Bluetopia®

Generic Audio/Video Distribution Profile (GAVD)

Application Programming Interface Reference Manual

Profile Version: 1.0

Release: 2.1.3
January 19, 2011

stonestreet one

Louisville, KY   www.stonestreetone.com

Bluetooth®

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 (Synchronous Connection-Oriented) Link layers. In addition to basic functionality at these layers, the Bluetooth Protocol Stack 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.

This document focuses on the API reference that contains a description of all programming interfaces for the Bluetooth Generic Audio/Video Distribution Profile provided by Bluetopia. Chapter 2 contains a description of the programming interfaces for this profile. Chapter 3 contains the header file name list for the Bluetooth GAVD Profile library.

1.1   Scope

This reference manual provides information on the GAVD Profile 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
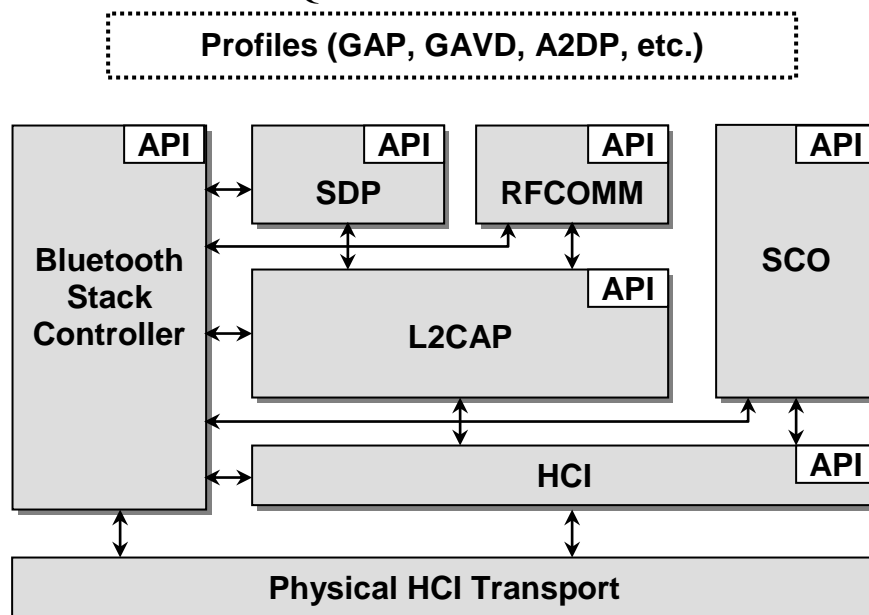
Figure 1-1   The Stonestreet One Bluetooth Protocol Stack

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, 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 2.0 + EDR, November 4, 2004.

4. *Specification of the Bluetooth System, Volume 2, Core System Package*, version 2.0 + EDR, November 4, 2004.

5. *Specification of the Bluetooth System, Volume 3, Core System Package*, version 2.0 + EDR, November 4, 2004.

6. *Bluetooth Assigned Numbers,* version 1.1, February 22, 2001.

7. *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.

8. *Audio/Video Distribution Transport Protocol Specification*, version 1.0, May 22, 2003.

9. *Generic Audio/Video Distribution Profile*, version 1.0, May 22, 2002.

10. *Bluetopia®Protocol Stack, Application Programming Interface Reference Manual,* version 2.1.3, August 31, 2010.

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
AVDTP	Audio/Video Distribution Transport Protocol
BD_ADDR	Bluetooth Device Address
BT	Bluetooth
GAVD	Generic Audio/Video Distribution
LSB	Least Significant Bit

Term	Meaning
LSEID	Local Stream Endpoint Identifier
MSB	Most Significant Bit
RSEID	Remote Stream Endpoint Identifier
SDP	Service Discovery Protocol
SPP	Serial Port Protocol
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

2. GAVD Profile Programming Interface

The GAVD Profile programming interface defines the protocols and procedures to be used to implement audio/video distribution/transport capabilities. The GAVD Profile commands are listed in section 2.1, the event callback prototype is described in section 2.2, and the GAVD Profile events are itemized in section 2.3. The actual prototypes and constants outlined in this section can be found in the **GAVDAPI.H** header file in the Bluetopia distribution.

2.1   GAVD Profile Commands

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

Function	Description
GAVD_Initialize	Initializes the GAVD Profile and starts a local Stream Endpoint Manager.
GAVD_Cleanup	Removes the GAVD Profile from the system and shuts down the local Stream Endpoint Manager.
GAVD_Connect_Request_Response	Responds to individual request to connect to a local GAVD/AVDTP Server.
GAVD_Register_SDP_Record	Adds a generic GAVD Service Record to the SDP database.
GAVD_Un_Register_SDP_Record	Removes a generic GAVD Service Record from the SDP database.
GAVD_Connect	Connects to a Stream Endpoint Manager on the specified remote device.
GAVD_Disconnect	Closes a connection to a Stream Endpoint Manager that was previously opened with the GAVD_Connect() function.
GAVD_Discover_End_Points	Sends a Discover request to a remote Stream Endpoint Manager.
GAVD_Get_End_Point_Capabilities	Sends a Get Capabilities request for a specified remote stream endpoint.
GAVD_Get_Configuration	Sends a Get Configuration request for a specified remote stream endpoint.
GAVD_Register_End_Point	Registers a local stream endpoint with a local Stream Endpoint Manager.
GAVD_Un_Register_End_Point	Unregisters a previously registered stream endpoint from a local Stream Endpoint Manager.

GAVD_Open_Remote_End_Point	Establishes a connection to a remote endpoint on a Remote Endpoint Stream Manager.
GAVD_Close_End_Point	Closes a connection to an endpoint that was previously opened via a call to GAVD_Register_End_Point() or GAVD_Open_Remote_End_Point().
GAVD_Set_Configuration_Response	Responds to a request from a remote GAVD device to set the configuration of a stream endpoint.
GAVD_Start_Stream_Request	Requests to start one or more streams on a remote GAVD device.
GAVD_Start_Stream_Response	Responds to a request from a remote GAVD device to start one or more streams.
GAVD_Suspend_Stream_Request	Suspends one or more streams on a remote GAVD device.
GAVD_Suspend_Stream_Response	Responds to a request from a remote GAVD device to suspend one or more streams.
GAVD_Reconfigure_Request	Requests the reconfiguration of an endpoint on a remote GAVD device.
GAVD_Reconfigure_Response	Responds to a request from a remote GAVD device to reconfigure a stream.
GAVD_Security_Control_Request	Requests the change of the security settings for a stream.
GAVD_Security_Control_Response	Responds to a request from a remote GAVD device to change the security settings for a stream.
GAVD_Abort_Stream_Request	Requests to abort one or more streams on a remote GAVD device.
GAVD_Data_Write	Sends Media data over a specified stream.
GAVD_Sender_Report_Data_Write	Sends Sender Report data over the specified stream.
GAVD_Receiver_Report_Data_Write	Sends Receiver Report data over the specified stream.
GAVD_SDES_Report_Data_Write	Sends SDES Report data over the specified stream.
GAVD_Recovery_Data_Write	Sends RTP FEC recovery data over a specified stream.
GAVD_Get_Server_Connection_Mode	Retrieves the current GAVD/AVDTP Server Connection Mode.
GAVD_Set_Server_Connection_Mode	Changes the current GAVD/AVDTP Server Connection Mode.

GAVD_Get_Data_Queueing_Parameters	Retrieves current GAVD/L2CAP data queueing mode
GAVD_Set_Data_Queueing_Parameters	Changes the current GAVD/L2CAP data queueing mode

GAVD_Initialize

This function initializes the GAVD Profile. This function must be called before any other GAVD Profile function can be called. This function can only be called once per Bluetooth Stack Instance.

Prototype:

int BTPSAPI **GAVD_Initialize**(unsigned int BluetoothStackID)

Parameters:

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

Return:

Zero if successful.

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

>   BTGAVD_ERROR_INSUFFICIENT_RESOURCES
>   BTGAVD_ERROR_NOT_INITIALIZED
>   BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
>   BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

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.

GAVD_Cleanup

This function is responsible for removing a GAVD Profile from the system and shutting down the local Stream Endpoint Manager.

Note:

This function does NOT delete any SDP Service Record Handles (i.e., added via a call to the GAVD_Register_SDP_Record() function).

Prototype:

void BTPSAPI **GAVD_Cleanup**(unsigned int BluetoothStackID)

Parameters:

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

Return:

None.

Possible Events:

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.

GAVD_Connect_Request_Response

This function responds to an individual request to connect to a local GAVD/AVDTP Server.

Prototype:

int BTPSAPI **GAVD_Connect_Request_Response**(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, Boolean_t AcceptConnection)

Parameters:

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

BD_ADDR   Address of the Bluetooth device that is attempting to connect to a local GAVD/AVDTP Server.

AcceptConnection   Boolean specifying if the pending connection request is to be accepted (TRUE).

Return:

Zero if successful.

An error code if negative; one of the following values:
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_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.

GAVD_Register_ SDP_Record

This function adds a generic GAVD Service Record to the SDP database.

Notes:

1. The Service Record Handle that is returned from this function will remain in the SDP Record Database until it is deleted by calling the SDP_Delete_Service_Record() function. A macro is provided to delete the Service Record from the SDP Database. This macro maps the GAVD_Un_Register_SDP_Record() to SDP_Delete_Service_Record(), and is defined as follows:

   GAVD_Un_Register_SDP_Record(__BluetoothStackID, __SDPRecordHandle)
   (SDP_Delete_Service_Record(__BluetoothStackID, __SDPRecordHandle))

2. Any Protocol Information that is specified will be added in the protocol attribute after the default protocol list of L2CAP and AVDTP.

3. The Service Name is always added at Attribute ID 0x0100. A Language Base Attribute ID List is created that specifies that 0x0100 is UTF-8 Encoded, English Language.

4. At least one Service Class (UUID) must be specified in the SDP Service Record structure.

5. The ProtocolList and ProfileList members of the SDP Service Record structure are optional (specified as NULL). The Protocol List information must be a Data Element Sequence, and the information contained in this sequence is added AFTER the AVDTP and L2CAP Protocol information. The ProfileList must also be a Data Element Sequence, however, this information is added as-is (nothing is added other than this information).

Prototype:

int BTPSAPI **GAVD_Register_SDP_Record**(unsigned int BluetoothStackID, GAVD_SDP_Service_Record_t *SDPServiceRecord, char *ServiceName, DWord_t *SDPServiceRecordHandle)

Parameters:

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

SDPServiceRecord   Specifies additional SDP information to add to the record. This is defined by the following structure:
```
typedef struct
{
  unsigned int            NumberServiceClassUUID;
  SDP_UUID_Entry_t   *SDPUUIDEntries;
  SDP_Data_Element_t *ProtocolList;
  SDP_Data_Element_t *ProfileList;
} GAVD_SDP_Service_Record_t;
```

ServiceName   Name to appear in the SDP Database for this service.

SDPServiceRecordHandle   Returned handle to the SDP Database entry that may be used to remove the entry at a later time.

Return:

None.

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.

GAVD_Connect

This function is used to connect to a remote Stream Endpoint Manager on the specified remote device.

Prototype:

int BTPSAPI **GAVD_Connect**(unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAVD_Event_Callback_t GAVDEventCallback, unsigned long CallbackParameter)

Parameters:

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

BD_ADDR   Address of the Bluetooth device to connect with.

GAVDEventCallback   Function that is called whenever events occur on this connection.

CallbackParameter   A user-defined parameter (e.g., a tag value) that will be passed back to the user in the callback function with each event callback.

Return:

Positive, non-zero if successful. If this function is successful, the return value will represent the GAVD Client ID that can be passed to all other functions that require it.

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

> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Connect_Confirmation

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.

GAVD_Disconnect

This function is used to close a connection to a remote Stream Endpoint Manager that was previously opened via the GAVD_Connect() function. This function does NOT close the remote Stream Endpoint Manager, it ONLY closes this client's connection to the remote Stream Endpoint Manager IF and ONLY IF no stream endpoints were created. If stream endpoints were created, calling this routine will not close connection. All stream endpoints must be closed before calling this routine.

Prototype:

int BTPSAPI **GAVD_Disconnect**(unsigned int BluetoothStackID, unsigned int GAVDID)

Parameters:

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

GAVDID   The GAVD Client ID. This is the value that was returned from the GAVD_Connect() function.

Return:

Zero if successful.

An error code if negative; one of the following values:
> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

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.

GAVD_Discover_End_Points

This function is used to send a Discover request to a remote Stream Endpoint Manager that was previously opened via the GAVD_Connect() function.

Prototype:

int BTPSAPI **GAVD_Discover_End_Points**(unsigned int BluetoothStackID, unsigned int GAVDID)

Parameters:

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

GAVDID   The GAVD client ID. This is the value that was returned from
                             the GAVD_Connect() function.

Return:

Zero if successful.

An error code if negative; one of the following values:
                             BTGAVD_ERROR_INVALID_OPERATION
                             BTGAVD_ERROR_NOT_INITIALIZED
                             BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
                             BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Discover_Confirmation
etGAVD_Disconnect_Indication

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.

GAVD_Get_End_Point_Capabilities

This function is used to send a Get Capabilities request to a remote Stream Endpoint
Manager to determine the supported capabilities of the specified remote stream endpoint.

Prototype:

int BTPSAPI **GAVD_Get_End_Point_Capabilities**(unsigned int BluetoothStackID,
    unsigned int GAVDID, unsigned int RSEID)

Parameters:

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

GAVDID   The GAVD client ID. This is the value that was returned from
                             the GAVD_Connect() function.

RSEID   The remote endpoint ID. This value is one of the values
                             returned in a Discover Confirmation event.

Return:

Zero if successful.

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

> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Get_Capabilities_Confirmation
etGAVD_Disconnect_Indication

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.

GAVD_Get_Configuration

This function is used to send a Get Configuration request to a remote Stream Endpoint Manager to determine the current configuration of the specified remote stream endpoint.

Prototype:

int BTPSAPI **GAVD_Get_Configuration**(unsigned int BluetoothStackID,
    unsigned int GAVDID, unsigned int RSEID)

Parameters:

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

GAVDID   The GAVD client ID. This is the value that was returned from the GAVD_Connect() function.

RSEID   The remote endpoint ID. This value is one of the values that was returned in a Discover Confirmation event.

Return:

Zero if successful.

An error code if negative; one of the following values:
> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Get_Configuration_Confirmation
etGAVD_Disconnect_Indication

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.

GAVD_Register_End_Point

This function is used to register a local stream endpoint with the local Stream Endpoint Manager. If this function is successful then an endpoint will be created that is ready to be discovered, configured, and connected to by remote AVDTP/GAVD clients.

Prototype:

int BTPSAPI **GAVD_Register_End_Point**(unsigned int BluetoothStackID,
    GAVD_Local_End_Point_Info_t *LocalEndPointInfo,
    GAVD_Event_Callback_t GAVDEventCallback, unsigned long CallbackParameter)

Parameters:

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

LocalEndPointInfo   Data containing the configuration information for the local endpoint. This is defined by the following structure:

typedef struct
{
  GAVD_TSEP_t   TSEP;
  GAVD_Media_Type_t   MediaType;
  Word_t   MediaInMTU;
  Word_t   ReportingInMTU;
  Word_t   RecoveryInMTU;
  unsigned int   NumberCapabilities;
  GAVD_Service_Capabilities_Info_t *CapabilitiesInfo;
} GAVD_Local_End_Point_Info_t;

where,

    TSEP defines whether the endpoint is a source or a sink using the following enumeration:
        typedef enum
        {
          tspSRC,
          tspSNK
        } GAVD_TSEP_t;

    MediaType identifies the media type as audio, video, or multimedia using the following enumeration:
        typedef enum
        {
          mtAudio,
          mtVideo,
          mtMultimedia
        } GAVD_Media_Type_t;

    MediaInMTU, ReportingInMTU, and RecoveryInMTU are the maximum allowable MTU sizes.

NumberCapabilities specifies how many service capability information elements are contained in the CapabilitiesInfo member.

CapabilitiesInfo defines the service capabilities that an endpoint supports using the following structure:

```
typedef struct
{
  GAVD_Service_Category_t ServiceCategory;
  union
  {
    GAVD_Recovery_Info_Element_Data_t
         GAVD_Recovery_Info_Element_Data;
    GAVD_Content_Protection_Info_Element_Data_t
         GAVD_Content_Protection_Info_Element_Data;
    GAVD_Header_Compression_Info_Element_Data_t
         GAVD_Header_Compression_Info_Element_Data;
    GAVD_Multiplexing_Info_Element_Data_t
         GAVD_Multiplexing_Info_Element_Data;
    GAVD_Media_Codec_Info_Element_Data_t
         GAVD_Media_Codec_Info_Element_Data;
    GAVD_Raw_Info_Element_Data_t
         GAVD_Raw_Info_Element_Data;
  } InfoElement;
} GAVD_Service_Capabilities_Info_t;
```

where,

ServiceCategory is defined by the following enumeration:

```
typedef enum
{
  scNone,
  scMediaTransport,
  scReporting,
  scRecovery,
  scContentProtection,
  scHeaderCompression,
  scMultiplexing,
  scMediaCodec,
  scUnknown
} GAVD_Service_Category_t;
```

GAVD_Recovery_Info_Element_Data is defined by the following structure:

```
typedef struct
{
  Byte_t    RecoveryType;
  Byte_t    MaxRecoveryWindowSize;
  Byte_t    MaxNumberMediaPackets;
} GAVD_Recovery_Info_Element_Data_t;
```

GAVD_Content_Protection_Info_Element_Data is defined by the following structure:

```
typedef struct
{
  Word_t    ContentProtectionType;
  Byte_t    ContentProtectionTypeSpecificInfoLength;
  Byte_t    *ContentProtectionTypeSpecificInfo;
} GAVD_Content_Protection_Info_Element_Data_t;
```

GAVD_Header_Compression_Info_Element_Data is defined by the following structure:

```
typedef struct
Data_t
{
  Boolean_t MediaPacketHeaderCompression;
  Boolean_t RecoveryPacketHeaderCompression;
  Boolean_t BackChannelSupported;
} GAVD_Header_Compression_Info_Element_Data_t;
```

GAVD_Multiplexing_Info_Element_Data is defined by the following structure:

```
typedef struct
{
  unsigned int              MediaMuxLSEID;
  GAVD_Transport_Channel_Type_t
                            MediaMuxChannel;
  Boolean_t                 UseReportingChannel;
  unsigned int              ReportingMuxLSEID;
  GAVD_Transport_Channel_Type_t
                            ReportingMuxChannel;
  Boolean_t                 UseRecoveryChannel;
  unsigned int              RecoveryMuxLSEID;
  GAVD_Transport_Channel_Type_t
                            RecoveryMuxChannel;
} GAVD_Multiplexing_Info_Element_Data_t;
```

where,
  MediaMuxChannel, ReportingMuxChannel, and RecoveryMuxChannel are defined by the following enumeration:

```
typedef enum
{
  trMedia,
  trReporting,
  trRecovery,
  trNone
} GAVD_Transport_Channel_Type_t;
```

GAVD_Media_Codec_Info_Element_Data is defined by the following structure:

```
typedef struct
{
  GAVD_Media_Type_t      MediaType;
  Byte_t                        MediaCodecType;
  Byte_t
            MediaCodecSpecificInfoLength;
  Byte_t                   *MediaCodecSpecificInfo;
} GAVD_Media_Codec_Info_Element_Data_t;
```

where,

MediaType is defined by the following enumeration:

```
typedef enum
{
  mtAudio,
  mtVideo,
  mtMultimedia
} GAVD_Media_Type_t;
```

GAVD_Raw_Info_Element_Data is defined by the following structure:

```
typedef struct
{
  Byte_t    RawDataLength;
  Byte_t    *RawData;
} GAVD_Raw_Info_Element_Data_t;
```

GAVDEventCallback	Function that is be called whenever events occur regarding the specified local endpoint.
CallbackParameter	A user-defined parameter (e.g., a tag value) that will be passed back to the user in the callback function with each event callback.

Return:

Positive, non-zero value if successful that is the LSEID that must be used to identify this Stream End Point in future calls.

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

BTGAVD_ERROR_INVALID_OPERATION
BTGAVD_ERROR_NOT_INITIALIZED
BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

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.

GAVD_Un_Register_End_Point

This function is used to unregister previously registered local stream endpoint from a local Stream Endpoint Manager. This endpoint was registered via the GAVD_Register_End_Point() function.

Prototype:

int BTPSAPI **GAVD_Un_Register_End_Point**(unsigned int BluetoothStackID, unsigned int LSEID)

Parameters:

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

LSEID   Endpoint ID of the local stream endpoint to unregister. This value was returned from a successful call to the GAVD_Register_End_Point() function.

Return:

Zero if successful.

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

                  BTGAVD_ERROR_INVALID_OPERATION
                  BTGAVD_ERROR_NOT_INITIALIZED
                  BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
                  BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

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.

GAVD_Open_Remote_End_Point

This function is used to establish a connection to a remote endpoint on a remote Endpoint Stream Manager. This function will establish a connection to the remote Endpoint Stream Manager if a connection is not already present and establish all required channels (Media, Reporting, Recovery) as needed. The local endpoint that is registered will not be able to be discovered or connected to. This endpoint will only exist as long as there is a connection to the specified remote endpoint.

Prototype:

> int BTPSAPI **GAVD_Open_Remote_End_Point**(unsigned int BluetoothStackID,
>   BD_ADDR_t BD_ADDR, unsigned int RSEID,
>   GAVD_Local_End_Point_Info_t *LocalEndPointInfo,
>   unsigned int NumberConfigurationCapabilities, GAVD_Service_Capabilities_Info_t
>   ConfigurationCapabilities[], GAVD_Event_Callback_t GAVDEventCallback,
>   unsigned long CallbackParameter)

Parameters:

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

BD_ADDR   Bluetooth address of the remote device.

RSEID   Identifier of the remote stream endpoint to open.

LocalEndPointInfo   Contains the necessary information for the local end point that
                           will connect to the remote end point that is being opened. This
                           is defined by the following structure:

```
typedef struct
{
  GAVD_TSEP_t                      TSEP;
  GAVD_Media_Type_t                MediaType;
  Word_t                           MediaInMTU;
  Word_t                           ReportingInMTU;
  Word_t                           RecoveryInMTU;
  unsigned int                     NumberCapabilities;
  GAVD_Service_Capabilities_Info_t *CapabilitiesInfo;
} GAVD_Local_End_Point_Info_t;
```

where,

> TSEP defines whether the endpoint is a source or a sink
> using the following enumeration:
> ```
>     typedef enum
>     {
>       tspSRC,
>       tspSNK
>     } GAVD_TSEP_t;
> ```

> MediaType identifies the media type as audio, video, or
> multimedia using the following enumeration:
> ```
>     typedef enum
>     {
>       mtAudio,
>       mtVideo,
>       mtMultimedia
>     } GAVD_Media_Type_t;
> ```

MediaInMTU, ReportingInMTU, and RecoveryInMTU are the maximum allowable MTU sizes.

NumberCapabilities specifies how many service capability information elements are contained in the CapabilitiesInfo member.

CapabilitiesInfo defines the service capabilities that an endpoint supports using the following structure:

```
typedef struct
{
  GAVD_Service_Category_t ServiceCategory;
  union
  {
    GAVD_Recovery_Info_Element_Data_t
         GAVD_Recovery_Info_Element_Data;
    GAVD_Content_Protection_Info_Element_Data_t
         GAVD_Content_Protection_Info_Element_Data;
    GAVD_Header_Compression_Info_Element_Data_t
         GAVD_Header_Compression_Info_Element_Data;
    GAVD_Multiplexing_Info_Element_Data_t
         GAVD_Multiplexing_Info_Element_Data;
    GAVD_Media_Codec_Info_Element_Data_t
         GAVD_Media_Codec_Info_Element_Data;
    GAVD_Raw_Info_Element_Data_t
         GAVD_Raw_Info_Element_Data;
  } InfoElement;
} GAVD_Service_Capabilities_Info_t;
```

where,

ServiceCategory is defined by the following enumeration:

```
typedef enum
{
  scNone,
  scMediaTransport,
  scReporting,
  scRecovery,
  scContentProtection,
  scHeaderCompression,
  scMultiplexing,
  scMediaCodec,
  scUnknown
} GAVD_Service_Category_t;
```

GAVD_Recovery_Info_Element_Data is defined by the following structure:

```
typedef struct
{
  Byte_t    RecoveryType;
```

```
                              Byte_t    MaxRecoveryWindowSize;
                              Byte_t    MaxNumberMediaPackets;
                           } GAVD_Recovery_Info_Element_Data_t;
```

GAVD_Content_Protection_Info_Element_Data is defined by the following structure:

```
    typedef struct
    {
       Word_t    ContentProtectionType;
       Byte_t    ContentProtectionTypeSpecificInfoLength;
       Byte_t    *ContentProtectionTypeSpecificInfo;
    } GAVD_Content_Protection_Info_Element_Data_t;
```

GAVD_Header_Compression_Info_Element_Data is defined by the following structure:

```
    typedef struct
    {
       Boolean_t MediaPacketHeaderCompression;
       Boolean_t RecoveryPacketHeaderCompression;
       Booleant_t BackChannelSupported;
    } GAVD_Header_Compression_Info_Element_Data_t;
```

GAVD_Multiplexing_Info_Element_Data is defined by the following structure:

```
    typedef struct
    {
       unsigned int              MediaMuxLSEID;
       GAVD_Transport_Channel_Type_t
                                 MediaMuxChannel;
       Boolean_t                 UseReportingChannel;
       unsigned int              ReportingMuxLSEID;
       GAVD_Transport_Channel_Type_t
                                 ReportingMuxChannel;
       Boolean_t                 UseRecoveryChannel;
       unsigned int              RecoveryMuxLSEID;
       GAVD_Transport_Channel_Type_t
                                 RecoveryMuxChannel;
    } GAVD_Multiplexing_Info_Element_Data_t;
```

where,

```
       MediaMuxChannel, ReportingMuxChannel, and
       RecoveryMuxChannel are defined by the
       following enumeration:
           typedef enum
           {
             trMedia,
             trReporting,
             trRecovery,
             trNone
           } GAVD_Transport_Channel_Type_t;
```

GAVD_Media_Codec_Info_Element_Data is defined by the following structure:

```
typedef struct
{
  GAVD_Media_Type_t      MediaType;
  Byte_t                 MediaCodecType;
  Byte_t
              MediaCodecSpecificInfoLength;
  Byte_t                 *MediaCodecSpecificInfo;
} GAVD_Media_Codec_Info_Element_Data_t;
```

where,

MediaType is defined by the following enumeration:

```
typedef enum
{
  mtAudio,
  mtVideo,
  mtMultimedia
} GAVD_Media_Type_t;
```

GAVD_Raw_Info_Element_Data is defined by the following structure:

```
typedef struct
{
  Byte_t   RawDataLength;
  Byte_t   *RawData;
} GAVD_Raw_Info_Element_Data_t;
```

NumberConfigurationCapabilities	The number of configuration options.
ConfigurationCapabilities	The configurations options that will be used to configure the remote endpoint.
GAVDEventCallback	Function that is be called whenever events occur regarding the specified local endpoint.
CallbackParameter	A user-defined parameter (e.g., a tag value) that will be passed back to the user in the callback function with each event callback.

Return:

Positive, non-zero value if successful that is the LSEID that must be used to identify the connection to the remote stream endpoint in all other endpoint related functions.

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

> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

etGAVD_Open_End_Point_Confirmation

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.

GAVD_Close_End_Point

This function is used to close a connection to an endpoint that was previously opened via a call to GAVD_Register_End_Point() or GAVD_Open_Remote_End_Point().

Note:

Calling this function on a local endpoint that was registered via the GAVD_Register_End_Point() function does NOT remove the endpoint from the system, it merely disconnects the currently connected client (i.e., the end point is still present in the system and can be discovered and connected to).

Prototype:

int BTPSAPI **GAVD_Close_End_Point**(unsigned int BluetoothStackID, unsigned int LSEID)

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize().
LSEID	Endpoint ID of the local stream endpoint to close. This value was returned from a successful call to the GAVD_Register_End_Point() or GAVD_Open_Remote_End_Point() function.

Return:

Zero if successful.

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

BTGAVD_ERROR_INVALID_OPERATION
BTGAVD_ERROR_NOT_INITIALIZED
BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

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.

GAVD_Set_Configuration_Response

This function is used to respond to a request from a remote GAVD device to set the configuration of a stream endpoint.

Prototype:

int BTPSAPI **GAVD_Set_Configuraiton_Response**(unsigned int BluetoothStackID, unsigned int LSEID, GAVD_Service_Category_t FirstFailingServiceCategory, unsigned int ErrorCode)

Parameters:

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

LSEID   Local stream endpoint ID that is being configured.

FirstFailingServiceCategory This represents the first service category that was requested to be configured that was unable to be configured (scNone if configuration was successful). Defined by the following enumeration:
typedef enum
{
  scNone,
  scMediaTransport,
  scReporting,
  scRecovery,
  scContentProtection,
  scHeaderCompression,
  scMultiplexing,
  scMediaCodec,
  scUnknown
} GAVD_Service_Category_t;

ErrorCode   Error response code of the configuration response. This value is one of the following values:

                            GAVD_AVDTP_ERROR_SUCCESS
                            GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
                            GAVD_AVDTP_ERROR_BAD_LENGTH
                            GAVD_AVDTP_ERROR_BAD_ACP_SEID
                            GAVD_AVDTP_ERROR_SEP_IN_USE
                            GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
                            GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
                            GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
                            GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
                            GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
                            GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
                            GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
                                       _FORMAT
                            GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
                            GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
                            GAVD_AVDTP_ERROR_BAD_CP_FORMAT

> GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
> GAVD_AVDTP_ERROR_UNSUPPORTED
> _CONFIGURATION
> GAVD_AVDTP_ERROR_BAD_STATE
> GAVD_AVDTP_ERROR_TIMEOUT

Return:

Zero if successful.

An error code if negative; one of the following values:
> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

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.

GAVD_Start_Stream_Request

This function is used to request to start one or more streams on the remote GAVD device.

Prototype:

int BTPSAPI **GAVD_Start_Stream_Request**(unsigned int BluetoothStackID,
   unsigned int NumberStreams, unsigned int LSEID[])

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize().
NumberStreams	Number of streams to start. This specifies the number of local stream endpoint identifiers that are present in the LSEID parameter array.
LSEID[]	List of local stream endpoint IDs to start. This parameter must point to (at least) the number of local stream endpoints specified by the NumberStreams parameter.

Return:

Zero if successful.

An error code if negative; one of the following values:
> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_INSUFFICIENT_RESOURCES
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID

<div align="center">BTGAVD_ERROR_INVALID_PARAMETER</div>

Possible Events:
etGAVD_Start_Confirmation
etGAVD_Close_End_Point_Indication

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.

GAVD_Start_Stream_Response

This function is used to respond to a request from a remote GAVD device to start one or more streams.

Note:

This function can ONLY be called from within the context of the Stream Endpoint Event Callback function in response to an Endpoint Start Request. Failure to call this function in the event callback during a Stream Endpoint Start Request Event will cause the GAVD Profile to respond automatically (with an error response), which means the stream(s) will not have been started.

Prototype:

int BTPSAPI **GAVD_Start_Stream_Response**(unsigned int BluetoothStackID, unsigned int LSEID, unsigned int ErrorCode)

Parameters:

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

LSEID   Local stream endpoint ID of the endpoint that this response is for.

ErrorCode   Error response code of the start stream response. This value is one of the following values:

GAVD_AVDTP_ERROR_SUCCESS
GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
GAVD_AVDTP_ERROR_BAD_LENGTH
GAVD_AVDTP_ERROR_BAD_ACP_SEID
GAVD_AVDTP_ERROR_SEP_IN_USE
GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
_FORMAT

GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
GAVD_AVDTP_ERROR_BAD_CP_FORMAT
GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
GAVD_AVDTP_ERROR_UNSUPPORTED
_CONFIGURATION
GAVD_AVDTP_ERROR_BAD_STATE
GAVD_AVDTP_ERROR_TIMEOUT

Return:

Zero if successful.

An error code if negative; one of the following values:
BTGAVD_ERROR_INVALID_OPERATION
BTGAVD_ERROR_NOT_INITIALIZED
BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Close_End_Point_Indication

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.

GAVD_Suspend_Stream_Request

This function is used to request to suspend one or more streams on the remote GAVD device.

Prototype:

int BTPSAPI **GAVD_Suspend_Stream_Request**(unsigned int BluetoothStackID, unsigned int NumberStreams, unsigned int LSEID[])

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize().
NumberStreams	Number of streams to suspend. This specifies the number of local stream endpoint identifiers that are present in the LSEID parameter array.
LSEID[]	List of local stream endpoint IDs to suspend. This parameter must point to (at least) the number of local stream endpoints specified by the NumberStreams parameter.

Return:

Zero if successful.

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

> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_INSUFFICIENT_RESOURCES
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

etGAVD_Suspend_Confirmation
etGAVD_Close_End_Point_Indication

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.

GAVD_Suspend_Stream_Response

This function is used to respond to a request from a remote GAVD device to suspend one or more streams.

Note:

This function can ONLY be called from within the context of the Stream Endpoint Event Callback function in response to an Endpoint Suspend Request. Failure to call this function in the event callback during a Stream Endpoint Suspend Request Event will cause the GAVD Profile to respond automatically (with an error response), which means the stream(s) will not have been suspended.

Prototype:

int BTPSAPI **GAVD_Suspend_Stream_Response**(unsigned int BluetoothStackID, unsigned int LSEID, unsigned int ErrorCode)

Parameters:

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

LSEID   Local stream endpoint ID of the endpoint that this response is for.

ErrorCode   Error response code of the suspend stream response. This value is one of the following values:

> GAVD_AVDTP_ERROR_SUCCESS
> GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
> GAVD_AVDTP_ERROR_BAD_LENGTH
> GAVD_AVDTP_ERROR_BAD_ACP_SEID
> GAVD_AVDTP_ERROR_SEP_IN_USE
> GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
> GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
> GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT

GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
_FORMAT
GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
GAVD_AVDTP_ERROR_BAD_CP_FORMAT
GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
GAVD_AVDTP_ERROR_UNSUPPORTED
_CONFIGURATION
GAVD_AVDTP_ERROR_BAD_STATE
GAVD_AVDTP_ERROR_TIMEOUT

Return:

Zero if successful.

An error code if negative; one of the following values:
BTGAVD_ERROR_INVALID_OPERATION
BTGAVD_ERROR_NOT_INITIALIZED
BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Close_End_Point_Indication

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.

GAVD_Reconfigure_Request

This function is used to request the reconfiguration of an endpoint on a remote GAVD
device.

Note:

The only service capabilities that are allowed to be reconfigured are Media Codec
Capabilities and Content Protection Capabilities.

Prototype:

int BTPSAPI **GAVD_Reconfigure_Request**(unsigned int BluetoothStackID,
unsigned int LSEID, unsigned int NumberServiceCapabilities,
GAVD_Service_Capabilities_Info_t ServiceCapabilities[])

Parameters:

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

LSEID	Local stream endpoint ID of the endpoint that is connected and is being reconfigured.
NumberServiceCapabilities	Number of service capabilities to reconfigure. This value specifies the number of Service Capabilities pointed to by the ServiceCapabilities parameter.
ServiceCapabilities[]	Array of service capabilities to reconfigure. This is defined by the following structure:

```
typedef struct
{
  GAVD_Service_Category_t    ServiceCategory;
  union
  {
    GAVD_Recovery_Info_Element_Data_t
            GAVD_Recovery_Info_Element_Data;
    GAVD_Content_Protection_Info_Element_Data_t
            GAVD_Content_Protection_Info_Element_Data;
    GAVD_Header_Compression_Info_Element_Data_t
            GAVD_Header_Compression_Info_Element_Data;
    GAVD_Multiplexing_Info_Element_Data_t
            GAVD_Multiplexing_Info_Element_Data;
    GAVD_Media_Codec_Info_Element_Data_t
            GAVD_Media_Codec_Info_Element_Data;
    GAVD_Raw_Info_Element_Data_t
            GAVD_Raw_Info_Element_Data;
  } InfoElement;
} GAVD_Service_Capabilities_Info_t;
```

Return:

Zero if successful.

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

BTGAVD_ERROR_INVALID_OPERATION
BTGAVD_ERROR_INSUFFICIENT_RESOURCES
BTGAVD_ERROR_NOT_INITIALIZED
BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Reconfigure_Confirmation
etGAVD_Close_End_Point_Indication

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.

GAVD_Reconfigure_Response

This function is used to respond to a request from a remote GAVD device to reconfigure a stream.

Prototype:

int BTPSAPI **GAVD_Reconfigure_Response**(unsigned int BluetoothStackID, unsigned int LSEID, GAVD_Service_Category_t FirstFailingServiceCategory, unsigned int ErrorCode)

Parameters:

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

LSEID   Local stream endpoint ID of the endpoint that is connected to the stream that is being reconfigured.

FirstFailingServiceCategory   This represents the first service category that was requested to be reconfigured that was unable to be reconfigured (scNone if reconfiguration was successful). Defined by the following enumeration:
typedef enum
{
  scNone,
  scMediaTransport,
  scReporting,
  scRecovery,
  scContentProtection,
  scHeaderCompression,
  scMultiplexing,
  scMediaCodec,
  scUnknown
} GAVD_Service_Category_t;

ErrorCode   Error response code of the reconfigure response. This value is one of the following values:

      GAVD_AVDTP_ERROR_SUCCESS
      GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
      GAVD_AVDTP_ERROR_BAD_LENGTH
      GAVD_AVDTP_ERROR_BAD_ACP_SEID
      GAVD_AVDTP_ERROR_SEP_IN_USE
      GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
      GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
      GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
      GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
      GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
      GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
      GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
            _FORMAT
      GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
      GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT

GAVD_AVDTP_ERROR_BAD_CP_FORMAT
GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
GAVD_AVDTP_ERROR_UNSUPPORTED
_CONFIGURATION
GAVD_AVDTP_ERROR_BAD_STATE
GAVD_AVDTP_ERROR_TIMEOUT

Return:

Zero if successful.

An error code if negative; one of the following values:
BTGAVD_ERROR_INVALID_OPERATION
BTGAVD_ERROR_NOT_INITIALIZED
BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Close_End_Point_Indication

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.

GAVD_Security_Control_Request

This function is used to request the change of the security settings for a stream.

Prototype:

int BTPSAPI **GAVD_Security_Control_Request**(unsigned int BluetoothStackID, unsigned int LSEID, unsigned int ContentProtectionDataLength, Byte_t *ContentProtectionData)

Parameters:

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

LSEID   Local stream endpoint ID of the endpoint that is connected to the stream.

ContentProtectionDataLenth   Length of the security data. This specifies the length (in bytes) of the Content Protection Data (this is the length of the buffer that is pointed to by the ContentProtectionData parameter).

ContentProtectionData   Pointer to the Content Protection Data. This is pointer to a buffer that contains the Content Protection Data (of length specified by the ContentProtectionDataLength parameter).

Return:

Zero if successful.

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

> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_INSUFFICIENT_RESOURCES
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

etGAVD_Security_Control_Confirmation
etGAVD_Close_End_Point_Indication

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.

GAVD_Security_Control_Response

This function is used to respond to a request from a remote GAVD device to change the security settings for a stream.

Prototype:

int BTPSAPI **GAVD_Security_Control_Response**(unsigned int BluetoothStackID, unsigned int LSEID, unsigned int ErrorCode, unsigned int ContentProtectionDataLength, Byte_t *ContentProtectionData)

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize().
LSEID	Local stream endpoint ID of the endpoint that is connected to the stream.
ErrorCode	Error response code of the security control response. This value is one of the following values:

> GAVD_AVDTP_ERROR_SUCCESS
> GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
> GAVD_AVDTP_ERROR_BAD_LENGTH
> GAVD_AVDTP_ERROR_BAD_ACP_SEID
> GAVD_AVDTP_ERROR_SEP_IN_USE
> GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
> GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
> GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
> GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
> GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
> GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
> GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
> _FORMAT
> GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT

                                   GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
                                   GAVD_AVDTP_ERROR_BAD_CP_FORMAT
                                   GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
                                   GAVD_AVDTP_ERROR_UNSUPPORTED
                                          _CONFIGURATION
                                   GAVD_AVDTP_ERROR_BAD_STATE
                                   GAVD_AVDTP_ERROR_TIMEOUT

ContentProtectionDataLenth	Length of the security data. This specifies the length (in bytes) of the Content Protection Data (this is the length of the buffer that is pointed to by the ContentProtectionData parameter).
ContentProtectionData	Pointer to the Content Protection Data. This is pointer to a buffer that contains the Content Protection Data (of length specified by the ContentProtectionDataLength parameter).

Return:

Zero if successful.

An error code if negative; one of the following values:
                                   BTGAVD_ERROR_INVALID_OPERATION
                                   BTGAVD_ERROR_INSUFFICIENT_RESOURCES
                                   BTGAVD_ERROR_NOT_INITIALIZED
                                   BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
                                   BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Close_End_Point_Indication

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.

GAVD_Abort_Stream_Request

This function is used to request to abort one or more streams on the remote GAVD device.

Prototype:

int BTPSAPI **GAVD_Abort_Stream_Request**(unsigned int BluetoothStackID, unsigned int LSEID)

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize().
LSEID	Local stream endpoint ID of the endpoint that is connected to the stream.

Return:

Zero if successful.

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

BTGAVD_ERROR_INVALID_OPERATION
BTGAVD_ERROR_NOT_INITIALIZED
BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

etGAVD_Abort_Confirmation
etGAVD_Close_End_Point_Indication

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.

GAVD_Data_Write

This function is used to send data over the specified (started) stream.

Prototype:

int BTPSAPI **GAVD_Data_Write**(unsigned int BluetoothStackID, unsigned int LSEID, Boolean_t Marker, Byte_t PayloadType, DWord_t TimeStamp, Word_t DataLength, Byte_t *DataBuffer)

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize().
LSEID	Local stream endpoint ID of the endpoint that is connected to the stream.
Marker	Specifies whether the marker bit (used by certain codec payload formats) is to be used with the Data written.
PayloadType	Indicates the content type of the data.
TimeStamp	Holds time stamp information.
DataLength	Specifies the length of the payload. This parameter specifies the length (in bytes) of the payload data that is to be written.
DataBuffer	Points to the payload data. This parameter is a pointer to the payload data to be written to the specified stream endpoint. This pointer must point to (at least) the number of bytes specified by the DataLength parameter.

Return:

Zero if successful.

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

>   BTGAVD_ERROR_INVALID_OPERATION
>   BTGAVD_ERROR_L2CAP_MTU_EXCEEDED
>   BTGAVD_ERROR_INSUFFICIENT_RESOURCES
>   BTGAVD_ERROR_NOT_INITIALIZED
>   BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
>   BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

etGAVD_Close_End_Point_Indication

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.

GAVD_Sender_Report_Data_Write

This function is used to send Sender Report data over the specified stream.

Prototype:

int BTPSAPI **GAVD_Sender_Report_Data_Write**(unsigned int BluetoothStackID,
    unsigned int LSEID, GAVD_Sender_Info_t *SenderInfo,
    unsigned int NumberReportBlocks, GAVD_Report_Block_t *ReportBlocks,
    Word_t ExtensionDataLength, DWord_t *ExtensionData)

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize().
LSEID	Local stream endpoint ID of the endpoint that is sourcing the data.

SenderInfo	Points to the Sender Information in the Sender Report. This is defined by the following structure:
	typedef struct
	{
	  DWord_t NTPTimeStampMSW;
	  DWord_t NTPTimeStampLSW;
	  DWord_t RTPTimeStamp;
	  DWord_t PacketCount;
	  DWord_t OctetCount;
	} GAVD_Sender_Info_t;

NumberReportBlocks	Indicates the number of report blocks pointed to by the ReportBlocks parameter.
ReportBlocks	Points to the report blocks. This is defined by the following structure: typedef struct

```
                                {
                                  DWord_t     SourceID;
                                  Byte_t      FractionLost;
                                  DWord_t     TotalPacketsLost;
                                  Word_t      HighestSeqNumReceivedCycleCount;
                                  Word_t      HighestSeqNumReceived;
                                  DWord_t     IntervalJitter;
                                  DWord_t     LastReport;
                                  DWord_t     DelaySinceLastReport;
                                } GAVD_Report_Block_t;
```

ExtensionDataLength   Indicates how many 32-bit words are in the extension data.

ExtensionData   Points to the extension data. This parameter points to an array
                          of 32-bit words. The number of 32-bit words pointed to by this
                          parameter is given by the ExtensionDataLength parameter.

Return:

Zero if successful.

An error code if negative; one of the following values:
     BTGAVD_ERROR_INVALID_OPERATION
     BTGAVD_ERROR_L2CAP_MTU_EXCEEDED
     BTGAVD_ERROR_INSUFFICIENT_RESOURCES
     BTGAVD_ERROR_NOT_INITIALIZED
     BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
     BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Close_End_Point_Indication

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.

GAVD_Receiver_Report_Data_Write

This function is used to send Receiver Report data over the specified stream.

Prototype:

int BTPSAPI **GAVD_Receiver_Report_Data_Write**(unsigned int BluetoothStackID,
 unsigned int LSEID, unsigned int NumberReportBlocks,
 GAVD_Report_Block_t *ReportBlocks, Word_t ExtensionDataLength,
 DWord_t *ExtensionData)

Parameters:

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

LSEID   Local stream endpoint ID of the endpoint that is sourcing the data.

NumberReportBlocks   Indicates the number of report blocks pointed to by the ReportBlocks parameter.

ReportBlocks   Points to the report blocks. This is defined by the following structure:

```
typedef struct
{
  DWord_t      SourceID;
  Byte_t       FractionLost;
  DWord_t      TotalPacketsLost;
  Word_t       HighestSeqNumReceivedCycleCount;
  Word_t       HighestSeqNumReceived;
  DWord_t      IntervalJitter;
  DWord_t      LastReport;
  DWord_t      DelaySinceLastReport;
} GAVD_Report_Block_t;
```

ExtensionDataLength   Indicates how many 32-bit words are in the extension data.

ExtensionData   Points to the extension data. This parameter points to an array of 32-bit words. The number of 32-bit words pointed to by this parameter is given by the ExtensionDataLength parameter.

Return:

Zero if successful.

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

> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_L2CAP_MTU_EXCEEDED
> BTGAVD_ERROR_INSUFFICIENT_RESOURCES
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

etGAVD_Close_End_Point_Indication

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.

GAVD_SDES_Report_Data_Write

This function is used to send SDES Report data over the specified stream.

Prototype:

int BTPSAPI **GAVD_SDES_Report_Data_Write**(unsigned int BluetoothStackID,
    unsigned int LSEID, unsigned int NumberSDESChunks,
    GAVD_SDES_Chunk_t *SDESChunks)

Parameters:

BluetoothStackID[1]	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize().
LSEID	Local stream endpoint ID of the endpoint that is sourcing the data.
NumberSDESChunks	Indicates the number of SDES chucks pointed to by the SDESChunks parameter.

SDESChunks	Points to the SDES chunks. This is defined by the following structure:
	typedef struct
	{
	  DWord_t SourceID;
	  unsigned int NumberSDESItems;
	  GAVD_SDES_Item_t *SDESItems;
	} GAVD_SDES_Chunk_t;
	
	where, SDESItems is defined by the following structure:
	   typedef struct
	   {
	    Byte_t ItemDescriptor;
	    Byte_t ItemLength;
	    Byte_t *ItemData;
	   } GAVD_SDES_Item_t;

Return:

Zero if successful.

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

> BTGAVD_ERROR_INVALID_OPERATION
> BTGAVD_ERROR_L2CAP_MTU_EXCEEDED
> BTGAVD_ERROR_INSUFFICIENT_RESOURCES
> BTGAVD_ERROR_NOT_INITIALIZED
> BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
> BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:
etGAVD_Close_End_Point_Indication

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.

GAVD_Recovery_Data_Write

This function is used to send RTP FEC recovery data over the specified stream.

Prototype:

int BTPSAPI **GAVD_Recovery_Data_Write**(unsigned int BluetoothStackID, unsigned int LSEID, Boolean_t Marker, Byte_t PayloadType, DWord_t TimeStamp, GAVD_FEC_Block_t *FECBlock)

Parameters:

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

LSEID   Local stream endpoint ID of the endpoint that is sourcing the data.

Marker   Specifies the marker bit used by certain codec payload formats.

PayloadType   Indicates the content type of the data.

TimeStamp   Holds time stamp information.

FECBlock   Points to the structure with the information to send. This is defined by the following structure:

```
typedef struct
{
  Word_t        SequenceNumberBase;
  Word_t        LengthRecovery;
  Byte_t        PayloadTypeRecovery;
  DWord_t       Mask;
  DWord_t       TimeStampRecovery;
  unsigned int  FECDataLength;
  Byte_t        *FECData;
} GAVD_FEC_Block_t;
```

Return:

Zero if successful.

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

BTGAVD_ERROR_INVALID_OPERATION
BTGAVD_ERROR_L2CAP_MTU_EXCEEDED
BTGAVD_ERROR_INSUFFICIENT_RESOURCES
BTGAVD_ERROR_NOT_INITIALIZED
BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
BTGAVD_ERROR_INVALID_PARAMETER

Possible Events:

etGAVD_Close_End_Point_Indication

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.

GAVD_Get_Server_Connection_Mode

Retrieves the current GAVD/AVDTP Server Connection Mode. The default is gsmAutomaticAccept. This function is used for GAVD/AVDTP Servers which use Bluetooth Security Mode 2.

Prototype:

int BTPSAPI **GAVD_Get_Server_Connection_Mode**(unsigned int BluetoothStackID, GAVD_Server_Connection_Mode_t *ServerConnectionMode)

Parameters:

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

ServerConnectionMode   Pointer Server Connection Mode variable to put the current GAVD/AVDTP Server Connection Mode in. The possible values returned are:
>   gsmAutomaticAccept
>   gsmAutomaticReject
>   gsmManualAccept

Return:

Zero if successful.

An error code if negative; one of the following values:
>   BTGAVD_ERROR_NOT_INITIALIZED
>   BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
>   BTGAVD_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.

GAVD_Set_Server_Connection_Mode

Changes the current GAVD/AVDTP Server Connection Mode. The default is gsmAutomaticAccept. This function is used for GAVD/AVDTP Servers which use Bluetooth Security Mode 2.

Prototype:

>int BTPSAPI **GAVD_Set_Server_Connection_Mode**(unsigned int BluetoothStackID,
>   GAVD_Server_Connection_Mode_t ServerConnectionMode, GAVD_Event_Callback_t
>   GAVDEventCallback, unsigned long CallbackParameter)

Parameters:

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

ServerConnectionMode   Value to chang the current GAVD/AVDTP Server Connection
                          Mode to. If the server mode is anything other than
                          gsmManualAccept then the final two parameters are ignored.
                          The possible values are:
>   gsmAutomaticAccept
>   gsmAutomaticReject
>   gsmManualAccept

GAVDEventCallback   Event Callback that is used to receive notifications of a
                          Blueooth Connection Request.

CallbackParameter   Parameter to the callback function described above.

Return:

Zero if successful.

An error code if negative; one of the following values:
>   BTGAVD_ERROR_NOT_INITIALIZED
>   BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
>   BTGAVD_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.

GAVD_Get_Data_Queueing_Parameters

Retrieves the current GAVD/L2CAP data queueing parameters. These parameters dictate
how the data packets are queued into L2CAP (when calling the GAVD_Data_Write()
function). This mechanism allows for the ability to implement a streaming type interface
by limiting the number of packets that can queued (simultaneously) in L2CAP. This is
useful to keep L2CAP from infinitely queing packets which can lead to stale data if there
is an issue sending the data to the remote device (i.e. interference).

Notes:

This function sets the queing parameters globally for GAVD. Setting the Queing
parameters for an individual stream endpoint is currently not supported. This is because
multiple stream endpoints could be multiplexed over the same channel.

A value of zero for the QueueLimit member of the L2CAP queing parameters means that there is no queing active (i.e. all packets are queued, regardless of the queue depth).

It is recommended to ALWAYS use the L2CA_QUEUEING_FLAG_DISCARD_OLDEST flag when specifying queing parameters. Although the threshold method can be used (by having the queue fail and waiting for a Data buffer empty indication), it is recommended to allow the data to always be queued and have the oldest data deleted from the queue and have the newest data placed at the end of the queue (i.e. always queue the data).

Prototype:

int BTPSAPI **GAVD_Get_Data_Queueing_Parameters**(unsigned int BluetoothStackID, L2CA_Queueing_Parameters_t *QueueingParameters)

Parameters:

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

QueueingParameters   Pointer to a structure that will contain the currently configured Queing Parameters that are currently used by GAVD. See the L2CAP_Enhanced_Data_Write() function (in Bluetopia Core API documentation) for more information on the values for this parameter.

Return:

Zero if successful.

An error code if negative; one of the following values:
                    BTGAVD_ERROR_NOT_INITIALIZED
                    BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
                    BTGAVD_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.

GAVD_Set_Data_Queueing_Parameters

Sets the current GAVD/L2CAP data queueing parameters. These parameters dictate how the data packets are queued into L2CAP (when calling the GAVD_Data_Write() function). This mechanism allows for the ability to implement a streaming type interface by limiting the number of packets that can queued (simultaneously) in L2CAP. This is useful to keep L2CAP from infinitely queing packets which can lead to stale data if there is an issue sending the data to the remote device (i.e. interference).

Notes:

This function sets the queing parameters globally for GAVD. Setting the Queing parameters for an individual stream endpoint is currently not supported. This is because multiple stream endpoints could be multiplexed over the same channel.

A value of zero for the QueueLimit member of the L2CAP queing parameters means that there is no queing active (i.e. all packets are queued, regardless of the queue depth).

It is recommended to ALWAYS use the L2CA_QUEUEING_FLAG_DISCARD_OLDEST flag when specifying queing parameters. Although the threshold method can be used (by having the queue fail and waiting for a Data buffer empty indication), it is recommended to allow the data to always be queued and have the oldest data deleted from the queue and have the newest data placed at the end of the queue (i.e. always queue the data).

Prototype:

int BTPSAPI **GAVD_Set_Data_Queueing_Parameters**(unsigned int BluetoothStackID, L2CA_Queueing_Parameters_t *QueueingParameters)

Parameters:

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

QueueingParameters   Pointer to a structure that contains the new Queing Parameters to set. See the L2CAP_Enhanced_Data_Write() function (in Bluetopia Core API documentation) for more information on the values for this parameter.

Return:

Zero if successful.

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

                          BTGAVD_ERROR_NOT_INITIALIZED
                          BTGAVD_ERROR_INVALID_BLUETOOTH_STACK_ID
                          BTGAVD_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.

2.2   GAVD Profile Event Callback Prototypes

The event callback functions mentioned in the GAVD Profile functions all accept the callback function described by the following prototype.

GAVD_Event_Callback_t

Prototype of callback function passed in to the GAVD_Connect(), GAVD_Register_End_Point(), and GAVD_Open_Remote_End_Point() functions. This callback function is dispatched whenever events destined for the specified connection/endpoint occur.

Prototype:

void (BTPSAPI ***GAVD_Event_Callback_t**)(unsigned int BluetoothStackID, GAVD_Event_Data_t *GAVD_Event_Data, unsigned long CallbackParameter);

Parameters:

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

GAVD_Event_Data   Data describing the event for which the callback function is called. This is defined by the following structure:

```
typedef struct
{
  GAVD_Event_Type_t    Event_Data_Type;
  Word_t               Event_Data_Size;
  union
  {
    GAVD_Connect_Request_Indication_Data_t
            *GAVD_Connect_Request_Indication_Data;
    GAVD_Connect_Confirmation_Data_t
            *GAVD_Connect_Confirmation_Data;
    GAVD_Disconnect_Indication_Data_t
            *GAVD_Disconnect_Indication_Data;
    GAVD_Discover_Confirmation_Data_t
            *GAVD_Discover_Confirmation_Data;
    GAVD_Get_Capabilities_Confirmation_Data_t
            *GAVD_Get_Capabilities_Confirmation_Data;
    GAVD_Get_Configuration_Confirmation_Data_t
            *GAVD_Get_Configuration_Confirmation_Data;
    GAVD_Set_Configuration_Indication_Data_t
            *GAVD_Set_Configuration_Indication_Data;
    GAVD_Open_End_Point_Indication_Data_t
            *GAVD_Open_End_Point_Indication_Data;
    GAVD_Open_End_Point_Confirmation_Data_t
            *GAVD_Open_End_Point_Confirmation_Data;
    GAVD_Close_End_Point_Indication_Data_t
            *GAVD_Close_End_Point_Indication_Data;
    GAVD_Start_Indication_Data_t
            *GAVD_Start_Indication_Data;
    GAVD_Start_Confirmation_Data_t
            *GAVD_Start_Confirmation_Data;
    GAVD_Suspend_Indication_Data_t
            *GAVD_Suspend_Indication_Data;
```

```
                            GAVD_Suspend_Confirmation_Data_t
                                    *GAVD_Suspend_Confirmation_Data;
                            GAVD_Reconfigure_Indication_Data_t
                                    *GAVD_Reconfigure_Indication_Data;
                            GAVD_Reconfigure_Confirmation_Data_t
                                    *GAVD_Reconfigure_Confirmation_Data;
                            GAVD_Security_Control_Indication_Data_t
                                    *GAVD_Security_Control_Indication_Data;
                            GAVD_Security_Control_Confirmation_Data_t
                                    *GAVD_Security_Control_Confirmation_Data;
                            GAVD_Abort_Indication_Data_t
                                    *GAVD_Abort_Indication_Data;
                            GAVD_Abort_Confirmation_Data_t
                                    *GAVD_Abort_Confirmation_Data;
                            GAVD_Data_Indication_Data_t
                                    *GAVD_Data_Indication_Data;
                            GAVD_Sender_Report_Data_Indication_Data_t
                                    *GAVD_Sender_Report_Data_Indication_Data;
                            GAVD_Receiver_Report_Data_Indication_Data_t
                                    *GAVD_Receiver_Report_Data_Indication_Data;
                            GAVD_SDES_Report_Data_Indication_Data_t
                                    *GAVD_SDES_Report_Data_Indication_Data;
                            GAVD_Recovery_Data_Indication_Data_t
                                    *GAVD_Recovery_Data_Indication_Data;
                            GAVD_Channel_Empty_Indication_Data_t
                                    *GAVD_Channel_Empty_Indication_Data;
                          } Event_Data;
                        } GAVD_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 (e.g., tag value) that was defined in the callback registration. This value is passed to the caller each time the event callback is dispatched.

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   GAVD Profile Events

The possible GAVD Profile events from the Bluetooth stack are listed in the table below and are described in the text that follows:

Event	Description
etGAVD_Connect_Request_Indication	Dispatched when a remote device is requesting a connection to the local device.
etGAVD_Connect_Confirmation	Confirms that a GAVD remote Stream Endpoint Manager connect request has been responded to or has encountered an error.
etGAVD_Disconnect_Indication	Indicates that a local client has been disconnected from a remote Stream Endpoint Manager.
etGAVD_Discover_Confirmation	Confirms the completion of a Stream Discover command that was sent to a remote Stream Endpoint Manager.
etGAVD_Get_Capabilities_Confirmation	Confirms the completion of a Get Capabilities command that was sent to a remote Stream Endpoint Manager.
etGAVD_Get_Configuration_Confirmation	Confirms the completion of a Get Configuration command.
etGAVD_Set_Configuration_Indication	Indicates that a Set Configuration command has been received from a remote Stream Endpoint Manager.
etGAVD_Open_End_Point_Indication	Indicates that an Open Endpoint command has been received from a remote Stream Endpoint Manager.
etGAVD_Open_End_Point_Confirmation	Confirms the completion of an Open Endpoint command that was sent to a remote Stream Endpoint Manager.
etGAVD_Close_End_Point_Indication	Indicates that a Close Endpoint command has been received from a remote Stream Endpoint Manager.
etGAVD_Start_Indication	Indicates that a Start Stream command has been received from a remote Stream Endpoint Manager.
etGAVD_Start_Confirmation	Confirms that a Start Stream command sent to a remote Stream Endpoint Manager was received.
etGAVD_Suspend_Indication	Indicates that a Suspend Stream command has been received from a remote Stream Endpoint Manager.
etGAVD_Suspend_Confirmation	Confirms that a Suspend Stream command sent to a remote Stream Endpoint was received.
etGAVD_Reconfigure_Indication	Indicates that a Reconfigure Stream command has been received from a remote Stream Endpoint Manager.
etGAVD_Reconfigure_Confirmation	Confirms the completion of a Reconfigure Stream command that was sent to a remote Stream Endpoint

	Manager.
etGAVD_Security_Control_Indication	Indicates that a Security Control command has been received from a remote Stream Endpoint Manager.
etGAVD_Security_Control_Confirmation	Confirms the completion of a Security Control command that was sent to a remote Stream Endpoint Manager.
etGAVD_Abort_Indication	Indicates that an Abort Stream command has been received from a remote Stream Endpoint Manager.
etGAVD_Abort_Confirmation	Confirms the completion of an Abort Stream command that was sent to a remote Stream Endpoint Manager.
etGAVD_Data_Indication	Indicates that data has been received from a remote Endpoint.
etGAVD_Sender_Report_Data_Indication	Indicates that Sender Report data has been received from a remote Endpoint.
etGAVD_Receiver_Report_Data_Indication	Indicates that Receiver Report data has been received from a remote Endpoint.
etGAVD_SDES_Report_Data_Indication	Indicates that SDES Report data has been received from a remote Endpoint.
etGAVD_Recovery_Data_Indication	Indicates that Recovery data has been received from a remote Endpoint.
etGAVD_Channel_Empty_Indication	Dispatched by a GAVD/AVDTP entity to the local application when a Stream End Point no longer has any data queued to be sent on the Data Channel.

etGAVD_Connect_Request_Indication

This event is dispatched when a remote device is requesting a connection to the local device. This event is only dispatched to servers in Manual Accept Mode. This event must be responded to with a call to GAVD_Connect_Request_Response() in order to accept or reject the pending connection request.

Return Structure:

typedef struct
{
  BD_ADDR_t BD_ADDR;
} GAVD_Connect_Request_Indication_Data_t;

Event Parameters:

BD_ADDR   Bluetooth Address of the Remote Device that is attempting to
                       connect.

etGAVD_Connect_Confirmation

Confirms that a Connect request has been responded to or has encountered an error.

Return Structure:

typedef struct
{
  unsigned int GAVDID;
  unsigned int Status;
} GAVD_Connect_Confirmation_Data_t;

Event Parameters:

GAVDID   Identifier of the local GAVD client.

Status   Success or failure error code. This is one of the following
                                values:

                                        GAVD_STATUS_SUCCESS
                                        GAVD_STATUS_CONNECTION_TIMEOUT
                                        GAVD_STATUS_CONNECTION_REFUSED
                                        GAVD_STATUS_UNKNOWN_ERROR

etGAVD_Disconnect_Indication

Indicates that a local client has been disconnected from a remote Stream Endpoint
Manager.

Return Structure:

typedef struct
{
  unsigned int   GAVDID;
  BD_ADDR_t   BD_ADDR;
} GAVD_Disconnect_Indication_Data_t;

Event Parameters:

GAVDID   Identifier of the local GAVD client.

BD_ADDR   Bluetooth Device address of the remote device that was
                                disconnected.

etGAVD_Discover_Confirmation

Confirms the completion of a Stream Discover command that was sent to a remote
Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int                         GAVDID;
  unsigned int                         ErrorCode;
  unsigned int                         NumberRemoteEndPoints;
  GAVD_Remote_End_Point_Data_t    *RemoteEndPoints;
} GAVD_Discover_Confirmation_Data_t;
```

Event Parameters:

GAVDID   Identifier of the local GAVD client.

ErrorCode   Error response code of the discover confirmation. This value is one of the following values:

> GAVD_AVDTP_ERROR_SUCCESS
> GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
> GAVD_AVDTP_ERROR_BAD_LENGTH
> GAVD_AVDTP_ERROR_BAD_ACP_SEID
> GAVD_AVDTP_ERROR_SEP_IN_USE
> GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
> GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
> GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
> GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
> GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
> GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
> GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
>   _FORMAT
> GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
> GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
> GAVD_AVDTP_ERROR_BAD_CP_FORMAT
> GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
> GAVD_AVDTP_ERROR_UNSUPPORTED
>   _CONFIGURATION
> GAVD_AVDTP_ERROR_BAD_STATE
> GAVD_AVDTP_ERROR_TIMEOUT

NumberRemoteEndPoints   Number of remote endpoints that are contained in the array pointed to by the RemoteEndPoints parameter.

RemoteEndPoints   Pointer to an array of stream endpoints that are available on the remote GAVD device. This is defined by the following structure:

```
typedef struct
{
  unsigned int              RSEID;
  GAVD_TSEP_t               TSEP;
  GAVD_Media_Type_t     MediaType;
  Boolean_t                  InUse;
} GAVD_Remote_End_Point_Data_t;
```

where;

RSEID is the identifier of the remote stream endpoint.

TSEP specifies the type (source or sink) of the stream endpoint.

MediaType specifies the media type of the stream endpoint,
using the following enumeration:

```
    typedef enum
    {
      mtAudio,
      mtVideo,
      mtMultimedia
    } GAVD_Media_Type_t;
```

InUse indicates whether the stream is currently in use or not.

etGAVD_Get_Capabilities_Confirmation

Confirms the completion of a Get Capabilities command that was sent to a remote Stream
Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int                          GAVDID;
  unsigned int                          RSEID;
  unsigned int                          ErrorCode;
  unsigned int                          NumberServiceCapabilities;
  GAVD_Service_Capabilities_Info_t   *ServiceCapabilities;
} GAVD_Get_Capabilities_Confirmation_Data_t;
```

Event Parameters:

GAVDID   Identifier of the local GAVD client.

RSEID   Identifier of the remote stream endpoint.

ErrorCode   Error response code of the get capabilities response. This
                        value is one of the following values:

                            GAVD_AVDTP_ERROR_SUCCESS
                            GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
                            GAVD_AVDTP_ERROR_BAD_LENGTH
                            GAVD_AVDTP_ERROR_BAD_ACP_SEID
                            GAVD_AVDTP_ERROR_SEP_IN_USE

<div style="text-align:center">

GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
_FORMAT
GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
GAVD_AVDTP_ERROR_BAD_CP_FORMAT
GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
GAVD_AVDTP_ERROR_UNSUPPORTED
_CONFIGURATION
GAVD_AVDTP_ERROR_BAD_STATE
GAVD_AVDTP_ERROR_TIMEOUT

</div>

NumberServiceCapabilities	Number of service capabilities that are contained in the array pointed to by the ServiceCapabilities parameter.
ServiceCapabilities	Pointer to an array of service capabilities. This is defined by the following structure:

```
typedef struct
{
  GAVD_Service_Category_t ServiceCategory;
  union
  {
    GAVD_Recovery_Info_Element_Data_t
            GAVD_Recovery_Info_Element_Data;
    GAVD_Content_Protection_Info_Element_Data_t
            GAVD_Content_Protection_Info_Element_Data;
    GAVD_Header_Compression_Info_Element_Data_t
            GAVD_Header_Compression_Info_Element_Data;
    GAVD_Multiplexing_Info_Element_Data_t
            GAVD_Multiplexing_Info_Element_Data;
    GAVD_Media_Codec_Info_Element_Data_t
            GAVD_Media_Codec_Info_Element_Data;
    GAVD_Raw_Info_Element_Data_t
            GAVD_Raw_Info_Element_Data;
  } InfoElement;
} GAVD_Service_Capabilities_Info_t;
```

etGAVD_Get_Configuration_Confirmation

Confirms the completion of a Get Configuration command.

Return Structure:

```
typedef struct
{
  unsigned int                        GAVDID;
  unsigned int                        RSEID;
  unsigned int                        ErrorCode;
  unsigned int                        NumberServiceCapabilities;
  GAVD_Service_Capabilities_Info_t    *ServiceCapabilities;
} GAVD_Get_Configuration_Confirmation_Data_t;
```

Event Parameters:

GAVDID   Identifier of the local GAVD client.

RSEID   Identifier of the remote stream endpoint.

ErrorCode   Error response code of the get configuration response. This
                             value is one of the following values:

> GAVD_AVDTP_ERROR_SUCCESS
> GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
> GAVD_AVDTP_ERROR_BAD_LENGTH
> GAVD_AVDTP_ERROR_BAD_ACP_SEID
> GAVD_AVDTP_ERROR_SEP_IN_USE
> GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
> GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
> GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
> GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
> GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
> GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
> GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
>   _FORMAT
> GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
> GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
> GAVD_AVDTP_ERROR_BAD_CP_FORMAT
> GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
> GAVD_AVDTP_ERROR_UNSUPPORTED
>   _CONFIGURATION
> GAVD_AVDTP_ERROR_BAD_STATE
> GAVD_AVDTP_ERROR_TIMEOUT

NumberServiceCapabilities   Number of service capabilities that are contained in the array
                             pointed to by the ServiceCapabilities parameter.

ServiceCapabilities   Pointer to an array of service capabilities. This is defined by
                             the following structure:
```
typedef struct
{
  GAVD_Service_Category_t ServiceCategory;
  union
  {
    GAVD_Recovery_Info_Element_Data_t
            GAVD_Recovery_Info_Element_Data;
```

```
            GAVD_Content_Protection_Info_Element_Data_t
                  GAVD_Content_Protection_Info_Element_Data;
            GAVD_Header_Compression_Info_Element_Data_t
                  GAVD_Header_Compression_Info_Element_Data;
            GAVD_Multiplexing_Info_Element_Data_t
                  GAVD_Multiplexing_Info_Element_Data;
            GAVD_Media_Codec_Info_Element_Data_t
                  GAVD_Media_Codec_Info_Element_Data;
            GAVD_Raw_Info_Element_Data_t
                  GAVD_Raw_Info_Element_Data;
          } InfoElement;
        } GAVD_Service_Capabilities_Info_t;
```

etGAVD_Set_Configuration_Indication

Indicates that a Set Configuration command has been received from a remote Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  BD_ADDR_t                        BD_ADDR;
  unsigned int                     LSEID;
  unsigned int                      NumberServiceCapabilities;
  GAVD_Service_Capabilities_Info_t  *ServiceCapabilities;
} GAVD_Set_Configuration_Indication_Data_t;
```

Event Parameters:

BD_ADDR	Bluetooth address of the remote device.
LSEID	Identifier of the local stream endpoint.
NumberServiceCapabilities	Number of service capabilities that are contained in the array pointed to by the ServiceCapabilities parameter.
ServiceCapabilities	Pointer to an array of service capabilities. This is defined by the following structure:

```
typedef struct
{
  GAVD_Service_Category_t ServiceCategory;
  union
  {
    GAVD_Recovery_Info_Element_Data_t
          GAVD_Recovery_Info_Element_Data;
    GAVD_Content_Protection_Info_Element_Data_t
          GAVD_Content_Protection_Info_Element_Data;
    GAVD_Header_Compression_Info_Element_Data_t
          GAVD_Header_Compression_Info_Element_Data;
    GAVD_Multiplexing_Info_Element_Data_t
          GAVD_Multiplexing_Info_Element_Data;
    GAVD_Media_Codec_Info_Element_Data_t
          GAVD_Media_Codec_Info_Element_Data;
```

```
                              GAVD_Raw_Info_Element_Data_t
                                   GAVD_Raw_Info_Element_Data;
                          } InfoElement;
                      } GAVD_Service_Capabilities_Info_t;
```

etGAVD_Open_End_Point_Indication

Indicates that an Open Endpoint command has been received from a remote Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int   LSEID;
  Word_t         MediaOutMTU;
  Word_t         ReportingOutMTU;
  Word_t         RecoveryOutMTU;
} GAVD_Open_End_Point_Indication_Data_t;
```

Event Parameters:

LSEID	Identifier of the local stream endpoint.
MediaOutMTU	Maximum output MTU size for data payload. This is the largest negotiated MTU size for the data channel.
ReportingOutMTU	Maximum output MTU size for reporting data payload. This is the largest negotiated MTU size for the reporting channel.
RecoveryOutMTU	Maximum output MTU size for recovery data payload. This is the largest negotiated MTU size for the recovery channel.

etGAVD_Open_End_Point_Confirmation

Confirms the completion of an Open Endpoint command that was sent to a remote Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int              LSEID;
  Word_t                    MediaOutMTU;
  Word_t                    ReportingOutMTU;
  Word_t                    RecoveryOutMTU;
  unsigned int              ErrorCode;
  GAVD_Service_Category_t   FirstFailingServiceCategory;
} GAVD_Open_End_Point_Confirmation_Data_t;
```

Event Parameters:

LSEID	Identifier of the local stream endpoint.

MediaOutMTU	Maximum output MTU size for data channel. This value will be used for the negotiation of the MTU of the data channel.
ReportingOutMTU	Maximum output MTU size for reporting data channel. This value will be used for the negotiation of the MTU of the reporting channel.
RecoveryOutMTU	Maximum output MTU size for recovery data channel. This value will be used for the negotiation of the MTU of the recovery channel.
ErrorCode	Error response code of the configuration confirmation. This value is one of the following values:

        GAVD_AVDTP_ERROR_SUCCESS
        GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
        GAVD_AVDTP_ERROR_BAD_LENGTH
        GAVD_AVDTP_ERROR_BAD_ACP_SEID
        GAVD_AVDTP_ERROR_SEP_IN_USE
        GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
        GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
        GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
        GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
        GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
        GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
        GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
                _FORMAT
        GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
        GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
        GAVD_AVDTP_ERROR_BAD_CP_FORMAT
        GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
        GAVD_AVDTP_ERROR_UNSUPPORTED
                _CONFIGURATION
        GAVD_AVDTP_ERROR_BAD_STATE
        GAVD_AVDTP_ERROR_TIMEOUT

FirstFailingServiceCategory	This represents the first service category that was requested to be configured that was unable to be configured (scNone if configuration was successful). Defined by the following enumeration:

```
typedef enum
{
{
  scNone,
  scMediaTransport,
  scReporting,
  scRecovery,
  scContentProtection,
  scHeaderCompression,
  scMultiplexing,
  scMediaCodec,
  scUnknown
```

                                    } GAVD_Service_Category_t;

etGAVD_Close_End_Point_Indication

Indicates that a Close Endpoint command has been received from a remote Stream Endpoint Manager.

Return Structure:

typedef struct
 {
   unsigned int   LSEID;
 } GAVD_Close_End_Point_Indication_Data_t;

Event Parameters:

LSEID   Identifier of the local stream endpoint.

etGAVD_Start_Indication

Indicates that a Start Stream command has been received from a remote Stream Endpoint Manager.

Return Structure:

typedef struct
 {
   unsigned int   LSEID;
 } GAVD_Start_Indication_Data_t;

Event Parameters:

LSEID   Identifier of the local stream endpoint.

etGAVD_Start_Confirmation

Confirms that a Start Stream command sent to a remote Stream Endpoint Manager was received.

Return Structure:

typedef struct
 {
   unsigned int   LSEID;
   unsigned int   ErrorCode;
   unsigned int   FirstFailingLSEID;
 } GAVD_Start_Confirmation_Data_t;

Event Parameters:

LSEID   Identifier of the local stream endpoint.

ErrorCode   Error response code of the start stream endpoint confirmation.
                            This value is one of the following values:

                                GAVD_AVDTP_ERROR_SUCCESS
                                GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT

                                          GAVD_AVDTP_ERROR_BAD_LENGTH
                                          GAVD_AVDTP_ERROR_BAD_ACP_SEID
                                          GAVD_AVDTP_ERROR_SEP_IN_USE
                                          GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
                                          GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
                                          GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
                                          GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
                                          GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
                                          GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
                                          GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
                                                  _FORMAT
                                          GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
                                          GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
                                          GAVD_AVDTP_ERROR_BAD_CP_FORMAT
                                          GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
                                          GAVD_AVDTP_ERROR_UNSUPPORTED
                                                  _CONFIGURATION
                                          GAVD_AVDTP_ERROR_BAD_STATE
                                          GAVD_AVDTP_ERROR_TIMEOUT

    FirstFailingLSEID   Identifier of the first local stream endpoint that failed to start.

etGAVD_Suspend_Indication

Indicates that a Suspend Stream command has been received from a remote Stream
Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int   LSEID;
} GAVD_Suspend_Indication_Data_t;
```

Event Parameters:

    LSEID   Identifier of the local stream endpoint.

etGAVD_Suspend_Confirmation

Confirms that a Suspend Stream command sent to a remote Stream Endpoint was
received.

Return Structure:

```
typedef struct
{
   unsigned int   LSEID;
   unsigned int   ErrorCode;
   unsigned int   FirstFailingLSEID;
} GAVD_Suspend_Confirmation_Data_t;
```

Event Parameters:

LSEID   Identifier of the local stream endpoint.

ErrorCode   Error response code of the start stream endpoint confirmation.
                            This value is one of the following values:

> GAVD_AVDTP_ERROR_SUCCESS
> GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
> GAVD_AVDTP_ERROR_BAD_LENGTH
> GAVD_AVDTP_ERROR_BAD_ACP_SEID
> GAVD_AVDTP_ERROR_SEP_IN_USE
> GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
> GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
> GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
> GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
> GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
> GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
> GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
>   _FORMAT
> GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
> GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
> GAVD_AVDTP_ERROR_BAD_CP_FORMAT
> GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
> GAVD_AVDTP_ERROR_UNSUPPORTED
>   _CONFIGURATION
> GAVD_AVDTP_ERROR_BAD_STATE
> GAVD_AVDTP_ERROR_TIMEOUT

FirstFailingLSEID   Identifier of the first local stream endpoint that failed to
                            suspend.

etGAVD_Reconfigure_Indication

Indicates that a Reconfigure Stream command has been received from a remote Stream
Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int                      LSEID;
  unsigned int                      NumberServiceCapabilities;
  GAVD_Service_Capabilities_Info_t   *ServiceCapabilities;
} GAVD_Reconfigure_Indication_Data_t;
```

Event Parameters:

LSEID   Identifier of the local stream endpoint.

NumberServiceCapabilities   Number of service capabilities that are contained in the array
                            pointed to by the ServiceCapabilities parameter.

ServiceCapabilities   Pointer to an array of service capabilities. This is defined by
                            the following structure:
```
typedef struct
{
  GAVD_Service_Category_t ServiceCategory;
  union
  {
    GAVD_Recovery_Info_Element_Data_t
            GAVD_Recovery_Info_Element_Data;
    GAVD_Content_Protection_Info_Element_Data_t
            GAVD_Content_Protection_Info_Element_Data;
    GAVD_Header_Compression_Info_Element_Data_t
            GAVD_Header_Compression_Info_Element_Data;
    GAVD_Multiplexing_Info_Element_Data_t
            GAVD_Multiplexing_Info_Element_Data;
    GAVD_Media_Codec_Info_Element_Data_t
            GAVD_Media_Codec_Info_Element_Data;
    GAVD_Raw_Info_Element_Data_t
            GAVD_Raw_Info_Element_Data;
  } InfoElement;
} GAVD_Service_Capabilities_Info_t;
```

etGAVD_Reconfigure_Confirmation

Confirms the completion of a Reconfigure Stream command that was sent to a remote
Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int                LSEID;
  unsigned int                ErrorCode;
  GAVD_Service_Category_t     FirstFailingServiceCategory;
} GAVD_Reconfigure_Confirmation_Data_t;
```

Event Parameters:

LSEID	Identifier of the local stream endpoint.
ErrorCode	Error response code of the reconfiguration confirmation. This value is one of the following values:

> GAVD_AVDTP_ERROR_SUCCESS
> GAVD_AVDTP_ERROR_BAD_HEADER_FORMAT
> GAVD_AVDTP_ERROR_BAD_LENGTH
> GAVD_AVDTP_ERROR_BAD_ACP_SEID
> GAVD_AVDTP_ERROR_SEP_IN_USE
> GAVD_AVDTP_ERROR_SEP_NOT_IN_USE
> GAVD_AVDTP_ERROR_BAD_SERV_CATEGORY
> GAVD_AVDTP_ERROR_BAD_PAYLOAD_FORMAT
> GAVD_AVDTP_ERROR_NOT_SUPPORTED_COMMAND
> GAVD_AVDTP_ERROR_INVALID_CAPABILITIES
> GAVD_AVDTP_ERROR_BAD_RECOVERY_TYPE
> GAVD_AVDTP_ERROR_BAD_MEDIA_TRANSPORT
>   _FORMAT
> GAVD_AVDTP_ERROR_BAD_RECOVERY_FORMAT
> GAVD_AVDTP_ERROR_BAD_ROHC_FORMAT
> GAVD_AVDTP_ERROR_BAD_CP_FORMAT
> GAVD_AVDTP_ERROR_BAD_MULTIPLEXING_FORMAT
> GAVD_AVDTP_ERROR_UNSUPPORTED
>   _CONFIGURATION
> GAVD_AVDTP_ERROR_BAD_STATE
> GAVD_AVDTP_ERROR_TIMEOUT

FirstFailingServiceCategory This represents the first service category that was requested to be reconfigured that was unable to be reconfigured (scNone if reconfiguration was successful). Defined by the following enumeration:

```
typedef enum
{
  scNone,
  scMediaTransport,
  scReporting,
  scRecovery,
  scContentProtection,
  scHeaderCompression,
  scMultiplexing,
  scMediaCodec,
  scUnknown
```

} GAVD_Service_Category_t;

etGAVD_Security_Control_Indication

Indicates that a Security Control command has been received from a remote Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int    LSEID;
  unsigned int    SecurityDataLength;
  Byte_t          *SecurityData;
} GAVD_Security_Control_Indication_Data_t;
```

Event Parameters:

LSEID	Identifier of the local stream endpoint.
SecurityDataLength	Length of the security data. This value specifies the size (in bytes) of the Security Data information that is pointed to by the SecurityData member.
SecurityData	Pointer to the security control data.

etGAVD_Security_Control_Confirmation

Confirms the completion of a Security Control command that was sent to a remote Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int    LSEID;
  unsigned int    ErrorCode;
  unsigned int    SecurityDataLength;
  Byte_t          *SecurityData;
} GAVD_Security_Control_Confirmation_Data_t;
```

Event Parameters:

LSEID	Identifier of the local stream endpoint.
ErrorCode	Status of the stream suspend attempt.
SecurityDataLength	Length of the security data. This value specifies the size (in bytes) of the Security Control Data information that is pointed to by the SecurityData member.
SecurityData	Pointer to the security control data.

etGAVD_Abort_Indication

Indicates that an Abort Stream command has been received from a remote Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int   LSEID;
} GAVD_Abort_Indication_Data_t;
```

Event Parameters:

LSEID   Identifier of the local stream endpoint.

etGAVD_Abort_Confirmation

Confirms the completion of an Abort Stream command that was sent to a remote Stream Endpoint Manager.

Return Structure:

```
typedef struct
{
  unsigned int   LSEID;
} GAVD_Abort_Confirmation_Data_t;
```

Event Parameters:

LSEID   Identifier of the local stream endpoint.

etGAVD_Data_Indication

Indicates that data has been received from a remote Endpoint.

Return Structure:

```
typedef struct
{
  unsigned int   LSEID;
  DWord_t        TimeStamp;
  Byte_t         PayloadType;
  Boolean_t      Marker;
  Word_t         SequenceNumber;
  unsigned int   DataLength;
  Byte_t         *DataBuffer;
} GAVD_Data_Indication_Data_t;
```

Event Parameters:

LSEID   Identifier of the local stream endpoint.

TimeStamp   Holds time stamp information.

PayloadType   Indicates the content type of the data.

Marker	Specifies whether the marker bit (used by certain codec payload formats) was specified or not.
SequenceNumber	Specifies the sequence number of the received data.
DataLength	Specifies the length of the received data. This value represents the size (in bytes) of the data that is pointed to by the DataBuffer member.
DataBuffer	Points to the received data.

etGAVD_Sender_Report_Data_Indication

Indicates that Sender Report data has been received from a remote Endpoint.

Return Structure:

```
typedef struct
{
  unsigned int            LSEID;
  GAVD_Sender_Info_t      *SenderInfo;
  unsigned int            NumberReportBlocks;
  GAVD_Report_Block_t     *ReportBlocks;
  unsigned int            ExtensionDataLength;
  DWord_t                 *ExtensionData;
} GAVD_Sender_Report_Data_Indication_Data_t;
```

Event Parameters:

LSEID	Identifier of the local stream endpoint.
SenderInfo	Contains the RTCP Sender Information. This is defined by the following structure: ```typedef struct\n{\n DWord_t NTPTimeStampMSW;\n DWord_t NTPTimeStampLSW;\n DWord_t RTPTimeStamp;\n DWord_t PacketCount;\n DWord_t OctetCount;\n} GAVD_Sender_Info_t;```
NumberReportBlocks	Indicates the number of report blocks pointed to by the ReportBlocks parameter.
ReportBlocks	Points to the report blocks. This is defined by the following structure: ```typedef struct\n{\n DWord_t SourceID;\n Byte_t FractionLost;\n DWord_t TotalPacketsLost;\n Word_t HighestSeqNumReceivedCycleCount;\n Word_t HighestSeqNumReceived;```

```
                          DWord_t      IntervalJitter;
                          DWord_t      LastReport;
                          DWord_t      DelaySinceLastReport;
                        } GAVD_Report_Block_t;
```

ExtensionDataLength   Indicates how many 32-bit words are in the extension data.

ExtensionData   Points to the extension data. This member points to an array of
                         32-bit words. The number of 32-bit words pointed to by this
                         member is given by the ExtensionDataLength member.

etGAVD_Receiver_Report_Data_Indication

Indicates that Receiver Report data has been received from a remote Endpoint.

Return Structure:

```
typedef struct
{
  unsigned int              LSEID;
  unsigned int              NumberReportBlocks;
  GAVD_Report_Block_t    *ReportBlocks;
  unsigned int              ExtensionDataLength;
  DWord_t                  *ExtensionData;
} GAVD_Receiver_Report_Data_Indication_Data_t;
```

Event Parameters:

LSEID   Identifier of the local stream endpoint.

NumberReportBlocks   Indicates the number of report blocks pointed to by the
                         ReportBlocks parameter.

ReportBlocks   Points to the report blocks. This is defined by the following
                         structure:
```
                        typedef struct
                        {
                          DWord_t      SourceID;
                          Byte_t       FractionLost;
                          DWord_t      TotalPacketsLost;
                          Word_t       HighestSeqNumReceivedCycleCount;
                          Word_t       HighestSeqNumReceived;
                          DWord_t      IntervalJitter;
                          DWord_t      LastReport;
                          DWord_t      DelaySinceLastReport;
                        } GAVD_Report_Block_t;
```

ExtensionDataLength   Indicates how many 32-bit words are in the extension data.

ExtensionData   Points to the extension data. This member points to an array of
                         32-bit words. The number of 32-bit words pointed to by this
                         member is given by the ExtensionDataLength member.

etGAVD_SDES_Report_Data_Indication

Indicates that SDES Report data has been received from a remote Endpoint.

Return Structure:

```
typedef struct
{
  unsigned int          LSEID;
  unsigned int          NumberSDESChunks;
  GAVD_SDES_Chunk_t   *SDESChunks;
} GAVD_SDES_Report_Data_Indication_Data_t;
```

Event Parameters:

LSEID   Identifier of the local stream endpoint.

NumberSDESChunks   Indicates the number of SDES chucks pointed to by the SDESChunks parameter.

SDESChunks   Points to the SDES chunks. This is defined by the following structure:

```
typedef struct
{
  DWord_t                SourceID;
  unsigned int           NumberSDESItems;
  GAVD_SDES_Item_t      *SDESItems;
} GAVD_SDES_Chunk_t;
```

where, SDESItems is defined by the following structure:

```
typedef struct
{
  Byte_t    ItemDescriptor;
  Byte_t    ItemLength;
  Byte_t   *ItemData;
} GAVD_SDES_Item_t;
```

etGAVD_Recovery_Data_Indication

Indicates that Recovery data has been received from a remote Endpoint.

Return Structure:

```
typedef struct
{
  unsigned int          LSEID;
  DWord_t               TimeStamp;
  Byte_t                PayloadType;
  Boolean_t             Marker;
  Word_t                SequenceNumber;
  GAVD_FEC_Block_t     *FECBlock;
} GAVD_Recovery_Data_Indication_Data_t;
```

Event Parameters:

LSEID   Identifier of the local stream endpoint.

TimeStamp	Holds time stamp information.
PayloadType	Indicates the content type of the data.
Marker	Specifies the marker bit used by certain codec payload formats.
SequenceNumber	Specifies the sequence number of the received data.
FECBlock	Points to the structure with the information to send. This is defined by the following structure:

```
typedef struct
{
  Word_t        SequenceNumberBase;
  Word_t        LengthRecovery;
  Byte_t        PayloadTypeRecovery;
  DWord_t       Mask;
  DWord_t       TimeStampRecovery;
  unsigned int  FECDataLength;
  Byte_t        *FECData;
} GAVD_FEC_Block_t;
```

etGAVD_Channel_Empty_Indication

Dispatched by a GAVD/AVDTP entity to the local application when a Stream End Point no longer has any data queued to be sent on the Data Channel. This event is only dispatched when the Data Write function returned the error code BTPS_ERROR_INSUFFICIENT_RESOURCES.

Return Structure:

```
typedef struct
{
  unsigned int            LSEID;
} GAVD_Channel_Empty_Indication_Data_t;
```

Event Parameters:

LSEID	Identifier of the local stream endpoint.

3.   File Distributions

The header files that are distributed with the Bluetooth GAVD Profile Library are listed in the table below.

File	Contents/Description
GAVDAPI.h	Bluetooth GAVD Profile API definitions
SS1BTGAV.h	Bluetooth GAVD Profile Include file
GAVDType.h	Bluetooth GAVD Profile type definitions