkey_Request_Event	Yes
etRemote_OOB_Data_Request_Event	Yes
etSimple_Pairing_Complete_Event	Yes
etLink_Supervision_Timeout_Changed_Event	Yes
etEnhanced_Flush_Complete_Event	Yes
etUser_Passkey_Notification_Event	Yes
etKeypress_Notification_Event	Yes
etRemote_Host_Supported_Features_Notification_Event	Yes
etPhysical_Link_Complete_Event	No
etChannel_Selected_Event	No
etDisconnection_Physical_Link_Complete_Event	No
etPhysical_Link_Loss_Early_Warning_Event	No
etPhysical_Link_Recovery_Event	No
etLogical_Link_Complete_Event	No
etDisconnection_Logical_Link_Complete_Event	No
etFlow_Spec_Modify_Complete_Event	No
etNumber_Of_Completed_Data_Blocks_Event	No
etShort_Range_Mode_Change_Complete_Event	No
etAMP_Status_Change_Event	No
etAMP_Start_Test_Event	No
etAMP_Test_End_Event	No
etAMP_Receiver_Report_Event	No

Event	Supported
etLE_Meta_Event*	No
etPlatform_Specific_Event	No

*LE sub-events listed in section 2.2.10

2.2.10 HCI LE Meta Event Sub-events

The supported HCI API LE Meta Event sub-events are listed in the table below. These are included in the event data for the etLE_Met_Event HCI Event (see 2.2.9).

The supported HCI API LE events are listed in the able below.

Subevent	Supported
meConnection_Complete_Event	No
meAdvertising_Report_Event	No
meConnection_Update_Complete_Event	No
meRead_Remote_Used_Features_Complete_Event	No
meLong_Term_Key_Request_Event	No

2.3 L2CAP API

L2CAP provides connection-oriented and connectionless data services to upper layer protocols with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions. L2CAP permits higher level protocols and applications to transmit and receive L2CAP data packets up to 64 kilobytes in length.

2.3.1 L2CAP Commands

The supported L2CAP API commands are listed below.

Command	Supported
L2CA_Set_Timer_Values	Yes
L2CA_Get_Timer_Values	Yes
L2CA_Connect_Request	Yes
L2CA_Connect_Response	Yes
L2CA_Config_Request	Yes
L2CA_Config_Response	Yes
L2CA_Disconnect_Request	Yes
L2CA_Disconnect_Response	Yes

L2CA_Enhanced_Data_Write	Yes
L2CA_Data_Write	Yes
L2CA_Group_Data_Write	Yes
L2CA_Ping	Yes
L2CA_Get_Info	Yes
L2CA_Group_Create	Yes
L2CA_Group_Close	Yes
L2CA_Group_Add_Member	Yes
L2CA_Group_Remove_Member	Yes
L2CA_Get_Group_Membership	Yes
L2CA_Enable_CLT	Yes
L2CA_Disable_CLT	Yes
L2CA_Flush_Channel_Data	Yes
L2CA_Get_Current_Channel_Configuration	Yes
L2CA_Get_Link_Connection_Configuration	Yes
L2CA_Set_Link_Connection_Configuration	Yes
L2CA_Get_Channel_Queue_Threshold	Yes
L2CA_Set_Channel_Queue_Threshold	Yes
L2CA_Register_PSM	Yes
L2CA_Un_Register_PSM	Yes

2.3.2 L2CAP Events

The supported L2CAP API events are listed in the table below.

Event	Supported
etConnect_Indication	Yes
etConnect_Confirmation	Yes
etConfig_Indication	Yes
etConfig_Confirmation	Yes
etDisconnect_Indication	Yes
etDisconnect_Confirmation	Yes
etEcho_Confirmation	Yes

etInformation_Confirmation	Yes
etTimeout_Indication	Yes
etData_Indication	Yes
etData_Error_Indication	Yes
etGroup_Data_Indication	Yes
etGroup_Add_Member_Confirmation	Yes
etChannel_Buffer_Empty_Indication	Yes

2.4 SDP API

The Service Discovery Protocol (SDP) provides a means for finding services available from or through a Bluetooth device.

2.4.1 SDP Commands

The supported SDP API functions are listed below.

Command	Supported
SDP_Create_Service_Record	Yes
SDP_Delete_Service_Record	Yes
SDP_Add_Attribute	Yes
SDP_Delete_Attribute	Yes
SDP_Service_Search_Request	Yes
SDP_Service_Attribute_Request	Yes
SDP_Service_Search_Attribute_Request	Yes
SDP_Cancel_Service_Request	Yes
SDP_Set_Disconnect_Mode	Yes
SDP_Disconnect_Server	Yes

2.4.2 SDP Response Events

The supported SDP API Response Events are listed in the table below.

Event	Supported
rdTimeout	Yes
rdConnectionError	Yes
rdErrorResponse	Yes

Event	Supported
rdServiceSearchResponse	Yes
rdServiceAttributeResponse	Yes
rdServiceSearchAttributeResponse	Yes

2.5 RFCOMM API

RFCOMM provides serial port emulation over top of the L2CAP protocol, which supports up to 60 simultaneous connections between two Bluetooth devices (or device-specific limits). RFCOMM emulates the nine circuits used in RS-232 serial communications utilizing a subset of the ETSI TS 07.10 standard (see applicable documents). The SPP (Serial Port Profile) is built on top of RFCOMM and for many users provides an easier to use interface.

2.5.1 RFCOMM Commands

The supported RFCOMM command functions are listed in the table below.

Command	Supported
RFCOMM_Set_System_Parameters	Yes
RFCOMM_Get_System_Parameters	Yes
RFCOMM_Set_Data_Queuing_Parameters	Yes
RFCOMM_Get_Data_Queuing_Parameters	Yes
RFCOMM_Register_Server_Channel	Yes
RFCOMM_Un_Register_Server_Channel	Yes
RFCOMM_Open_Request	Yes
RFCOMM_Open_Response	Yes
RFCOMM_Release_Request	Yes
RFCOMM_Send_Credits	Yes
RFCOMM_Send_Data	Yes
RFCOMM_Send_Data_With_Credits	Yes
RFCOMM_Parameter_Negotiation_Response	Yes
RFCOMM_Test_Request	Yes
RFCOMM_Flow_Request	Yes
RFCOMM_Modem_Status	Yes
RFCOMM_Line_Status_Change	Yes
RFCOMM_Remote_Port_Negotiation_Request	Yes

RFCOMM_Remote_Port_Negotiation_Response	Yes
RFCOMM_Query_Remote_Port_Negotiation	Yes
RFCOMM_Get_Channel_Status	Yes

2.5.2 RFCOMM Events

The supported RFCOMM API events are listed in the table below.

Event	Supported
etOpen_Indication	Yes
etOpen_Confirmation	Yes
etRelease_Indication	Yes
etDLCI_Data_Indication	Yes
etDLCI_Param_Negotiation_Indication	Yes
etRemote_Port_Negotiation_Indication	Yes
etRemote_Port_Negotiation_Confirmation	Yes
etRemote_Line_Status_Indication	Yes
etRemote_Line_Status_Confirmation	Yes
etModem_Status_Indication	Yes
etModem_Status_Confirmation	Yes
etTest_Confirmation	Yes
etFlow_Indication	Yes
etFlow_Confirmation	Yes
etCredit_Indication	Yes
etNon_Supported_Command_Indication	Yes
etTransport_Buffer_Empty_Indication	Yes

2.6 SCO API

The Synchronous Connection-Oriented link API provides capabilities for managing SCO Connections.

2.6.1 SCO Commands

The table below shows the supported SCO layer API commands.

Command	Supported
SCO_Setup_Synchronous_Connection	No
SCO_Add_Connection	No
SCO_Close_Connection	No
SCO_Accept_Synchronous_Connection	No
SCO_Accept_Connection	No
SCO_Modify_Synchronous_Connection	No
SCO_Send_Data	No
SCO_Query_Packet_Information	No
SCO_Query_Data_Format	No
SCO_Change_Data_Format	No
SCO_Change_Buffer_Size	No
SCO_Purge_Buffer	No
SCO_Queue_Data	No
SCO_Change_Packet_Information	No
SCO_Set_Connection_Mode	No
SCO_Set_Physical_Transport	No

2.6.2 SCO Events

The supported SCO API events are listed in the table below.

Event	Supported
etSCO_Connect_Result	No
etSCO_Disconnect	No
etSCO_Data_Indication	No
etSCO_Transmit_Buffer_Empty_Indication	No
etSCO_Synchronous_Connection_Changed	No

3. Profile Interfaces

The following Profile Interfaces are included in the Stonestreet One Bluetooth Stack Protocol at present and the sections they are documented in are:

3.1 GAP (Generic Access Protocol) Programming Interface

3.2 SPP (Serial Port Protocol) Programming Interface

3.1 GAP Programming Interface

The GAP (Generic Access Protocol) programming interface provides features related to: (1) discovery of other Bluetooth devices, (2) link management aspects of connecting to those devices, and (3) using different levels of security.

3.1.1 GAP Commands

The supported GAP API commands are listed in the table below.

Command	Supported
GAP_Set_Discoverability_Mode	Yes
GAP_Query_Discoverability_Mode	Yes
GAP_Set_Connectability_Mode	Yes
GAP_Query_Connectability_Mode	Yes
GAP_Set_Pairability_Mode	Yes
GAP_Query_Pairability_Mode	Yes
GAP_Set_Authentication_Mode	Yes
GAP_Query_Authentication_Mode	Yes
GAP_Set_Encryption_Mode	Yes
GAP_Cancel_Set_Encryption _Mode	Yes
GAP_Query_Encryption_Mode	Yes
GAP_Authenticate_Remote_Device	Yes
GAP_Cancel_Authenticate_Remote_Device	Yes
GAP_Register_Remote_Authentication	Yes
GAP_Un_Register_Remote_Authentication	Yes
GAP_Authentication_Response	Yes
GAP_Perform_Inquiry	Yes
GAP_Cancel_Inquiry	Yes

GAP_Set_Inquiry_Mode	Yes
GAP_Query_Inquiry_Mode	Yes
GAP_Query_Remote_Device_Name	Yes
GAP_Cancel_Query_Remote_Device_Name	Yes
GAP_Query_Remote_Features	Yes
GAP_Query_Remote_Version_Information	Yes
GAP_Initiate_Bonding	Yes
GAP_Cancel_Bonding	Yes
GAP_End_Bonding	Yes
GAP_Query_Local_BD_ADDR	Yes
GAP_Set_Class_Of_Device	Yes
GAP_Query_Class_Of_Device	Yes
GAP_Set_Local_Device_Name	Yes
GAP_Query_Local_Device_Name	Yes
GAP_Disconnect_Link	Yes
GAP_Query_Connection_Handle	Yes
GAP_Query_Local_Out_Of_Band_Data	Yes
GAP_Refresh_Encryption_Key	Yes
GAP_Read_Extended_Inquiry_Information	Yes
GAP_Write_Extended_Inquiry_Information	Yes
GAP_Convert_Extended_Inquiry_Response_Data	Yes
GAP_Parse_Extended_Inquiry_Response_Data	Yes

3.1.2 GAP Events

The supported GAP API events are listed in the table below.

Event	Supported
etInquiry_Result	Yes
etEncryption_Change_Result	Yes
etAuthentication*	Yes
etRemote_Name_Result	Yes
etInquiry_Entry_Result	Yes

etInquiry_With_RSSI_Entry_Result	Yes
etExtended_Inquiry_Entry_Result	Yes
etEncryption_Refresh_Complete	Yes
etRemote_Features_Result	Yes
etRemote_Version_Information_Result	Yes

*Sub-events listed in section 3.1.3

3.1.3 GAP Authentication Sub-Events

The supported GAP API Authentication sub-events are listed in the table below. These are included in the event data for the etAuthentication GAP Event (see 3.1.2).

The supported GAP API sub-events are listed in the table below.

Event	Supported
atLinkKeyRequest	Yes
atPINCodeRequest	Yes
atAuthenticationStatus	Yes
atLinkKeyCreation	Yes
atIOCapabilityRequest	Yes
atUserConfirmationRequest	Yes
atPasskeyRequest	Yes
atPasskeyNotification	Yes
atKeypressNotification	Yes
atRemoteOutOfBandDataRequest	Yes
atIOCapabilityResponse	Yes

3.2 SPP Programming Interface

The SPP (Serial Port Profile) programming interface provides an emulation of serial cable connections (e.g., RS-232) between two Bluetooth devices via the RFCOMM protocol.

3.2.1 SPP Commands

The supported SPP API commands are listed in the table below.

Command	Supported
SPP_Open_Server_Port	Yes

SPP_Close_Server_Port	Yes
SPP_Open_Port_Request_Response	Yes
SPP_Register_SDP_Record	Yes
SPP_Open_Remote_Port	Yes
SPP_Close_Port	Yes
SPP_Data_Read	Yes
SPP_Data_Write	Yes
SPP_Change_Buffer_Size	Yes
SPP_Purge_Buffer	Yes
SPP_Send_Break	Yes
SPP_Line_Status	Yes
SPP_Port_Status	Yes
SPP_Send_Port_Information	Yes
SPP_Respond_Port_Information	Yes
SPP_Query_Remote_Port_Information	Yes
SPP_Respond_Query_Port_Information	Yes
SPP_Get_Configuration_Parameters	Yes
SPP_Set_Configuration_Parameters	Yes
SPP_Get_Server_Connection_Mode	Yes
SPP_Set_Server_Connection_Mode	Yes
SPP_Get_Port_Connection_State	Yes
SPP_Set_Queuing_Parameters	Yes
SPP_Get_Queuing_Parameters	Yes

3.2.2 SPP Events

The supported SPP API events are listed in the table below.

Event	Supported
etPort_Open_Indication	Yes
etPort_Open_Confirmation	Yes
etPort_Close_Port_Indication	Yes
etPort_Status_Indication	Yes

etPort_Data_Indication	Yes
etPort_Transmit_Buffer_Empty_Indication	Yes
etPort_Line_Status_Indication	Yes
etPort_Send_Port_Information_Indication	Yes
etPort_Send_Port_Information_Confirmation	Yes
etPort_Query_Port_Information_Indication	Yes
etPort_Query_Port_Information_Confirmation	Yes
etPort_Open_Request_Indication	Yes

3.3 GOEP Programming Interface

The GOEP (Generic Object Exchange Profile) programming interface defines the protocols and procedures to be used to implement Object Exchange (OBEX) capabilities such as folder synchronization, file transfer, and Object Push activities.

3.3.1 GOEP Commands

The supported GOEP API commands are listed in the table below.

Command	Supported
GOEP_Open_Server_Port	No
GOEP_Close_Server_Port	No
GOEP_Open_Port_Request_Response	No
GOEP_Register_SDP_Record	No
GOEP_Open_Remote_Port	No
GOEP_Close_Port	No
GOEP_Connect_Request	No
GOEP_Disconnect_Request	No
GOEP_Put_Request	No
GOEP_Get_Request	No
GOEP_Set_Path_Request	No
GOEP_Abort_Request	No
GOEP_Command_Response	No
GOEP_Get_Server_Connection_Mode	No
GOEP_Set_Server_Connection_Mode	No

GOEP_Find_Application_Parameter_Header_By_Tag_ID	No
GOEP_Find_Header	No
GOEP_Generate_Digest_Nonce	No

3.3.2 GOEP Events

The supported GOEP API events are listed in the table below.

Event	Supported
etOBEX_Port_Open_Indication	No
etOBEX_Port_Open_Confirmation	No
etOBEX_Port_Open_Request_Indication	No
etOBEX_Port_Close_Indication	No
etOBEX_Connect_Indication	No
etOBEX_Connect_Confirmation	No
etOBEX_Disconnect_Indication	No
etOBEX_Disconnect_Confirmation	No
etOBEX_Put_Indication	No
etOBEX_Put_Confirmation	No
etOBEX_Get_Indication	No
etOBEX_Get_Confirmation	No
etOBEX_Set_Path_Indication	No
etOBEX_Set_Path_Confirmation	No
etOBEX_Abort_Indication	No
etOBEX_Abort_Confirmation	No

3.4 OTP Programming Interface

The OTP (Object Transfer Protocol) programming interface defines the protocols and procedures to be used to perform File Transfer Protocol (FTP) and Object Transfer Protocol functions called out in the Bluetooth Profile specification.

3.4.1 OTP Commands

The supported OTP API commands are listed in the table below.

Command	Supported
OTP_Open_Server_Port	No
OTP_Close_Server_Port	No
OTP_Open_Port_Request_Response	No
OTP_Register_SDP_Record	No
OTP_Open_Remote_Port	No
OTP_Close_Port	No
OTP_Client_Connect	No
OTP_Client_Disconnect	No
OTP_Client_Get_Directory	No
OTP_Client_Get_Object	No
OTP_Client_Put_Object_Request	No
OTP_Client_Put_Sync_Object_Request	No
OTP_Client_Put_Object	No
OTP_Client_Set_Path	No
OTP_Client_Delete_Object_Request	No
OTP_Client_Delete_Sync_Object_Request	No
OTP_Client_Abort_Request	No
OTP_Connect_Response	No
OTP_Get_Directory_Request_Response	No
OTP_Set_Path_Response	No
OTP_Abort_Response	No
OTP_Get_Object_Response	No
OTP_Delete_Object_Response	No
OTP_Delete_Sync_Object_Response	No
OTP_Put_Object_Response	No
OTP_Put_Sync_Object_Response	No
OTP_Get_Server_Connection_Mode	No
OTP_Set_Server_Connection_Mode	No

3.4.2 OTP Events

The supported OTP API events are listed in the table below.

Event	Supported
etOTP_Port_Open_Indication	No
etOTP_Port_Open_Confirmation	No
etOTP_Port_Open_Request_Indication	No
etOTP_Port_Close_Port_Indication	No
etOTP_Connect_Request	No
etOTP_Connect_Response	No
etOTP_Disconnect_Request	No
etOTP_Disconnect_Response	No
etOTP_Set_Path_Request	No
etOTP_Set_Path_Response	No
etOTP_Abort_Request	No
etOTP_Abort_Response	No
etOTP_Delete_Object_Request	No
etOTP_Delete_Sync_Object_Request	No
etOTP_Delete_Object_Response	No
etOTP_Delete_Sync_Object_Response	No
etOTP_Put_Object_Request	No
etOTP_Put_Sync_Object_Request	No
etOTP_Put_Object_Response	No
etOTP_Put_Sync_Object_Response	No
etOTP_Get_Object_Request	No
etOTP_Get_Object_Response	No
etOTP_Get_Directory_Request	No
etOTP_Get_Directory_Response	No
etOTP_Free_Directory_Information	No