



Phonebook Access Profile (PBAP)

Application Programming Interface Reference Manual

Profile Version: 1.3.2

Release: 5.1.0
August 3, 2021



Bluetooth and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc., USA and licensed to Stonestreet One, LLC. Bluetopia®, Stonestreet One™, and the Stonestreet One logo are registered trademarks of Stonestreet One, LLC, Louisville, Kentucky, USA. All other trademarks are property of their respective owners.
Copyright © 2000-2014 by Stonestreet One, LLC. All rights reserved.

Table of Contents

1. INTRODUCTION.....	3
1.1 Scope	3
1.2 Applicable Documents	4
1.3 Acronyms and Abbreviations	5
2. PBAP PROGRAMMING INTERFACE	7
2.1 PBAP Commands	7
PBAP_Open_Server_Port.....	8
PBAP_Open_Server_Port_Goep2	10
PBAP_Close_Server_Port	11
PBAP_Register_Server_SDP_Record.....	11
PBAP_Register_Server_SDP_Record_Goep2	13
PBAP_Register_Client_SDP_Record.....	14
PBAP_Open_Remote_Server_Port	15
PBAP_Open_Remote_Server_Port_Goep2.....	16
PBAP_Open_Request_Response.....	17
PBAP_Close_Connection	18
PBAP_Disconnect_Request.....	18
PBAP_Abort_Request	19
PBAP_Pull_Phonebook_Request	20
PBAP_Pull_Phonebook_Response.....	23
PBAP_Set_Phonebook_Request.....	25
PBAP_Set_Phonebook_Response	26
PBAP_Pull_vCard_Listing_Request	27
PBAP_Pull_vCard_Listing_Response.....	29
PBAP_Pull_vCard_Entry_Request.....	31
PBAP_Pull_vCard_Entry_Response	33
PBAP_Get_Server_Mode	35
PBAP_Set_Server_Mode.....	35
PBAP_EnableSRM.....	36
PBAP_EnableSRMP.....	37
PBAP_DisableSRM.....	38
PBAP_Set_Folder_Version	38
PBAP_Increase_Folder_Version	39
PBAP_Set_Database_Identifier.....	40
2.2 PBAP Event Callback Prototypes	41
PBAP_Event_Callback_t.....	41
2.3 PBAP Events	42
etPBAP_Open_Port_Indication	43
etPBAP_Open_Port_Confirmation.....	44
etPBAP_Open_Port_Request_Indication	44
etPBAP_Close_Port_Indication.....	45
etPBAP_Abort_Indication	45

etPBAP_Abort_Confirmation.....	46
etPBAP_Pull_Phonebook_Indication.....	46
etPBAP_Pull_Phonebook_Confirmation.....	48
etPBAP_Set_Phonebook_Indication	49
etPBAP_Set_Phonebook_Confirmation.....	50
etPBAP_Pull_vCard_Listing_Indication.....	51
etPBAP_Pull_vCard_Listing_Confirmation.....	52
etPBAP_Pull_vCard_Entry_Indication	53
etPBAP_Pull_vCard_Entry_Confirmation	55
3. FILE DISTRIBUTIONS.....	57

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 Phone Book Access Profile, PBAP, provided by Bluetopia. Chapter 2 contains a description of the programming interfaces for this profile. And, Chapter 3 contains the header file name list for the Bluetooth Phone Book Access Profile library.

1.1 Scope

This reference manual provides information on the PBAP 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

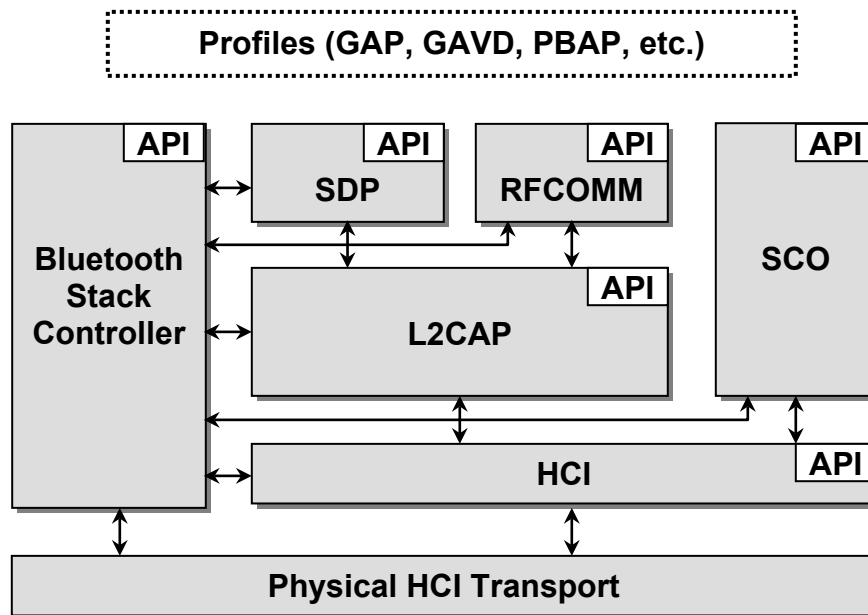


Figure 1-1 The Stonestreet One Bluetooth Protocol Stack

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. *Specification of the Bluetooth System, Volume 0, Master Table of Contents & Compliance Requirements*, version 2.1+EDR, July 26, 2007.
7. *Specification of the Bluetooth System, Volume 1, Architecture and Terminology Overview*, version 2.1+EDR, July 26, 2007.
8. *Specification of the Bluetooth System, Volume 2, Core System Package [Controller Volume]*, version 2.1+EDR, July 26, 2007.
9. *Specification of the Bluetooth System, Volume 3, Core System Package [Host Volume]*, version 2.1+EDR, July 26, 2007.
10. *Specification of the Bluetooth System, Volume 4, Host Controller Interface [Transport Layer]*, version 2.1+EDR, July 26, 2007.
11. *Specification of the Bluetooth System, Bluetooth Core Specification Addendum 1*, June 26, 2008.
12. *Specification of the Bluetooth System, Volume 0, Master Table of Contents & Compliance Requirements*, version 3.0+HS, April 21, 2009.
13. *Specification of the Bluetooth System, Volume 1, Architecture and Terminology Overview*, version 3.0+HS, April 21, 2009.
14. *Specification of the Bluetooth System, Volume 2, Core System Package [Controller Volume]*, version 3.0+HS, April 21, 2009.
15. *Specification of the Bluetooth System, Volume 3, Core System Package [Host Volume]*, version 3.0+HS, April 21, 2009.
16. *Specification of the Bluetooth System, Volume 4, Host Controller Interface [Transport Layer]*, version 3.0+HS, April 21, 2009.
17. *Specification of the Bluetooth System, Volume 5, Core System Package [AMP Controller Volume]*, version 3.0+HS, April 21, 2009.

18. *Specification of the Bluetooth System, Volume 0, Master Table of Contents & Compliance Requirements*, version 4.0, June 30, 2010.
19. *Specification of the Bluetooth System, Volume 1, Architecture and Terminology Overview*, version 4.0, June 30, 2010.
20. *Specification of the Bluetooth System, Volume 2, Core System Package [BR/EDR Controller Volume]*, version 4.0, June 30, 2010.
21. *Specification of the Bluetooth System, Volume 3, Core System Package [Host Volume]*, version 4.0, June 30, 2010.
22. *Specification of the Bluetooth System, Volume 4, Host Controller Interface [Transport Layer]*, version 4.0, June 30, 2010.
23. *Specification of the Bluetooth System, Volume 5, Core System Package [AMP Controller Volume]*, version 4.0, June 30, 2010.
24. *Specification of the Bluetooth System, Volume 6, Core System Package [Low Energy Controller Volume]*, version 4.0, June 30, 2010.
25. *Bluetooth Phone Book Access Profile*, version 1.0, April 27, 2006
26. *IrDA IrOBEX Specification*
27. *Bluetooth Assigned Numbers*, version 1.1, February 22, 2001.
28. *Bluetopia® Protocol Stack, Application Programming Interface Reference Manual*, version 4.0.1, January 10, 2013.

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
BD_ADDR	Bluetooth Device Address
BR	Basic Rate
BT	Bluetooth
EDR	Enhanced Data Rate
HS	High Speed
LE	Low Energy
LSB	Least Significant Bit

Term	Meaning
MSB	Most Significant Bit
PBAP	Phone Book Access Profile
SDP	Service Discovery Protocol
SPP	Serial Port Protocol
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

2. PBAP Programming Interface

The Phone Book Access Profile, PBAP, programming interface defines the protocols and procedures to be used to implement PBAP capabilities for both Server and Client services. The PBAP commands are listed in section 2.1, the event callback prototype is described in section 2.2, and the PBAP events are itemized in section 2.3.

2.1 PBAP Commands

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

Server Commands	
Function	Description
PBAP_Open_Server_Port PBAP_Open_Server_Port_Goep2	Opens a local PBAP server port.
PBAP_Close_Server_Port	Closes a currently open PBAP server port.
PBAP_Register_Server_SDP_Record PBAP_Register_Server_SDP_Record_Goep2	Registers a PBAP service record to the local SDP database for a currently open PBAP server port.
PBAP_Un_Register_SDP_Record	Remove a PBAP Service Record from the SDP Database
PBAP_Register_Client_SDP_Record	Registers a PBAP server record to the local SDP database for client connections.
PBAP_Open_Remote_Server_Port PBAP_Open_Remote_Server_Port_Goep2	Makes a connection to a remote PBAP server.
PBAP_Open_Request_Response	Responds to an individual request to connect to a local PBAP Server.
PBAP_Close_Connection	Closes a currently ongoing PBAP connection.
PBAP_Disconnect_Request	Issues an OBEX Disconnect request to the remote PBAP server.
PBAP_Abort_Request	Sends an Abort Request to the remote PBAP Server.
PBAP_Pull_Phonebook_Request PBAP12_Pull_Phonebook_Request	Retrieves the entire phonebook object (list of vCards) from the server.
PBAP_Pull_Phonebook_Response	Responds to a Pull Phonebook Request with the vCard list of entries to the issuing

	client.
PBAP_Set_Phonebook_Request	Sets the current folder in the virtual folder architecture on the server.
PBAP_Set_Phonebook_Response	Status response for a Set Phonebook Request to the issuing client.
PBAP_Pull_vCard_Listing_Request PBAP12_Pull_vCard_Listing_Request	Retrieves the phonebook listing from the server.
PBAP_Pull_vCard_Listing_Response	Responds to a Pull vCard Listing to the issuing client in XML format.
PBAP_Pull_vCard_Entry_Request	Retrieves a specific vCard Entry from the server.
PBAP_Pull_vCard_Entry_Response	Responds to a Pull vCard Entry with the specific vCard entry requested to the issuing client.
PBAP_Get_Server_Mode	Queries the current PBAP Server Mode.
PBAP_Set_Server_Mode	Changes the current PBAP Server Mode.
PBAP_EnableSRM	Enable single-response mode.
PBAP_EnableSRMP	Enable single-response mode parameters.
PBAP_DisableSRM	Disable single-response mode.
PBAP_DisableSRMP	Disable single-response mode parameters.
PBAP_Set_Folder_Version	Sets the initial primary or secondary folder version maintained by the application.
PBAP_Increase_Folder_Version	Increases the folder version (primary or secondary).
PBAP_Set_Database_Identifier	Set the initial database identifier maintained by the application.

PBAP_Open_Server_Port

The following function is responsible for opening a local PBAP server using a legacy RFCOMM port. Prefer PBAP_Open_Server_Port_Goep2().

Prototype:

```
int BTPSAPI PBAP_Open_Server_Port(unsigned int BluetoothStackID, unsigned int
    ServerPort, Byte_t SupportedRepositories, PBAP_Event_Callback_t EventCallback,
    unsigned long CallbackParameter);
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
ServerPort	Server Port number to use. The selected port value must fall in the range defined by the following constants: PBAP_PORT_NUMBER_MINIMUM PBAP_PORT_NUMBER_MAXIMUM
SupportedRepositories	A bit mask used to indicate the types of Object Stores that are supported by the local PBAP Server. The following bits are defined: PBAP_SUPPORTED_REPOSITORIES_LOCAL_PHONEBOOK PBAP_SUPPORTED_REPOSITORIES_SIM_CARD
EventCallback	Function to call when events occur on this server.
CallbackParameter	A user-defined parameter (e.g., a tag value) that will be passed back to the user in the callback function.

Return:

Positive, non-zero if successful. If this function is successful, the return value will represent the PBAP ID that can be passed to all other functions that reference the opened PBAP Server.

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

```
BTPBAP_ERROR_INVALID_PARAMETER
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INSUFFICIENT_RESOURCES
```

Possible Events:

```
etPBAP_Open_Port_Indication
etPBAP_Open_Port_Request_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.

PBAP_Open_Server_Port_Goep2

The following function is responsible for opening a local PBAP server using a L2CAP PSM.

Prototype:

```
int BTPSAPI PBAP_Open_Server_Port_Goep2(unsigned int BluetoothStackID, unsigned int
    ServerPort, Word_t L2CAPPSM, Byte_t SupportedRepositories, PBAP_Event_Callback_t
    EventCallback, unsigned long CallbackParameter);
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
ServerPort	Server Port number to use. The selected port value must fall in the range defined by the following constants: PBAP_PORT_NUMBER_MINIMUM PBAP_PORT_NUMBER_MAXIMUM
L2CAPPSM	L2CAP PSM to use for the GOEP 2.0 server.
SupportedRepositories	A bit mask used to indicate the types of Object Stores that are supported by the local PBAP Server. The following bits are defined: PBAP_SUPPORTED_REPOSITORIES_LOCAL_PHONEBOOK PBAP_SUPPORTED_REPOSITORIES_SIM_CARD
EventCallback	Function to call when events occur on this server.
CallbackParameter	A user-defined parameter (e.g., a tag value) that will be passed back to the user in the callback function.

Return:

Positive, non-zero if successful. If this function is successful, the return value will represent the PBAP ID that can be passed to all other functions that reference the opened PBAP Server.

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

```

BTPBAP_ERROR_INVALID_PARAMETER
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INSUFFICIENT_RESOURCES

```

Possible Events:

```

etPBAP_Open_Port_Indication
etPBAP_Open_Port_Request_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.

PBAP_Close_Server_Port

This function is responsible for closing a server opened by a successful call to the PBAP_Open_Server_Port () function. Note, this function does NOT delete any Service Record Handles (i.e., added via a PBAP_Register_Server_SDP_Record () function call).

Prototype:

int BTPSAPI **PBAP_Close_Server_Port**(unsigned int BluetoothStackID, unsigned int PBAPID)

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	The unique identifier of the PBAP Server to be closed. This is the value that was returned from the PBAP_Open_Server_Port () function.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_PARAMETER
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INVALID_OPERATION

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.

PBAP_Register_Server_SDP_Record

This function adds a PBAP Server Service Record to the SDP Database based on the type of PBAP Server created with a previous call to the PBAP_Open_Server_Port() function. This function will create an appropriate SDP record for each Server Type and uses parameters passed to the previous call to populate relevant information.

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 `PBAP_Un_Register_SDP_Record()` to `SDP_Delete_Service_Record()`, and is defined as follows:

```
PBAP_Un_Register_SDP_Record (__BluetoothStackID, __PBAPID,  
    __SDPRecordHandle) (SDP_Delete_Service_Record(__BluetoothStackID,  
    __SDPRecordHandle))
```

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.

Prototype:

```
int BTPSAPI PBAP_Register_Server_SDP_Record(unsigned int BluetoothStackID,  
    unsigned int PBAPID, char *ServiceName, DWORD_t *SDPServiceRecordHandle)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to <code>BSC_Initialize</code> .
PBAPID	The unique identifier that was returned from the call to the <code>PBAP_Open_Server_Port()</code> function.
ServiceName	The Service Name to be associated with the Service Name SDP Attribute.
SDPServiceRecordHandle	Returned handle to the SDP Database entry that may be used to remove the entry at a later time. This value is valid if the return value is zero (successful), otherwise it is undefined.

Return:

Zero if successful.

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

```
BTPBAP_ERROR_ACTION_NOT_ALLOWED  
BTPBAP_ERROR_INVALID_PARAMETER  
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID  
BTPBAP_ERROR_NOT_INITIALIZED  
BTPBAP_ERROR_INVALID_OPERATION
```

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.

PBAP_Register_Server_SDP_Record_Goep2

This function adds a PBAP Server Service Record to the SDP Database based on the type of PBAP Server created with a previous call to the PBAP_Open_Server_Port_Goep2() function. This function will create an appropriate SDP record for each Server Type and uses parameters passed to the previous call to populate relevant information. Do not use this version if PBAP_Open_Server_Port() was called instead.

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 PBAP_Un_Register_SDP_Record () to SDP_Delete_Service_Record(), and is defined as follows:

```
PBAP_Un_Register_SDP_Record ( __BluetoothStackID, __PBAPID,  
__SDPRecordHandle) (SDP_Delete_Service_Record(__BluetoothStackID,  
__SDPRecordHandle))
```

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.

Prototype:

```
int BTPSAPI PBAP_Register_Server_SDP_Record(unsigned int BluetoothStackID,  
unsigned int PBAPID, char *ServiceName, Word_t L2CAPPSM, DWord_t  
*SDPServiceRecordHandle)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	The unique identifier that was returned from the call to the PBAP_Open_Server_Port() function.
ServiceName	The Service Name to be associated with the Service Name SDP Attribute.
L2CAPPSM	The L2CAPPSM used when registering the server.
SDPServiceRecordHandle	Returned handle to the SDP Database entry that may be used to remove the entry at a later time. This value is valid if the return value is zero (successful), otherwise it is undefined.

Return:

Zero if successful.

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

```
BTPBAP_ERROR_ACTION_NOT_ALLOWED  
BTPBAP_ERROR_INVALID_PARAMETER  
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID  
BTPBAP_ERROR_NOT_INITIALIZED  
BTPBAP_ERROR_INVALID_OPERATION
```

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.

PBAP_Register_Client_SDP_Record

Adds a PBAP Client (PCE) Service Record to the SDP database for client connections.

Prototype:

```
int BTPSAPI PBAP_Register_Client_SDP_Record(unsigned int BluetoothStackID, char
    *ServiceName, DWord_t *SDPServiceRecordHandle)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
ServiceName ²	Pointer to ASCII, null terminated, string containing the Service Name to include in the SDP Service Record.
SDPServiceRecordHandle ³	Pointer to Dword_t which will receive the SDP Service Record Handle if this function successfully creates an SDP Service Record.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_PARAMETER
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID

Notes:

1. The BluetoothStackID parameter is not included in versions of Bluetopia that have been optimized to only control a single Bluetooth device, such as some embedded versions of Bluetopia. Please refer to the appropriate header file to determine if this parameter is part of the function call or not.
2. 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.
3. If the Service Record is created successfully and a handle is returned in SDPServiceRecordHandle, it will remain in the SDP Record Database until it is deleted using a call to the PBAP_Un_Register_SDP_Record() macro. See the documentation for PBAP_Register_Server_SDP_Record() above for a description of this MACRO.

PBAP_Open_Remote_Server_Port

The following function is responsible for opening a connection from the local device to a remote PBAP server.

Prototype:

```
int BTPSAPI PBAP_Open_Remote_Server_Port(unsigned int BluetoothStackID,  
    BD_ADDR_t BD_ADDR, unsigned int ServerPort, PBAP_Event_Callback_t EventCallback,  
    unsigned long CallbackParameter)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
BD_ADDR	Board Address of the Remote PBAP Server Device with which to connect.
ServerPort	The Server Port number on the remote device with which to connect. Valid values are in the following range: PBAP_PORT_NUMBER_MINIMUM PBAP_PORT_NUMBER_MAXIMUM
EventCallback	Function to call when events occur on this server.
CallbackParameter	A user-defined parameter (e.g., a tag value) that will be passed back to the user in the callback function.

Return:

Positive, non-zero if successful. If this function is successful, the return value will represent the Phonebook Access ID or handle, PBAPID, that can be passed to all other functions that reference the opened PBAP connection to a remote Sender.

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

BTPBAP_ERROR_INVALID_PARAMETER
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INSUFFICIENT_RESOURCES

Possible Events:

etPBAP_Open_Port_Confirmation

etPBAP_Close_Port_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.

PBAP_Open_Remote_Server_Port_Goep2

The following function is responsible for opening a connection from the local device to a remote PBAP server that is using L2CAP.

Prototype:

```
int BTPSAPI PBAP_Open_Remote_Server_Port_Goep2(unsigned int BluetoothStackID,
        BD_ADDR_t BD_ADDR, unsigned int ServerPort, Boolean_t IsLegacy,
        PBAP_Event_Callback_t EventCallback, unsigned long CallbackParameter)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
BD_ADDR	Board Address of the Remote PBAP Server Device with which to connect.
ServerPort	The Server Port number on the remote device with which to connect. Valid values are in the following range: PBAP_PORT_NUMBER_MINIMUM PBAP_PORT_NUMBER_MAXIMUM
IsLegacy	TRUE if a connection to legacy RFCOMM PBAP server is being established, FALSE if a connection to an L2CAP PBAP server is being established.
EventCallback	Function to call when events occur on this server.
CallbackParameter	A user-defined parameter (e.g., a tag value) that will be passed back to the user in the callback function.

Return:

Positive, non-zero if successful. If this function is successful, the return value will represent the Phonebook Access ID or handle, PBAPID, that can be passed to all other functions that reference the opened PBAP connection to a remote Sender.

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

```
BTPBAP_ERROR_INVALID_PARAMETER
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INSUFFICIENT_RESOURCES
```

Possible Events:

etPBAP_Open_Port_Confirmation

etPBAP_Close_Port_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.

PBAP_Open_Request_Response

This function is responsible for responding to an individual request to connect to a local PBAP Server. This function is only used if the server mode is set to manual. See the PBAP_Get_Server_Mode() and PBAP_Set_Server_Mode() functions for more information.

Prototype:

```
int BTPSAPI PBAP_Open_Request_Response(unsigned int BluetoothStackID,  
    unsigned int PBAPID, Boolean_t AcceptConnection)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	The unique identifier of the PBAP for which a connection request was received.
AcceptConnection	The Boolean value indicating whether or not to accept the pending connection request.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_PARAMETER
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED

Possible Events:

etPBAP_Close_Port_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.

PBAP_Close_Connection

This function is used to close a currently ongoing PBAP connection. This function is capable of closing connections on an opened PBAP Server without closing the server.

This function may close connections opened using calls to the PBAP_Open_Remote_Server_Port() function. This function may also be used to close connections established through calls to the PBAP_Open_Server_Port() function. In the later case, all client connections will be terminated but the server will remain registered.

Prototype:

```
int BTPSAPI PBAP_Close_Connection(unsigned int BluetoothStackID,  
unsigned int PBAPID)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique Identifier of the PBAP connection to be closed. This is the value that was returned from a call to either the PBAP_Open_Server_Port() or PBAP_Open_Remote_Server_Port() functions.

Return:

Zero if successful.

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

```
BTPBAP_ERROR_INVALID_PARAMETER  
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID  
BTPBAP_ERROR_NOT_INITIALIZED
```

Notes:

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

PBAP_Disconnect_Request

This function is used to send an OBEX Disconnect Request to the remote PBAP Server.

Issuing an OBEX Disconnect Request is not always required when closing a connection (PBAP_Close_Connection() or PBAP_Close_Server_Port() can be used in place of or following this function). Calling this function results in the OBEX Disconnect Request being issued and may be required for strict OBEX compliance.

Prototype:

int BTPSAPI **PBAP_Disconnect_Request**(unsigned int BluetoothStackID, unsigned int PBAPID)

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique Identifier of the PBAP Client making this call.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_PARAMETER
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_ACTION_NOT_ALLOWED
BTPBAP_ERROR_REQUEST_ALREADY_OUTSTANDING

Possible Events:

etPBAP_Close_Port_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.

PBAP_Abort_Request

This function is used to send an Abort Request to the remote PBAP Server.

Upon the reception of the Abort Confirmation Event it may be assumed that the currently ongoing transaction has been successfully aborted and new requests may be submitted.

Prototype:

int BTPSAPI **PBAP_Abort_Request**(unsigned int BluetoothStackID,
unsigned int PBAPID)

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique Identifier of the PBAP Client making this call.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_PARAMETER
 BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
 BTPBAP_ERROR_NOT_INITIALIZED
 BTPBAP_ERROR_ACTION_NOT_ALLOWED
 BTPBAP_ERROR_REQUEST_ALREADY_OUTSTANDING

Possible Events:

etPBAP_Abort_Confirmation
 etPBAP_Close_Port_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.

PBAP_Pull_Phonebook_Request

This client side function is used to retrieve an entire Phonebook object or the number of current Phonebook entries from a remote PBAP server. If an entire or partial Phonebook is requested, *MaxListCount* > 0, then they are returned in vCard format.

There are five phonebooks typically located in a PBAP for each repository that exists on the server:

- **pb.vcf** (main phonebook)
- **ich.vcf** (incoming call history)
- **och.vcf** (outgoing call history)
- **mch.vcf** (missed call history)
- **cch.vcf** (complete call history)

When only the count of indexes or entries in the phonebook is requested, *MaxListCount* is set to 0.

Prototype:

```
int BTPSAPI PBAP_Pull_Phonebook_Request(unsigned int BluetoothStackID,
    unsigned int PBAPID, char *ObjectName, DWord_t FilterLow, DWord_t FilterHigh,
    PBAP_Format_t Format, Word_t MaxListCount, Word_t ListStartOffset)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to pull the phonebook.
ObjectName	Contains the absolute path in the virtual folder architecture appended with the name of the file, file extension (<i>vcf</i>)

included, of one of the phonebook objects. For example to pull the main Phonebooks in a remote server, “*telecom/pb.vcf*” or “*SIM/telecom/pb.vcf*”. The extension, .vcf, must be present in the object name. In most cases this request is issued while the server is set to the root directory and would be issued as “*telecom/pb.vcf*”. Otherwise, if using phonebook browsing then the pathname should be relative to the current folder the server is set to (see *PBAP_Set_Phonebook_Request*). NOTE: The path extensions, “.” and “..” are not valid.

FilterLow

This is the low DWORD of the filter used to indicate which attributes to include in the requested vCard object. To request all attributes available in each vCard, leave this value 0, otherwise, it can be set to one or more of the following values OR’ed:

PBAP_FILTER_VERSION
 PBAP_FILTER_FN
 PBAP_FILTER_N
 PBAP_FILTER_PHOTO
 PBAP_FILTER_BDAY
 PBAP_FILTER_ADR
 PBAP_FILTER_LABEL
 PBAP_FILTER_TEL
 PBAP_FILTER_EMAIL
 PBAP_FILTER_MAILER
 PBAP_FILTER_TZ
 PBAP_FILTER_GEO
 PBAP_FILTER_TITLE
 PBAP_FILTER_ROLE
 PBAP_FILTER_LOGO
 PBAP_FILTER_AGENT
 PBAP_FILTER_ORG
 PBAP_FILTER_NOTE
 PBAP_FILTER_REV
 PBAP_FILTER_SOUND
 PBAP_FILTER_URL
 PBAP_FILTER_UID
 PBAP_FILTER_KEY
 PBAP_FILTER_NICKNAME
 PBAP_FILTER_CATEGORIES
 PBAP_FILTER_PROID
 PBAP_FILTER_CLASS
 PBAP_FILTER_SORT_STRING
 PBAP_FILTER_X_IRMC_CALL_DATETIME

FilterHigh

This is the high DWORD of the filter. At this time there are only Proprietary bits that are vendor specific. The proprietary attribute mask *i* describes the client requirements for

proprietary vCard attributes. If the the following bit is set then bits that are above this one are defined specific to that vendor and for this to work, the client must be able to identify the server as a device that uses the same mask.

PBAP_FILTER_PROPRIETARY_FILTER_HIGH

Format This is the vCard format that is requested to be returned. This value is one of the following:

formatvCard21
formatvCard30
formatDefault

The default format is 2.1.

MaxListCount Value to indicate the maximum number of entries returned. The value ranges from 0 to 65535. A value of 65535 indicates that the number of entries returned is not limited. A value of 0 indicates that only the number of indexes (entries) in the phonebook be returned.

NOTE: In the case where this is set to 0, FilterLow, FilterHigh, Format, and ListStartOffset are ignored.

ListStartOffset Value to indicate to the server the offset of the first entry to return. A value of 0 request that the list starts from the beginning.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INVALID_PARAMETER

Possible Events:

etPBAP_PullPhoneBook_Confirmation

etPBAP_Close_Port_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.

PBAP_Pull_Phonebook_Response

This server side function responds to a Pull Phonebook Request with the vCard list of entries to the requesting remote client.

This function should be called in response to the etPBAP_PullPhoneBook_Request event. Multiple PullPhonebook responses may be required to complete a single Object transaction.

Several parameters to this function cause the inclusion of optional OBEX headers. The ObjectName parameter can be used to include an OBEX Name header in the response. The ObjectLength parameter can be used to include an OBEX Length header in the response. The Buffer and BufferSize parameters determine if the response should contain data.

Responses which contain data (BufferSize is non-zero and Buffer is a valid pointer) include a Body or End-Of-Body OBEX header. The response code determines which header to include. An OK response indicates that all data for the response has been included in the Buffer. A CONTINUE response indicates that not all data for the response was included in Buffer (or Buffer is NULL and no data is included).

NOTE: The minimum vCard attributes included are based upon the format of the vCard requested.

- Mandatory attributes for vCard 2.1 returned are VERSION ,N and TEL.
- Mandatory attributes for vCard 3.0 returned are VERSION, N, FN and TEL

Prototype:

```
int BTPSAPI PBAP_Pull_Phonebook_Response(unsigned int BluetoothStackID,  
    unsigned int PBAPID, Byte_t ResponseCode, Word_t *PhonebookSize,  
    Byte_t *NewMissedCalls, unsigned int BufferSize, Byte_t *Buffer,  
    unsigned int *AmountWritten)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to get the current Server Mode Mask.
ResponseCode	The Response Code to be associated with this response. The following values are valid for the Response Code parameter:

PBAP_OBEX_RESPONSE_CONTINUE
 PBAP_OBEX_RESPONSE_OK
 PBAP_OBEX_RESPONSE_NOT_FOUND
 PBAP_OBEX_RESPONSE_SERVICE_UNAVAILABLE
 PBAP_OBEX_RESPONSE_BAD_REQUEST
 PBAP_OBEX_RESPONSE_NOT_IMPLEMENTED
 PBAP_OBEX_RESPONSE_UNAUTHORIZED
 PBAP_OBEX_RESPONSE_PRECONDITION_FAILED
 PBAP_OBEX_RESPONSE_NOT_ACCEPTABLE
 PBAP_OBEX_RESPONSE_FORBIDDEN

PhonebookSize	This value is only filled if the <u>MaxListCount</u> on the Request is 0, incating the number of Phonebook indexes (entries) are required.
NewMissedCalls	This value is only filled if the target phonebook is the missed call history, <i>mch.vcf</i> , which indicates the number of missed calls not currently logged into <i>mch.vcf</i> .
BufferSize	Size in bytes of the data to be included in the response. This data is taken from the <u>Buffer</u> pointer. If the <u>Buffer</u> pointer is valid and this parameter is non-zero, a Body or End-Of-Body header will be included in this response.
Buffer	Pointer to the data to be included in this response. If this pointer is valid and <u>BufferSize</u> is non-zero, a Body or End-Body header will be included in this response.
AmountWritten	This parameter is a pointer to a variable that will contain the total amount of data that was successfully written in the current response packet. If this value is less than the requested <u>BufferSize</u> then only a portion of the data was sent. A subsequent call to this function will be required to send the remaining data (following receipt of the appropriate indication event).

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
 BTPBAP_ERROR_NOT_INITIALIZED
 BTPBAP_ERROR_INVALID_PARAMETER

Possible Events:

etPBAP_PullPhoneBook_Indication
 etPBAP_Close_Port_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.

PBAP_Set_Phonebook_Request

This function sets the current folder in the virtual folder architecture on the server. On initial open of the PBAP server, the default is always at the root of the virtual folder architecture.

Prototype:

```
int BTPSAPI PBAP_Set_Phonebook_Request(unsigned int BluetoothStackID, unsigned int
PBAPID, PBAP_SetPath_Option_t PathOption, char *ObjectName)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to get the current Server Mode Mask.
PathOption	Flag to indicate which direction in the virtual folder architecture to set path to.
	setRoot setDown setUp
ObjectName ²	<p>This is the child folder to set the current folder to in the virtual folder architecture. If the <u>PathOption</u> is set to <i>setRoot</i> this field is ignored.</p> <p>NOTE: The path extensions, “.” and “..” are not valid. Also on some server implementations the Object Name may be ignored when the <u>PathOption</u> is <i>setUp</i>.</p>

Return:

Zero if successful.

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

```
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INVALID_PARAMETER
```

Possible Events:

etPBAP_SetPhoneBook_Confirmation
 etPBAP_Close_Port_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.
2. The OBEX specification defined by IrDA clearly states that when a SetPath operation, upon which SetPhoneBook is based, is sent with the *setUp* flag enabled, a Name header may accompany it. This gives the effect of a setting to a new folder as, “../<foldername>”. It has been discovered that some server side implementations ignore the name header and still return a success code in this situation, giving the illusion that the new folder, phonebook, is “../<foldername>” when in fact it is “..”, i.e. the parent folder. To ensure compatibility, it is recommended if issuing a SetPhoneBook with the *setUp* option, that the ObjectName be left empty. The user can then follow it with another SetPhoneBook option with the *setDown* option with the Phonebook they wish to set to.

Example:

```
PBAP_SetPhoneBook_Request(BluetoothStackID, PBAPID, setUp, “ich”);
```

should be replaced with

```
PBAP_SetPhoneBook_Request(BluetoothStackID, PBAPID, setUp, NULL);
```

```
PBAP_SetPhoneBook_Request(BluetoothStackID, PBAPID, setDown, “ich”);
```

PBAP_Set_Phonebook_Response

This function sends the status response for a Set Phonebook Request to the requesting client.

Prototype:

```
int BTPSAPI PBAP_Set_Phonebook_Response(unsigned int BluetoothStackID, unsigned  
int PBAPID, Byte_t ResponseCode)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to get the current Server Mode Mask.
ResponseCode	The Response Code to be associated with this response. The following values are valid for the Response Code parameter:

PBAP_OBEX_RESPONSE_CONTINUE

PBAP_OBEX_RESPONSE_OK
PBAP_OBEX_RESPONSE_NOT_FOUND
PBAP_OBEX_RESPONSE_SERVICE_UNAVAILABLE
PBAP_OBEX_RESPONSE_BAD_REQUEST
PBAP_OBEX_RESPONSE_NOT_IMPLEMENTED
PBAP_OBEX_RESPONSE_UNAUTHORIZED
PBAP_OBEX_RESPONSE_PRECONDITION_FAILED
PBAP_OBEX_RESPONSE_NOT_ACCEPTABLE
PBAP_OBEX_RESPONSE_FORBIDDEN

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INVALID_PARAMETER

Possible Events:

etPBAP_SetPhoneBook_Indication
etPBAP_Close_Port_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.

PBAP_Pull_vCard_Listing_Request

This function retrieves the phonebook listing object from the server. The result is returned in XML format <x-bt/vcard-listing>. As with PBAP_Pull_Phonebook_Request, there are five phonebooks typically located in a PBAP for each repository that exists on the server corresponding to the following folders:

- **pb** (main phonebook)
- **ich** (incoming call history)
- **och** (outgoing call history)
- **mch** (missed call history)
- **cch** (complete call history)

Each entry is returned as an Entry Handle of the following form (see etPBAP_Pull_vCard_Listing_Confirmation for a full description):

```
<card handle = "0.vcf" name = "Smith;John"/>
```

Prototype:

```
int BTPSAPI PBAP_Pull_vCard_Listing_Request(unsigned int BluetoothStackID,
    unsigned int PBAPID, char *ObjectName, PBAP_ListOrder_t ListOrder,
    PBAP_SearchAttribute_t SearchAttribute, char *SearchValue, Word_t MaxListCount,
    Word_t ListStartOffset)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to get the current Server Mode Mask.
ObjectName	This is the name of the folder to retrieve the listing from. Unlike PBAP_Pull_Phonebook_Request, the <i>vcf</i> extension is not used since this is the folder name to browse.
ListOrder	Value to indicate in what order the listing is returned. It can be one of the following: <div> orderIndexed orderAlphabetical orderPhonetical orderDefault </div> The default order is Indexed.
SearchAttribute	Value to indicate to the server which vCard attribute the search must be performed on. <div> searchName searchNumber searchSound searchDefault </div> The default search attribute is Name.
SearchValue	Value string, based on attribute, to which matching vCards shall be returned in the listing.
MaxListCount	Value to indicate the maximum number of entries returned. The value ranges from 0 to 65535. A value of 65535 indicates that the number of entries returned is not limited. A value of 0 indicates that only the number of indexes (entries) in the phonebook be returned. In the latter case the Filter, Format, and ListStartOffset values are ignored.

ListStartOffset Value to indicate to the server the offset of the first entry to return. A value of 0 request that the list starts from the beginning.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
 BTPBAP_ERROR_NOT_INITIALIZED
 BTPBAP_ERROR_INVALID_PARAMETER

Possible Events:

etPBAP_PullvCardListing_Confirmation

etPBAP_Close_Port_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.

PBAP_Pull_vCard_Listing_Response

This function responds to a Pull vCard Listing to the issuing client in XML format.

The result is returned in XML format <x-bt/vcard-listing> with each entry handle in the following form:

```
<card handle = "0.vcf" name = "StonestreetOne;John"/>
```

Prototype:

```
int BTPSAPI PBAP_Pull_vCard_Listing_Response(unsigned int BluetoothStackID,
  unsigned int PBAPID, Byte_t ResponseCode, Word_t *PhonebookSize,
  Byte_t *NewMissedCalls, unsigned int BufferSize, Byte_t *Buffer,
  unsigned int *AmountWritten)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to get the current Server Mode Mask.
ResponseCode	The Response Code to be associated with this response. The following values are valid for the Response Code parameter:

PBAP_OBEX_RESPONSE_CONTINUE

PBAP_OBEX_RESPONSE_OK
 PBAP_OBEX_RESPONSE_NOT_FOUND
 PBAP_OBEX_RESPONSE_SERVICE_UNAVAILABLE
 PBAP_OBEX_RESPONSE_BAD_REQUEST
 PBAP_OBEX_RESPONSE_NOT_IMPLEMENTED
 PBAP_OBEX_RESPONSE_UNAUTHORIZED
 PBAP_OBEX_RESPONSE_PRECONDITION_FAILED
 PBAP_OBEX_RESPONSE_NOT_ACCEPTABLE
 PBAP_OBEX_RESPONSE_FORBIDDEN

PhonebookSize	This value is only filled if the <u>MaxListCount</u> on the Request is 0, incating the number of Phonebook indexes (entries) are required.
NewMissedCalls	This value is only filled if the target phonebook is the missed call history, <i>mch.vcf</i> , which indicates the number of missed calls not currently logged into <i>mch.vcf</i> .
BufferSize	Size in bytes of the data to be included in the response. This data is taken from the <u>Buffer</u> pointer. If the <u>Buffer</u> pointer is valid and this parameter is non-zero, a Body or End-Of-Body header will be included in this response.
Buffer	Pointer to the data to be included in this response. If this pointer is valid and <u>BufferSize</u> is non-zero, a Body or End-Body header will be included in this response.
AmountWritten	This parameter is a pointer to a variable that will contain the total amount of data that was successfully written in the current response packet. If this value is less than the requested <u>BufferSize</u> then only a portion of the data was sent. A subsequent call to this function will be required to send the remaining data (following receipt of the appropriate indication event).

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
 BTPBAP_ERROR_NOT_INITIALIZED
 BTPBAP_ERROR_INVALID_PARAMETER

Possible Events:

etPBAP_PullvCardListing_Indication

etPBAP_Close_Port_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.

PBAP_Pull_vCard_Entry_Request

This function retrieves a specific vCard Entry from the server. The vCard retrieved is from the current folder of virtual folder architecture on the PBAP server (see PBAP_Set_Phonebook_Request).

Prototype:

```
int BTPSAPI PBAP_Pull_vCard_Entry_Request(unsigned int BluetoothStackID, unsigned
int PBAPID, char *ObjectName, DWORD_t FilterLow, DWORD_t FilterHigh,
PBAP_Format_t Format)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to get the current Server Mode Mask.
ObjectName	The name of the entry object to retrieve. Ex.: "1.vcf".
FilterLow	This is the low DWORD of the filter used to indicate which attributes to include in the requested vCard object. To request all attributes available in each vCard, leave this value 0, otherwise, it can be set to one or more of the following values OR'ed:

```
PBAP_FILTER_VERSION
PBAP_FILTER_FN
PBAP_FILTER_N
PBAP_FILTER_PHOTO
PBAP_FILTER_BDAY
PBAP_FILTER_ADR
PBAP_FILTER_LABEL
PBAP_FILTER_TEL
PBAP_FILTER_EMAIL
PBAP_FILTER_MAILER
PBAP_FILTER_TZ
PBAP_FILTER_GEO
PBAP_FILTER_TITLE
PBAP_FILTER_ROLE
PBAP_FILTER_LOGO
PBAP_FILTER_AGENT
PBAP_FILTER_ORG
```


PBAP_FILTER_NOTE
 PBAP_FILTER_REV
 PBAP_FILTER_SOUND
 PBAP_FILTER_URL
 PBAP_FILTER_UID
 PBAP_FILTER_KEY
 PBAP_FILTER_NICKNAME
 PBAP_FILTER_CATEGORIES
 PBAP_FILTER_PROID
 PBAP_FILTER_CLASS
 PBAP_FILTER_SORT_STRING
 PBAP_FILTER_X_IRMC_CALL_DATETIME

FilterHigh This is the high DWORD of the filter. At this time there are only Proprietary bits that are vendor specific. The proprietary attribute mask i describes the client requirements for proprietary vCard attributes. If the the following bit is set then bits that are above this one are defined specific to that vendor and for this to work, the client must be able to identify the server as a device that uses the same mask.

PBAP_FILTER_PROPRIETARY_FILTER_HIGH

Format This is the vCard format that is requested to be returned. This value is one of the following:

formatvCard21
 formatvCard30
 formatDefault

The default format is 2.1.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
 BTPBAP_ERROR_NOT_INITIALIZED
 BTPBAP_ERROR_INVALID_PARAMETER

Possible Events:

etPBAP_PullvCardEntry_Confirmation

etPBAP_Close_Port_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.

PBAP_Pull_vCard_Entry_Response

This function responds to a Pull vCard Entry with the specific vCard entry requested by the issuing client with attributes as defined by the filter on the request.

This function should be called in response to the etPBAP_PullPhoneBook_Request event. Multiple PullvCardEntry responses may be required to complete a single Object transaction.

Several parameters to this function cause the inclusion of optional OBEX headers. The ObjectName parameter can be used to include an OBEX Name header in the response. The ObjectLength parameter can be used to include an OBEX Length header in the response. The Buffer and BufferSize parameters determine if the response should contain data.

Responses which contain data (BufferSize is non-zero and Buffer is a valid pointer) include a Body or End-Of-Body OBEX header. The response code determines which header to include. An OK response indicates that all data for the response has been included in the Buffer. A CONTINUE response indicates that not all data for the response was included in Buffer (or Buffer is NULL and no data is included).

NOTE: The minimum vCard attributes included are based upon the format of the vCard requested.

- Mandatory attributes for vCard 2.1 returned are VERSION ,N and TEL.
- Mandatory attributes for vCard 3.0 returned are VERSION, N, FN and TEL

Prototype:

```
int BTPSAPI PBAP_Pull_vCard_Entry_Response(unsigned int BluetoothStackID,  
      unsigned int PBAPID, Byte_t ResponseCode, unsigned int BufferSize, Byte_t *Buffer,  
      unsigned int *AmountWritten)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to get the current Server Mode Mask.
ResponseCode	The Response Code to be associated with this response. The following values are valid for the Response Code parameter:

PBAP_OBEX_RESPONSE_CONTINUE
PBAP_OBEX_RESPONSE_OK
PBAP_OBEX_RESPONSE_NOT_FOUND
PBAP_OBEX_RESPONSE_SERVICE_UNAVAILABLE
PBAP_OBEX_RESPONSE_BAD_REQUEST
PBAP_OBEX_RESPONSE_NOT_IMPLEMENTED
PBAP_OBEX_RESPONSE_UNAUTHORIZED
PBAP_OBEX_RESPONSE_PRECONDITION_FAILED
PBAP_OBEX_RESPONSE_NOT_ACCEPTABLE
PBAP_OBEX_RESPONSE_FORBIDDEN

BufferSize	Size in bytes of the data to be included in the response. This data is taken from the <u>Buffer</u> pointer. If the <u>Buffer</u> pointer is valid and this parameter is non-zero, a Body or End-Of-Body header will be included in this response.
Buffer	Pointer to the data to be included in this response. If this pointer is valid and <u>BufferSize</u> is non-zero, a Body or End-Body header will be included in this response.
AmountWritten	This parameter is a pointer to a variable that will contain the total amount of data that was successfully written in the current response packet. If this value is less than the requested <u>BufferSize</u> then only a portion of the data was sent. A subsequent call to this function will be required to send the remaining data (following receipt of the appropriate indication event).

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INVALID_PARAMETER

Possible Events:

etPBAP_PullvCardEntry_Indication

etPBAP_Close_Port_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.

PBAP_Get_Server_Mode

This function queries the current PBAP Server Mode.

Prototype:

```
int BTPSAPI PBAP_Get_Server_Mode(unsigned int BluetoothStackID,  
    unsigned int PBAPID, unsigned long *ServerModeMask)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server from which to get the current Server Mode Mask.
ServerModeMask	Pointer to a variable to receive the current Server Mode Mask. The following bits are defined for the returned value:

```
PBAP_SERVER_MODE_MANUAL_ACCEPT_CONNECTION  
PBAP_SERVER_MODE_AUTOMATIC_ACCEPT_CONNECTION
```

Return:

Zero if successful.

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

```
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID  
BTPBAP_ERROR_NOT_INITIALIZED  
BTPBAP_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.

PBAP_Set_Server_Mode

This function is responsible for setting the current PBAP Server Mode.

Prototype:

```
int BTPSAPI PBAP_Set_Server_Mode(unsigned int BluetoothStackID,  
    unsigned int PBAPID, unsigned long ServerModeMask)
```

Parameters:

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

PBAPID	Unique identifier of the PBAP Server on which to set the new Server Mode Mask.
ServerModeMask	The Server Mode Mask to set. The following bits are defined:

PBAP_SERVER_MODE_MANUAL_ACCEPT_CONNECTION
PBAP_SERVER_MODE_AUTOMATIC_ACCEPT_CONNECTION

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_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.

PBAP_EnableSRM

The following function is responsible for providing a mechanism to enable SRM (Single Response Mode).

Prototype:

int BTPSAPI **PBAP_EnableSRM**(unsigned int BluetoothStackID, unsigned int PBAPID)

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server on which to enable single response mode.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INVALID_PARAMETER

Possible Events: 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.

PBAP_EnableSRMP

The following function is responsible for providing a mechanism to enable the SRMP (Single Response Mode Parameter) state of the PBAP Client or Server.

Prototype:

```
int BTPSAPI PBAP_EnableSRMP(unsigned int BluetoothStackID, unsigned int PBAPID,  
    PBAP_Role_t Role)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server on which to enable the single response mode parameter state.
PBAP_Role_t	Role of the PBAP server (client or server). May be one of the following: PBAP_Client PBAP_Server

Return:

Zero if successful.

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

```
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID  
BTPBAP_ERROR_NOT_INITIALIZED  
BTPBAP_ERROR_INVALID_PARAMETER
```

Possible Events: 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.

PBAP_DisableSRM

The following function is responsible for providing a mechanism to disable SRM (Single Response Mode).

Prototype:

```
int BTPSAPI PBAP_DisableSRM(unsigned int BluetoothStackID, unsigned int PBAPID)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server on which to disable single response mode.

Return:

Zero if successful.

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

```
BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID  
BTPBAP_ERROR_NOT_INITIALIZED  
BTPBAP_ERROR_INVALID_PARAMETER
```

Possible Events: 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.

PBAP_Set_Folder_Version

The following function is responsible for setting the initial folder version as maintained by the application.

Prototype:

```
int BTPSAPI PBAP_Set_Folder_Version(unsigned int BluetoothStackID, unsigned int  
PBAPID, Boolean_t IsPrimary, Byte_t FolderVersion[16])
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server on which to disable the single response mode parameter state.
IsPrimary	TRUE if the folder version is the PrimaryFolderVersion, FALSE if the folder version is the SecondaryFolderVersion.

FolderVersion 128-bit counter value to be initialized.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
 BTPBAP_ERROR_NOT_INITIALIZED
 BTPBAP_ERROR_INVALID_PARAMETER

Possible Events: 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.

PBAP_Increase_Folder_Version

The following function is responsible for incrementing folder version as specified in PBAP v1.2.3 clause 5.1.4.9. In case the folder version counter rolls over, Database Identifier automatically increases. Optionally application may set a separate DatabaseIdentifier by calling the PBAP_Set_Database_Identifier API by detecting the roll over or start over scenario.

Prototype:

```
int BTPSAPI PBAP_Increase_Folder_Version(unsigned int BluetoothStackID, unsigned
    int PBAPID, Boolean_t IsPrimary, Byte_t *NewFolderVersion, Byte_t
    *NewDatabaseIdentifier)
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server on which to disable the single response mode parameter state.
IsPrimary	TRUE if the folder version to update is the PrimaryFolderVersion, FALSE if the folder version to update is the SecondaryFolderVersion.
NewFolderVersion	New folder version after increment. Argument can be NULL. If non-NULL, the pointer must point to a contiguous array of at least 16 octets.
NewDatabaseIdentifier	The new database identifier. Can be same or incremented by one depending on the roll over condition. Can be set to NULL

if not required by the application. If non-NULL, the pointer must point to a contiguous array of at least 16 octets.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED
BTPBAP_ERROR_INVALID_PARAMETER

Possible Events: 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.

PBAP_Set_Database_Identifier

The following function is responsible for setting database identifier as specified in PBAP 1.2.3 clause 5.1.4.10. This API is primarily used to set the initial Database Identifier as maintained by the application. In case the folder version counter rolls over while calling PBAP_Increase_Folder_Version, the previously set Database Identifier automatically increases. If the auto-increment is not preferred, optionally the application may set a separate DatabaseIdentifier by calling this API again manually by detecting the roll over or start over scenario of Folder Version or Contact X-BT-UID.

Prototype:

```
int BTPSAPI PBAP_Set_Database_Identifier(unsigned int BluetoothStackID, unsigned int PBAPID, Byte_t DatabaseIdentifier[16])
```

Parameters:

BluetoothStackID ¹	Unique identifier assigned to this Bluetooth Protocol Stack via a call to BSC_Initialize.
PBAPID	Unique identifier of the PBAP Server on which to disable the single response mode parameter state.
DatabaseIdentifier	128-bit unique identifier of the database.

Return:

Zero if successful.

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

BTPBAP_ERROR_INVALID_BLUETOOTH_STACK_ID
BTPBAP_ERROR_NOT_INITIALIZED

BTPBAP_ERROR_INVALID_PARAMETER

Possible Events: 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.

2.2 PBAP Event Callback Prototypes

The event callback functions described in the PBAP Open Server Port and Open Remote Server Port commands both accept the callback function described by the following prototype.

PBAP_Event_Callback_t

Prototype of callback function passed in one of the PBAP Open commands.

Prototype:

```
void (BTPSAPI *PBAP_Event_Callback_t)(unsigned int BluetoothStackID,
    PBAP_Event_Data_t *PBAP_Event_Data, unsigned long CallbackParameter)
```

Parameters:

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

```
typedef struct
{
    PBAP_Event_Type_t Event_Data_Type;
    Word_t           Event_Data_Size;
    union
    {
        PBAP_Open_Port_Indication_Data_t      *PBAP_Open_Port_Indication_Data;
        PBAP_Open_Port_Confirmation_Data_t     *PBAP_Open_Port_Confirmation_Data;
        PBAP_Open_Port_Request_Indication_Data_t *PBAP_Open_Port_Request_Indication_Data;
        PBAP_Close_Port_Indication_Data_t       *PBAP_Close_Port_Indication_Data;
        PBAP_Abort_Indication_Data_t            *PBAP_Abort_Indication_Data;
        PBAP_Abort_Confirmation_Data_t          *PBAP_Abort_Confirmation_Data;
        PBAP_Pull_Phonebook_Indication_Data_t   *PBAP_Pull_Phonebook_Indication_Data;
        PBAP_Pull_Phonebook_Confirmation_Data_t *PBAP_Pull_Phonebook_Confirmation_Data;
        PBAP_Set_Phonebook_Indication_Data_t    *PBAP_Set_Phonebook_Indication_Data;
    }
}
```

```

PBAP_Set_Phonebook_Confirmation_Data_t
    *PBAP_Set_Phonebook_Confirmation_Data;
PBAP_Pull_vCardListing_Indication_Data_t
    *PBAP_Pull_vCardListing_Indication_Data;
PBAP_Pull_vCardListing_Confirmation_Data_t
    *PBAP_Pull_vCardListing_Confirmation_Data;
PBAP_Pull_vCardEntry_Indication_Data_t
    *PBAP_Pull_vCardEntry_Indication_Data;
PBAP_Pull_vCardEntry_Confirmation_Data_t
    *PBAP_Pull_vCardEntry_Confirmation_Data;

    } Event_Data;
} PBAP_Event_Data_t;

```

where Event_Data_Type is one of the enumerations of the event types listed in the table in section 2.3. Each data structure in the union is described with its event in section 2.3.

CallbackParameter User-defined parameter (e.g., tag value) that was defined in the callback registration.

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 PBAP Events

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

Event	Description
etPBAP_Open_Port_Indication	Dispatched when a PBAP client connects to a local PBAP Server.
etPBAP_Open_Port_Confirmation	Dispatched when a PBAP client receives a connection response from a remote PBAP Server to which it previously attempted to connect.
etPBAP_Open_Port_Request_Indication	Dispatched when a PBAP client requests a connection to a local PBAP Server. This event only occurs if the local server is in Manual Accept mode.
etPBAP_Close_Port_Indication	Dispatched when a PBAP Server or Client connection has been disconnected from either end.

etPBAP_Abort_Indication	Dispatched when the local PBAP Server receives an OBEX Abort Request from the remote Client.
etPBAP_Abort_Confirmation	Dispatched when the local PBAP Client receives an OBEX Abort Response from the remote Server.
etPBAP_Pull_Phonebook_Indication	Dispatched when the local PBAP Server receives a Pull Phonebook Request.
etPBAP_Pull_Phonebook_Confirmation	Dispatched when the local PBAP Client receives a Pull Phonebook Response.
etPBAP_Set_Phonebook_Indication	Dispatched when the local PBAP Server receives a Set Phonebook Request.
etPBAP_Set_Phonebook_Confirmation	Dispatched when the local PBAP Client receives a Set Phonebook Response.
etPBAP_Pull_vCard_Listing_Indication	Dispatched when the local PBAP Server receives a Pull vCard Listing Request.
etPBAP_Pull_vCard_Listing_Confirmation	Dispatched when the local PBAP Client receives a Pull vCard Listing Response.
etPBAP_Pull_vCard_Entry_Indication	Dispatched when the local PBAP Server receives a Pull vCard Entry Request.
etPBAP_Pull_vCard_Entry_Confirmation	Dispatched when the local PBAP Client receives a Pull vCard Entry Response.

etPBAP_Open_Port_Indication

Dispatched when a PBAP Client connects to a local PBAP Server.

In Manual Accept mode this indication occurs after the etPBAP_Open_Port_Request_Indication and its associated response.

Return Structure:

```
typedef struct
{
    unsigned int    PBAPID;
    BD_ADDR_t      BD_ADDR;
} PBAP_Open_Port_Indication_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Connection for which this event is intended.
BD_ADDR	The Board Address of the remote client that has connected to the locally registered server.

etPBAP_Open_Port_Confirmation

Dispatched when the local PBAP client receives a Connection Response from a remote PBAP server to which it has previously attempted to connect..

This event is only dispatched to servers that are in Manual Accept Mode. The current server mode can be modified by calling PBAP_Set_Server_Mode().

Return Structure:

```
typedef struct
{
    unsigned int    PBAPID;
    unsigned int    PBAPConnectStatus;
    BD_ADDR_t      BD_ADDR;
} PBAP_Open_Port_Confirmation_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Connection for which this event is intended.
PBAPConnectStatus	Specifies the status of the Connection Attempt. Valid values are: PBAP_OPEN_STATUS_SUCCESS PBAP_OPEN_STATUS_CONNECTION_TIMEOUT PBAP_OPEN_STATUS_CONNECTION_REFUSED PBAP_OPEN_STATUS_UNKNOWN_ERROR.
BD_ADDR	The Board Address of the remote server that has responded to the connection request.

etPBAP_Open_Port_Request_Indication

Dispatched when a remote PBAP Client requests a connection to a local PBAP server. The receiving server must respond with a call to PBAP_Open_Request_Response() to accept or reject the outstanding Open Request.

This event is only dispatched to servers that are in Manual Accept Mode. The current server mode can be modified by calling PBAP_Set_Server_Mode().

Return Structure:

```
typedef struct
{
    unsigned int    PBAPID;
    unsigned int    ServerPort;
    BD_ADDR_t      BD_ADDR;
} PBAP_Open_Request_Indication_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
ServerPort	The server port of the local PBAP Server that is being requested to create a new connection.
BD_ADDR	The Board Address of the remote client that has request to connect to the locally registered server.

etPBAP_Close_Port_Indication

Dispatched when a remote client has been disconnected from the local PBAP server. This disconnection can occur for a variety of reasons, and can be initiated from either side of the connection. The server should always be prepared to handle a Close event.

Return Structure:

```
typedef struct
{
    unsigned int    PBAPID;
} PBAP_Close_Port_Indication_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
--------	--

etPBAP_Abort_Indication

Dispatched when a local PBAP server receives an OBEX Abort Request from the remote PBAP client.

Return Structure:

```
typedef struct
{
    unsigned int PBAPID;
} PBAP_Abort_Indication_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
--------	--

etPBAP_Abort_Confirmation

Dispatched when a local PBAP client receives an OBEX Abort Response from the remote PBAP server.

Return Structure:

```
typedef struct
{
    unsigned int PBAPID;
} PBAP_Abort_Confirmation_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Connection for which this event is intended.
--------	--

etPBAP_Pull_Phonebook_Indication

Dispatched when a local PBAP server receives an Pull Phonebook Request from a valid remote PBAP client.

See etPBAP_Pull_vCard_Entry_Confirmation for an example of how the vCard response data appears. For a non-empty Pull Phonebook response this would be a list of one or more vCards delimited by the BEGIN and END tags ordered by index.

Return Structure:

```
typedef struct
{
    unsigned int    PBAPID;
    char            *ObjectName;
    DWord_t         FilterLow;
    DWord_t         FilterHigh;
    PBAP_Format_t   Format;
    Word_t          MaxListCount;
    Word_t          ListStartOffset;
} PBAP_Pull_Phonebook_Indication_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
ObjectName	An ASCII, Null-Terminated string containing the complete Name and Path that was included in the OBEX Name header for this packet.
FilterLow	Lower DWORD of the filter, attributes, to use when creating the vCard response for the corresponding Phonebook and its entries. If this value is 0, then all attributes for each vCard are included in the response.

At a minimum the following should be included on the response regardless of the value.

- Mandatory attributes for vCard 2.1 are VERSION ,N and TEL.
- Mandatory attributes for vCard 3.0 are VERSION, N, FN and TEL

The following is the list of valid filter values:

PBAP_FILTER_VERSION
PBAP_FILTER_FN
PBAP_FILTER_N
PBAP_FILTER_PHOTO
PBAP_FILTER_BDAY
PBAP_FILTER_ADR
PBAP_FILTER_LABEL
PBAP_FILTER_TEL
PBAP_FILTER_EMAIL
PBAP_FILTER_MAILER
PBAP_FILTER_TZ
PBAP_FILTER_GEO
PBAP_FILTER_TITLE
PBAP_FILTER_ROLE
PBAP_FILTER_LOGO
PBAP_FILTER_AGENT
PBAP_FILTER_ORG
PBAP_FILTER_NOTE
PBAP_FILTER_REV
PBAP_FILTER_SOUND
PBAP_FILTER_URL
PBAP_FILTER_UID
PBAP_FILTER_KEY
PBAP_FILTER_NICKNAME
PBAP_FILTER_CATEGORIES
PBAP_FILTER_PROID

PBAP_FILTER_CLASS
 PBAP_FILTER_SORT_STRING
 PBAP_FILTER_X_IRMC_CALL_DATETIME

FilterHigh Higher DWORD of the filter, attributes, to use when creating the vCard response for the corresponding Phonebook and its entries. If the proprietary bit is set, then the client has identified this server as a device that uses the same mask and bits set above this bit are requested attributes to be return.

PBAP_FILTER_PROPRIETARY_FILTER_HIGH

Format Format request for vCard entries in the corresponding Phonebook to return. The default on most systems is version 2.1.

formatvCard21
 formatvCard30
 formatDefault

MaxListCount This is an integer denoting the maximum number of entries to return. If this value is set to '0', then the response will indicate the total number of entries (indexes) in the requested phonebook.

ListStartOffset This is an integer of the first entry to start from. By default, 0, this is from the beginning of the list. If this value is greater than the number of indexes, then no entry data is returned to the issuing client.

etPBAP_Pull_Phonebook_Confirmation

Dispatched when a local PBAP client receives an Pull Phonebook Response from the remote PBAP server.

See etPBAP_Pull_vCard_Entry_Confirmation for an example of how the vCard response data appears. For a non-empty Pull Phonebook response this would be a list of one or more vCards delimited by the BEGIN and END tags ordered by index.

Return Structure:

```
typedef struct
{
    unsigned int    PBAPID;
    Byte_t          ResponseCode;
    unsigned int    BufferSize;
    Byte_t          *Buffer;
```

```

    Boolean_t      Final;
    Word_t         PhonebookSize;
    Byte_t         NewMissedCalls;
} PBAP_Pull_Phonebook_Confirmation_Data_t;

```

Event Parameters:

PBAPID The unique identifier of the Local PBAP Server for which this event is intended.

ResponseCode The ResponseCode to be associated with this response. The following values are valid for the Response Code parameter:

```

PBAP_OBEX_RESPONSE_CONTINUE
PBAP_OBEX_RESPONSE_OK
PBAP_OBEX_RESPONSE_NOT_FOUND
PBAP_OBEX_RESPONSE_SERVICE_UNAVAILABLE
PBAP_OBEX_RESPONSE_BAD_REQUEST
PBAP_OBEX_RESPONSE_NOT_IMPLEMENTED
PBAP_OBEX_RESPONSE_UNAUTHORIZED
PBAP_OBEX_RESPONSE_PRECONDITION_FAILED
PBAP_OBEX_RESPONSE_NOT_ACCEPTABLE
PBAP_OBEX_RESPONSE_FORBIDDEN

```

BufferSize Size in bytes of the data to be included in the response. This data is taken from the Buffer pointer. If the Buffer pointer is valid and this parameter is non-zero, a Body or End-Of-Body header will be included in this response.

Buffer Pointer to the data to be included in this response. If this pointer is valid and BufferSize is non-zero, a Body or End-Body header will be included in this response.

Final Boolean flag to indicate if the Final bit was set in the received OBEX packet. The Final bit indicates that this is the last response in the transaction.

PhonebookSize The number of entries in the target phonebook object, which is only filled if the MaxListCount on the issued request is 0

NewMissedCalls The number of missed calls not currently logged into mch.vcf, if and only if the target phonebook is mch.vcf on the issued request.

etPBAP_Set_Phonebook_Indication

Dispatched when a local PBAP server receives an Set Phonebook Request from a valid remote PBAP client.

Return Structure:

```
typedef struct
```

```

{
    unsigned int          PBAPID;
    PBAP_SetPath_Option_t PathOption;
    Char                  *ObjectName;
} PBAP_Set_Phonebook_Indication_Data_t;

```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
PathOption	A flag that indicates which direction in the virtual folder architecture to set path to.
	<pre> setRoot setDown setUp </pre>
ObjectName	An ASCII, Null-Terminated string containing the complete Name and Path that was included in the OBEX Name header for this packet.

etPBAP_Set_Phonebook_Confirmation

Dispatched when a local PBAP client receives an Set Phonebook Response from the remote PBAP server.

Return Structure:

```

typedef struct
{
    unsigned int    PBAPID;
    Byte_t          *ResponseCode;
} PBAP_Set_Phonebook_Confirmation_Data_t;

```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
ResponseCode	The ResponseCode to be associated with this response. The following values are valid for the Response Code parameter:

```

PBAP_OBEX_RESPONSE_CONTINUE
PBAP_OBEX_RESPONSE_OK
PBAP_OBEX_RESPONSE_NOT_FOUND
PBAP_OBEX_RESPONSE_SERVICE_UNAVAILABLE
PBAP_OBEX_RESPONSE_BAD_REQUEST
PBAP_OBEX_RESPONSE_NOT_IMPLEMENTED
PBAP_OBEX_RESPONSE_UNAUTHORIZED
PBAP_OBEX_RESPONSE_PRECONDITION_FAILED

```

PBAP_OBEX_RESPONSE_NOT_ACCEPTABLE
PBAP_OBEX_RESPONSE_FORBIDDEN

etPBAP_Pull_vCard_Listing_Indication

Dispatched when a local PBAP server receives an Pull vCard Listing Request from a valid remote PBAP client.

Return Structure:

```
typedef struct
{
    unsigned int          PBAPID;
    char                  *ObjectName;
    PBAP_ListOrder_t      ListOrder;
    PBAP_SearchAttribute_t SearchAttribute;
    char                  *SearchValue;
    Word_t                MaxListCount;
    Word_t                ListStartOffset;
} PBAP_Pull_vCard_Listing_Indication_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
ObjectName	An ASCII, Null-Terminated string containing the complete Name and Path that was included in the OBEX Name header for this packet.
ListOrder	This is a value to indicate in what order the listing is returned. It can be one of the following:

orderIndexed
orderAlphabetical
orderPhonetical
orderDefault

	The default order should be Indexed.
SearchAttribute	This is a value to indicate which vCard attribute the search must be performed on.

searchName
searchNumber
searchSound
searchDefault

The default search attribute should be Name.

SearchValue	Value string, based on attribute, to which matching vCards shall be returned in the listing.
MaxListCount	This is an integer denoting the maximum number of entries to return. If this value is set to '0', then the response will indicate the total number of entries (indexes) in the requested phonebook.
ListStartOffset	This is an integer of the first entry to start from. By default, 0, this is from the beginning o the list. If this value is greater than the number of indexes, then no entry data is retuned to the issuing client.

etPBAP_Pull_vCard_Listing_Confirmation

Dispatched when a local PBAP client receives an Pull vCard Listing Response from the remote PBAP server.

On a valid response the format of the Listing will appear similar to the following:

```
<?xml version="1.0"?>
<!DOCTYPE vcard-listing SYSTEM "vcard-listing.dtd">
<vCard-listing version="1.0">
  <card handle = "0.vcf" name = "Public;John"/>
  <card handle = "1.vcf" name = "Public;Steve"/>
  . . .
  <card handle = "n.vcf" name = "StonestreetOne;John"/>
</vCard-listing>
```

Return Structure:

```
typedef struct
{
  unsigned int    PBAPID;
  Byte_t          ResponseCode;
  unsigned int    BufferSize;
  Byte_t          *Buffer;
  Boolean_t       Final;
  Word_t          PhonebookSize;
  Byte_t          NewMissedCalls;
} PBAP_Pull_vCard_Listing_Confirmation_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
ResponseCode	The ResponseCode to be associated with this response. The following values are valid for the Response Code parameter:

```

PBAP_OBEX_RESPONSE_CONTINUE
PBAP_OBEX_RESPONSE_OK
PBAP_OBEX_RESPONSE_NOT_FOUND
PBAP_OBEX_RESPONSE_SERVICE_UNAVAILABLE
PBAP_OBEX_RESPONSE_BAD_REQUEST
PBAP_OBEX_RESPONSE_NOT_IMPLEMENTED
PBAP_OBEX_RESPONSE_UNAUTHORIZED
PBAP_OBEX_RESPONSE_PRECONDITION_FAILED
PBAP_OBEX_RESPONSE_NOT_ACCEPTABLE
PBAP_OBEX_RESPONSE_FORBIDDEN

```

BufferSize	Size in bytes of the data to be included in the response. This data is taken from the <u>Buffer</u> pointer. If the <u>Buffer</u> pointer is valid and this parameter is non-zero, a Body or End-Of-Body header will be included in this response.
Buffer	Pointer to the data to be included in this response. If this pointer is valid and <u>BufferSize</u> is non-zero, a Body or End-Body header will be included in this response.
Final	Boolean flag to indicate if the Final bit was set in the received OBEX packet. The Final bit indicates that this is the last response in the transaction.
PhonebookSize	The number of entries in the target phonebook object, which is only filled if if the <u>MaxListCount</u> on the issued request is 0
NewMissedCalls	The number of missed calls not currently logged into mch.vcf, if and only if the target folder is <i>mch</i> on the issued request.

etPBAP_Pull_vCard_Entry_Indication

Dispatched when a local PBAP server receives an Pull vCard Entry Request from a valid remote PBAP client.

See etPBAP_Pull_vCard_Entry_Confirmation for an example of how the vCard response data appears.

Return Structure:

```

typedef struct
{
    unsigned int    PBAPID;
    char            *ObjectName;
    DWord_t         FilterLow;
    DWord_t         FilterHigh;
    PBAP_Format_t   Format;
} PBAP_Pull_vCard_Entry_Indication_Data_t;

```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
ObjectName	An ASCII, Null-Terminated string containing the complete Name and Path that was included in the OBEX Name header for this packet.
FilterLow	<p>Lower DWORD of the filter, attributes, to use when creating the vCard response for the corresponding Phonebook and its entries. If this value is 0, then all attributes for the requested vCard are included in the response.</p> <p>At a minimum the following should be included on the response regardless of the value.</p> <ul style="list-style-type: none">• Mandatory attributes for vCard 2.1 are VERSION ,N and TEL.• Mandatory attributes for vCard 3.0 are VERSION, N, FN and TEL

The following is the list of valid filter values:

PBAP_FILTER_VERSION
PBAP_FILTER_FN
PBAP_FILTER_N
PBAP_FILTER_PHOTO
PBAP_FILTER_BDAY
PBAP_FILTER_ADR
PBAP_FILTER_LABEL
PBAP_FILTER_TEL
PBAP_FILTER_EMAIL
PBAP_FILTER_MAILER
PBAP_FILTER_TZ
PBAP_FILTER_GEO
PBAP_FILTER_TITLE
PBAP_FILTER_ROLE
PBAP_FILTER_LOGO
PBAP_FILTER_AGENT
PBAP_FILTER_ORG
PBAP_FILTER_NOTE
PBAP_FILTER_REV
PBAP_FILTER_SOUND
PBAP_FILTER_URL
PBAP_FILTER_UID
PBAP_FILTER_KEY
PBAP_FILTER_NICKNAME
PBAP_FILTER_CATEGORIES

PBAP_FILTER_PROID
 PBAP_FILTER_CLASS
 PBAP_FILTER_SORT_STRING
 PBAP_FILTER_X_IRMC_CALL_DATETIME

FilterHigh Higher DWORD of the filter, attributes, to use when creating the vCard response for the corresponding Phonebook and its entries. If the proprietary bit is set, then the client has identified this server as a device that uses the same mask and bits set above this bit are requested attributes to be return.

PBAP_FILTER_PROPRIETARY_FILTER_HIGH

Format Format request for vCard entry to return. The default on most systems is version 2.1.

formatvCard21
 formatvCard30
 formatDefault

etPBAP_Pull_vCard_Entry_Confirmation

Dispatched when a local PBAP client receives an Pull vCard Entry Response from the remote PBAP server.

On a valid response the format of a vCard will appear similar to the following:

```
BEGIN:VCARD
VERSION: 2.1
FN:John
N:Public;John;Q.;;
EMAIL;INTERNET:jpublic@stonestreetone.com
TEL;PREF;WORK:502-595-7000
TEL;PREF;CELL:502-555-1212
ADR;POSTAL;WORK:322 W Main St;Louisville, KY;;;40202;USA
END:VCARD
```

Return Structure:

```
typedef struct
{
    unsigned int    PBAPID;
    Byte_t          ResponseCode;
    unsigned int    BufferSize;
    Byte_t          *Buffer;
    Boolean_t       Final;
} PBAP_Pull_vCard_Entry_Confirmation_Data_t;
```

Event Parameters:

PBAPID	The unique identifier of the Local PBAP Server for which this event is intended.
ResponseCode	The ResponseCode to be associated with this response. The following values are valid for the Response Code parameter:

PBAP_OBEX_RESPONSE_CONTINUE
PBAP_OBEX_RESPONSE_OK
PBAP_OBEX_RESPONSE_NOT_FOUND
PBAP_OBEX_RESPONSE_SERVICE_UNAVAILABLE
PBAP_OBEX_RESPONSE_BAD_REQUEST
PBAP_OBEX_RESPONSE_NOT_IMPLEMENTED
PBAP_OBEX_RESPONSE_UNAUTHORIZED
PBAP_OBEX_RESPONSE_PRECONDITION_FAILED
PBAP_OBEX_RESPONSE_NOT_ACCEPTABLE
PBAP_OBEX_RESPONSE_FORBIDDEN

BufferSize	Size in bytes of the data to be included in the response. This data is taken from the <u>Buffer</u> pointer. If the <u>Buffer</u> pointer is valid and this parameter is non-zero, a Body or End-Of-Body header will be included in this response.
Buffer	Pointer to the data to be included in this response. If this pointer is valid and <u>BufferSize</u> is non-zero, a Body or End-Of-Body header will be included in this response.
Final	Boolean flag to indicate if the Final bit was set in the received OBEX packet. The Final bit indicates that this is the last response in the transaction.

3. File Distributions

The header files that are distributed with the PBAP Library are listed in the table below.

File	Contents/Description
PBAPAPI.h	PBAP API definitions
SS1BTPBA.h	PBAP Include file