



Release Notes

BTW 6.1.0.1506 SDK

April 20, 2009

Proprietary Information

© Broadcom Corporation, 2009, All rights reserved.
Printed in United States of America.
15435 Innovation Drive
San Diego, CA 92128
Phone: (858) 385-8800
Fax: (858) 385-8810

LICENSED SOFTWARE

© Copyright 2009, Broadcom® Corporation ("Broadcom"). All rights reserved.

WARNING:

This software and accompanying documentation are protected by copyright law and international treaties. Unauthorized reproduction or distribution of this software, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Use of this software is governed by the terms of the end user license agreement that accompanies or is included with such software. Unless otherwise noted in the end user license agreement, or herein, no part of the documentation accompanying this software, whether provided in printed or electronic form may be reproduced in any form, or stored in a database or retrieval system, or transmitted in any form or by any means, or used to make any derivative work (such as translation, transformation, or adaptation) without the express, prior written consent of Broadcom.

Trademarks

Broadcom® and the pulse logo are trademarks of Broadcom Corporation and/or its subsidiaries in the United States and certain other countries. Microsoft® and Windows® are trademarks of Microsoft Corporation.

Bluetooth® is a trademark of the Bluetooth SIG. Any other trademarks mentioned are the property of their respective owners.

Compatibility

This section discusses the compatibility of applications produced using the SDK when deployed against Broadcom's various WIDCOMM BTW Bluetooth software products. BTW refers to Broadcom's WIDCOMM Bluetooth software products for Windows PCs. Currently, there are two major BTW products - Broadcom's Bluetooth Software for Windows, and Broadcom's Vista Profile Pack. The latest SDK release is compatible with both BTW products.

Broadcom's Bluetooth Software for Windows refers to all versions of the Broadcom/WIDCOMM Bluetooth Stack software. Previous releases are identified by version as BTW 1.x, 3.x, and 4.x. Current versions are referred to as BTW 5, and are identified by version as 5.x.x.x. Broadcom's Vista Profile Pack is a version of the Broadcom/WIDCOMM Bluetooth software that runs on the Microsoft Bluetooth stack. It is referred to as BTW 6, and is identifiable by version as 6.x.x.x.

Additionally, there is Broadcom's Vista Audio Pack. This product is identified by version as 5.2.x.x. SDK compatibility with BTW versions 5.2.x.x is limited, as this product supports audio profiles only. Contact Broadcom Technical Support directly at http://www.broadcom.com/products/bluetooth_support.php for information on SDK and Vista Audio Pack compatibility.

Compatibility with BTW 5, Bluetooth Software for Windows

BTW and SDK software versions are designed to be forward- and backward-compatible with all combinations of SDK and BTW 5 software, where both components are version 1.4.2.10 SP5 or greater. However, changes in BTW 5 stack software have created some limitations to compatibility.

Specifically, the following compatibility limitations exist on BTW 5 deployments:

- In SDK applications built from SDK releases prior to version 6.1.0.1502, SCO/eSCO audio connections can be established, but the audio device is never enabled on the system and so the audio stream is not available. This impacts BTW 5 deployments on 5.1 versions 5.1.0.3400 and greater, and all 5.5 versions. This issue was fixed in the 6.1.0.1502 SDK release.
- SDK applications built from SDK releases prior to version 6.1.0.1501 cannot read Service Discovery Records on all BTW 5.5 versions. This issue was resolved in 6.1.0.1501 SDK release.
- SDK applications built from SDK releases prior to 6.1.0.1504 had various results and failures using the `CBtIf::Bond()` method for pairing when deployed on Bluetooth 2.1 SSP capable platforms (all BTW 5.5 deployments, and Vista deployments with SP1 and Wireless Feature Pack, or SP2). These issues are fixed in SDK version 6.1.0.1504. Note that the `Bond()` API will now function correctly on all deployments where 1 or both entities being paired support Bluetooth 2.0 or less – if both entities support Bluetooth 2.1, the Legacy `Bond()` method cannot work. For this reason, the `Bond()` API is deprecated, and developers are encouraged to migrate their applications to using the new `BondEx()` method, designed to handle all supported pairing scenarios.

For these reasons, Broadcom cannot guarantee full forward compatibility for applications built with previous versions of the SDK. Broadcom recommends recompiling applications with the 6.1.0.1504 version of the SDK to ensure forward and backward compatibility with all BTW 5 stack versions.

All of the SDK APIs and classes are fully supported in Bluetooth Software for Windows deployments, subject to documented deprecations and with the exceptions described below in "Compatibility and New Features".

Compatibility with BTW 6, Vista Profile Pack

SDK version 6.1.0.1506 is fully forward and backward compatible with all versions of Vista Profile Pack (BTW 6). BTW 6 compatibility was first introduced in SDK version 6.1.0.1501. Applications built with SDK versions prior to 6.1.0.1501 are only compatible with BTW versions prior to BTW 6, whereas applications built with SDK version 6.1.0.1501 or greater are forward and backward compatible with all supported BTW versions, including BTW 6.

BTW 6.1.0.1506 SDK

Vista Profile Pack runs on Microsoft's Bluetooth stack. As a result, some SDK features, APIs, and classes may not be fully supported in a particular BTW 6 version. See the "BTW 6 Vista Profile Pack Limitations" section in this document, and consult the latest SDK Release Notes for up to date SDK and BTW 6 API and class support details, available at http://www.broadcom.com/products/bluetooth_support.php.

Compatibility and Visual Studio 2005 Support

If an SDK application is built using Microsoft Visual Studio 2005, the application must be prepared to redistribute the Broadcom and the Microsoft runtime and interface library files to target platforms when appropriate. See the SDK Programmer's Guide, Section 3 – "Build Environments" for information on redistribution requirements.

Compatibility and New Features

When compatible BTW and SDK versions (as discussed earlier) are used together, the following apply:

- Applications built using an SDK version older than the BTW software version on which the application runs will run properly but may not be able to take advantage of newer features added in the more recent BTW software version.
- Applications built using an SDK version newer than the BTW software version on which the application runs will not run properly if it depends on newer features added in the more recent SDK version.

Changes in the SDK software generally consist of additions, such as new functions or new codes appended to enumerated constant lists. Such changes are documented in the SDK Release Notes for the version to which the change applies. In addition, concise comments in the affected SDK header file identify changes and specify the BTW and SDK versions in which the change(s) occur.

Changes from 6.1.0.1504

These Release Notes document the changes in the SDK from the previous SDK release, version 6.1.0.1504. For information on possible porting requirements when migrating your application from SDK versions previous to 6.1.0.1504, refer to the Release Notes for those versions, available at: http://www.broadcom.com/products/bluetooth_support.php.

There were significant changes in the 6.1.0.1501 SDK release that may impact developers, particularly the new CSdpService::Commit() method requirement, documented in that version of the Release Notes; the 6.1.0.1502 SDK release also contained developer impact changes, particularly the introduction of the CL2Caplf::AssignPsmEx() method and deprecation of the CL2Caplf::AssignPsm() method. Similarly, the 6.1.0.1504 SDK released introduced the CBtlf::BondEx() method and deprecated the CBtlf::Bond() method.

The 6.1.0.1506 release is intended primarily as a maintenance release to address changes in the SDK library code required in order to conform to Microsoft's OEM Ready program. This release addresses all OEM Ready issues discovered in the SDK library code.

One change that may require development modification:

BLTH00163441 changed the constant OBEX_DEFAULT_MTU to nearly 64K. This was done to increase data transfer performance to up to 4X increase.

Developer Impact: OBEX based communications may need to adjust buffer sizes or receive data logic to cater to the new default size, or specify non-default option as appropriate.

BTW 6 Vista Profile Pack Limitations

Limitations apply to the 6.1.0.1506 SDK release when applications are deployed against Vista Profile Pack BTW 6 target systems (systems running Microsoft Bluetooth stacks). Some classes and APIs are either currently not supported or are only supported with limited functionality under those deployments. Some of the limitations vary depending on the BTW 6 major version number, 6.0.x.x, 6.1.x.x, or 6.2.x.x currently.

This list of limitations may change as more functionality becomes available through the stack and SDK for these deployments. Check http://www.broadcom.com/products/bluetooth_support.php for updates to these Release Notes.

Classes:

- CL2CapConn and CL2CapIF are not supported in 6.0, but are supported in 6.1 and greater.
- CDunClient, CLapClient, CSppClient, and CSppServer are not supported in any 6.x version.

Sample apps:

- BlueComChat will not work on any 6.x version.
- BluePrint HCRP print profile will not work on 6.0 PP, but will work on 6.1 and greater.
- BlueTime will not work on 6.0, but will work on 6.1 and greater.
- BlueAudio only works for SCO on 6.0. 6.1 and greater support both SCO and eSCO.

APIs:

- CRfCommPort::OnEventReceived only reports a limited subset of events in all 6.x versions: RXFLAG, TXEMPTY, TXCHAR, CONNECTED_RFCOMM, CONNECT_ERR
- GetConnectionStats (all classes where present) do not support the RSSI data field in 6.x versions.
- SetEScoMode (all classes where present) supported in 6.1 and greater, not supported in 6.0.
- The following APIs are not supported on any 6.x version:
 - SetLinkSupervisionTimeout (all classes where present)
 - ReadEScoLinkData, ChangeEScoLinkParms, EScoConnRsp (all classes where present)
 - CL2CapConn:
 - Reconfigure
 - OnConnectPendingReceived
 - OnCongestionStatus
 - CBtIf:
 - ReadLinkMode
 - SendVendorSpecificHciCmd
 - SetSniffMode
 - CancelSniffMode
 - IsRemoteDevicePresent
 - GetLocalServiceName
 - GetNextLocalServiceName
 - CreateCOMPortAssociation
 - RemoveCOMPortAssociation
 - ReadCOMPortAssociation
 - CRfCommPort:
 - SetFlowEnabled
 - SetModemSignal
 - GetModemStatus
 - SendError
 - Purge

Resolved Issues

BLTH00153807

Port fixes from BTW baseline to SDK release for OEM Ready compliance.

Release Note:

Ported fixes into the SDK from the BTW 6.2 baseline for various handle and critical section errors that caused OEM Ready failures.

BLTH00159822

App Verifier failure, memory access error on close BlueTime after session.

Release Note:

BlueTime dialog destructor tries to detect situation where app is being closed without stopping session, and kills clock thread in that case. But normal path to stop session first was not clearing thread pointer, so the check caused an access exception in the normal path. Cleared the thread pointer after killing the thread in the Stop Session logic, destructor now cleanly handles both cases.

BLTH00160577

Need to cleanly and silently support Just Works SSP for Microsoft stack.

Release Note:

Cleaned up Just Works SSP support so that if SDK app initiated the pairing, SDK automatically acknowledges Just Works request.

BLTH00160894

SCO/ESCO state not initialized from default radio button position.

Release Note:

Initialize to SCO mode on dialog init.

BLTH00163157

BluePrint AppVerifier error on print server selection.

Release Note:

Corrected copy of remote server name to avoid crash.

BLTH00163441

Change BIP and SDK to run 64K OBEX MTU.

Release Note:

Changed OBEX and BIP default MTUs to almost 64K for significant performance improvement.

BTW 6.1.0.1506 SDK

BLTH00164434

BlueTime application crashes - while trying to pair the device for the 2nd time using correct pinkey.

Release Note:

Fixed multiple issues, including bad handle checks in SDK L2CAP, thread/callback problems in BlueTime, and app verifier crash from (bad) multiple thread simultaneous l2cap writes.

BLTH00164571

Pin-code asked twice from the user when BondEx is used for legacy pairing.

Release Note:

BondEx() method does not take pin-code as an argument and therefore when the pairing method used is legacy, then it has to ask the user for pin-code. The issue was that any earlier (before calling BondEx() method) pin-code entry by the user was ignored and the user was always asked again to enter the pin-code. Now, the pin-code entered by the user in the UI is saved and when the pin-code request callback is called after using BondEx(), the same pin-code is returned.

BLTH00166622

BluePrint app - Data transfer fails during print operation using HCRP & SPP Profile

Release Note:

Corrected GUID assignment conflict between BluePrint HCRP sample and BTW stack HCRP application.

BLTH00167126

Blueobex as client app - Crashes while closing the application after create obex connection fails.

Release Note:

Problem came from obex initialized flag not being cleared on SDK close, so re-init of SDK from same app caused obex critical section to not be re-initialized. Fixed to clear flag on SDK close.

BLTH00168458

L2CAP clients can leave handles open and leak memory on deregister.

Release Note:

Changed CL2CapIf destructor and internal deregister logic to free PSM and deregister with the btwl2cap driver and shut down the driver reader thread gracefully.

BLTH00168816

L2CAP remote MTU ignored.

Release Note:

With introduction of Microsoft stack support for L2CAP, MTU was being ignored from the app, and remote MTU was never reported back to the app. In addition, there was a conflict between the old Broadcom stack default MTU and Microsoft stack support MTU, such that if the app requested the default value, the actual value would be different than expected. Fixed all these issues, and added test code in BlueTime sample app to demonstrate/exercise. Also discovered and fixed Broadcom stack post-negotiation behavior so that remote MTU is reported correctly as final negotiated value, rather than remote initial request value.

Known Issues

BLTH00051402

OBEX header manipulation results in memory leaks. SDK applications experiencing memory leaks if the application performs operations on a COBexHeaders object received via a callback.

Workaround: Do not perform manipulations on received COBexHeaders object to turn around and resend, instead, copy object to local storage and return from callback.

BLTH00052086

CL2CapConn::Reconfigure only affects MTU.

BLTH00052416

Sample apps demonstrate CBtIf object usage badly. Only 1 CBtIf object should be instantiated.

BLTH00077280

BlueObex fails with new app param overflow logic. Unlimited number of Application Parameters now allowed, changed from 3 previously, so old check for expected failure after adding 4th now does not fail.

BLTH00080426

Audio gateway pops "no device" warning bubble on audio events even when SDK SCO connected

Workaround: Disable native stack audio gateway service and restart Bluetooth.

BLTH00087672

SDK CPrintClient uses internal CBtIf, conflicts with app object if exists.

Workaround: destroy the app CBtIf before starting the print session, and reinstantiate after the print session.

BLTH00087685

SPP print sessions from SDK leave some printers in bad state.

BLTH00087689

CHeadphoneClient GetConnectionStatistics only returns bConnected properly, no statistics.

BLTH00093909

BlueHeadphone and BlueAudio must pair with headset before able to connect on Vista.

Workaround: ensure pairing exists before trying to make audio connections.

BLTH00094318

CBtIf::Role_Switch always returns TRUE even if Role Switch fails.

BLTH00096952

BlueObex connection disconnect automatically after 3 minutes.

BLTH00115574

SDK classes need more thread protection. L2CAP, RFCOMM, and CBtIf classes can cause crashes if objects deleted while callbacks firing.

BLTH00118592

SDK Sample app inquiry/discovery improvements. Several known issues in Inquiry and Discovery process in most of the sample apps. BlueTime alone has fixed a number of these issues, use BlueTime as a model rather than other samples for Inquiry and Discovery. Some issues still exist for corner case operations, such as cancelling Discovery, closing the window while Inquiry still active, etc.

BTW 6.1.0.1506 SDK

BLTH00128125

Sample applications should disallow unsupported operations based on platform.

BLTH00122904

SDK Programmer's Guide references Link mode definitions in wrong file (BtIfDefinitions.h, actually in BtIfClasses.h).

BLTH00127221

SDK Sample apps should cleanly do successive client and server sessions without requiring app restart.
Workaround: Launch separate executable instances for each client or server session.

BLTH00127395

Blue Headphone returns no services found if already connected.
Workaround: check if connected to remote already before trying to discover A2DP service record.

BLTH00129615

BlueTime: Selecting OK before service is detected on the client hangs the sample app.
Workaround: Do not select OK until service is shown in dialog.

BLTH00136479

SDK Service Discovery performed twice unnecessarily on BTW 6

BLTH00138449

CHeadphoneClient class only supports 1 instantiated object.

BLTH00157205

BlueTime can crash if Bond window closed while Bond operation still active.
Workaround: Do not close Bond dialog window while Bond operations are in progress.

BLTH00157212

BlueTime can crash if Inquiry window closed while Inquiry results still being processed
Workaround: Do not close Inquiry window while Inquiry results still being processed.

BLTH00162399

Need to add check to ensure SDK app is in user mode only.
Workaround: Only use the SDK for user mode applications.

BLTH00164234

Bluechat as client app, fails to transfer more than 8190 bytes in one transfer.
Workaround: 1) Bluechat does not listen for or do anything with flow control events from the remote – need to should listen and throttle sending until clear by disable/enable Send button. 2) Bluechat does not check after call to Write if all data was written - can return count indicating partial write, which means it should then move the pointer and try to write the rest, till done.

BLTH00165621

CL2CapConn::GetCid() does not return actual CID on Microsoft stack.
Workaround: None – Microsoft stack does not provide the actual CID, the CID returned from GetCid is just an internal handle on Microsoft stack.

BLTH00165440

BlueObex as client app fails to reconnect with aborted/restarted server.

BLTH00174787

Under OEM ready condition set, Blueprint app crashes on inquiry dialog window close.
Note: This is only evidenced with OEM Ready checks, and is in the sample only, not the SDK library code.